Coordinated Control of Small, Remotely Operated and Submerged Vehicle-Manipulator Systems

by

Serdar Soylu
B.Eng., Dokuz Eylul University, 2002
M.A.Sc, University of Victoria, 2005

A Dissertation Submitted in Partial Fulfillment
of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Mechanical Engineering

Supervisory Committee

Coordinated Control of Small, Remotely Operated and Submerged Vehicle-Manipulator Systems

by

Serdar Soylu
B.Eng., Dokuz Eylul University, 2002
M.A.Sc., University of Victoria, 2005

Supervisory Committee

Dr. Bradley J. Buckham, Department of Mechanical Engineering
Supervisor

Dr. Yang Shi, Department of Mechanical Engineering
Departmental Member

Dr. Panajotis Agathoklis, Department of Electrical Engineering
Outside Member

Abstract

Supervisory Committee

Dr. Bradley J. Buckham, Department of Mechanical Engineering
Supervisor

Dr. Yang Shi, Department of Mechanical Engineering
Departmental Member

Dr. Pan Agathoklis, Department of Electrical Engineering
Outside Member

Current submerged science projects such as VENUS and NEPTUNE have revealed the need for small, low-cost and easily deployed underwater remotely operated vehicle-manipulator (**ROVM**) systems. Unfortunately, existing small remotely operated underwater vehicles (**ROV**) are not equipped to complete the complex and interactive submerged tasks required for these projects. Therefore, this thesis is aimed at adapting a popular small ROV into a ROVM that is capable of low-cost and time-efficient underwater manipulation. To realize this objective, the coordinated control of ROVM systems is required, which, in the context of this research, is defined as the collection of hardware and software that provides advanced functionalities to small ROVM systems. In light of this, the primary focus of this dissertation is to propose various technical building blocks that ultimately lead to the realization of such a coordinated control system for small ROVMs.

To develop such a coordinated control of ROVM systems, it is proposed that ROV and manipulator motion be coordinated optimally and intelligently. With coordination, the system becomes redundant: there are more degrees of freedom (**DOF**) than required. Hence, the extra DOFs can be used to achieve secondary objectives in addition to the primary end-effector following task with a redundancy resolution scheme. This eliminates the standard practice of holding the ROV stationary during a task and uncovers significant potential in the small ROVM platform.

In the proposed scheme, the ROV and manipulator motion is first coordinated such that singular configurations of the manipulator are avoided, and hence dexterous manipulation is ensured. This is done by using the ROV's mobility in an optimal, coordinated manner.

Later, to accommodate a more comprehensive set of secondary objectives, a fuzzy based approach is proposed. The method considers the human pilot as the main operator and the fuzzy machine as an artificial assistant pilot that dynamically prioritizes the secondary objectives and then determines the optimal motion.

Several model-based control methodologies are proposed for small ROV/ROVM systems to realize the desired motion produced by the redundancy resolution, including an adaptive sliding-mode control, an upper bound adaptive sliding-mode control with adaptive PID layer, and an H_∞ sliding-mode control. For the unified system (redundancy resolution and controller), a new human-machine interface (**HMI**) is designed that can facilitate the coordinated control of ROVM systems. This HMI involves a 6-DOF parallel joystick, and a 3-D visual display and a graphical user interface (**GUI**) that enables a human pilot to smoothly interact with the ROVM systems. Hardware-in-the-loop simulations are carried out to evaluate the performance of the coordination schemes.

On the thrust allocation side, a novel fault-tolerant thrust allocation scheme is proposed to distribute forces and moments commanded by the controller over the thrusters. The method utilizes the redundancy in the thruster layout of ROVM systems. The proposed scheme minimizes the largest component of the thrust vector instead of the two-norm, and hence provides better manoeuvrability.

In the first phase of implementation, a small inspection-class ROV, a Saab-Seaeye Falcon™ ROV, is adopted. To improve the navigation, a navigation skid is designed that contains a Doppler Velocity Log, a compass, an inertial measurement unit, and acoustic position data. The sensor data is blended using an Extended Kalman Filter. The developed ROV system uses the upper bound adaptive sliding-mode control with adaptive PID layer.

The theoretical and practical results illustrate that the proposed tools can transform, a small, low-cost ROVM system into a highly capable, time-efficient system that can complete complex subsea tasks.

Table of Contents

Supervisory Committee ii
Abstract iii
Table of Contents v
List of Figures vii
Acknowledgments ix
CHAPTER 1. INTRODUCTION 1
1.1 Background 1
1.2 Motivation 3
1.3 Objectives 8
1.4 Literature Survey 11
1.4.1 Survey on ROV Dynamics Modelling and Control 11
1.4.2 Survey on ROV Thrust Allocation 14
1.4.3 Survey on ROVM Kinematics 16
1.4.4 Survey on ROVM Dynamics and Control 18
1.5 Methodology 19
1.5.1 ROV Dynamics and Control 19
1.5.2 ROV Thrust Allocation 21
1.5.3 ROVM Kinematics 21
1.5.4 ROVM Dynamics and Control 22
1.6 The Experimental ROVM platform 23
1.7 Thesis Outline 27
CHAPTER 2. SUMMARY OF CONTRIBUTIONS TO ROV DYNAMICS MODELLING and CONTROL 28
2.1 Chapter Review 28
2.2 ROV Control: Station Keeping 29
2.2.1 Station Keeping without Explicit Disturbance Knowledge [A] 30
2.2.2 Station Keeping with Explicit Disturbance Knowledge [B] 33
2.3 ROV Control: Trajectory Following 36
2.3.1 ROV Control with Estimation of Lumped Uncertainty [C] 38
2.3.2 ROV Control with Upper Bound Estimation of Lumped Uncertainty [D] . 40
2.4 ROV Control: Thrust Allocation [C] 46
CHAPTER 3. SUMMARY OF CONTRIBUTIONS TO ROVM SYSTEMS 49
3.1 Chapter Review 49
3.2 ROVM Kinematics 51
3.2.1 Redundancy Resolution with Two Secondary Objectives [E] 51
3.2.2 Redundancy Resolution with Multiple Secondary Objectives [F] 54
3.3 ROVM Coordinated Control 59
3.3.1 Redundancy Resolution Coupled with a Controller [G] 59
3.3.2 Redundancy Resolution Coupled with a Controller and a HMI [H]. 62

CHAPTER 4. CONCLUSIONS AND FUTURE WORK 70

4.1 Conclusions 70
4.1.1 ROV Research 70
4.1.2 ROVM Research 77
4.2 Future Work 82
4.2.1 ROV Research 82
4.2.2 ROVM Research 83

Bibliography 85

Appendix A. MIMO Sliding-Mode and H∞ Controller Design for Dynamic Coupling Reduction in Underwater-Manipulator Systems 92

Appendix B. Dynamics and Control of Tethered Underwater-Manipulator Systems 106

Appendix C. A Chattering-Free Sliding-Mode Controller for Underwater Vehicles with Fault-Tolerant Infinity-Norm Thrust Allocation 115

Appendix D. Automatic Navigation and Control of the Falcon™ ROV 129

Appendix E. Dexterous Task-Priority Based Redundancy Resolution for Underwater Manipulator Systems 204

Appendix F. Redundancy Resolution for Underwater Mobile Manipulators 220

Appendix G. Exploiting Redundancy in Underwater Vehicle-Manipulator Systems 240

Appendix H. New Operation Scheme for Underwater Mobile Manipulator 250

List of Figures

Figure 1: A typical ROVM System In such systems, the major components are tether, ROV chassis, thrusters, on-board cameras and robotic manipulators. Image courtesy of http://seaeye.com). 2

Figure 2: A typical ROVM teleoperation scheme for ROVM systems. In this scheme, there are two pilots operating the system; first pilot drives the ROV and the second pilot commands the manipulator with their respective hand control units (HCU). The ROVM System and the Manipulator HCU images are courtesy of http://seaeye.com. 3

Figure 3: The difference in size can be seen between a work-class ROV and an inspection- class ROV. Image courtesy of http://oceanscan.net. 4

Figure 4: On the left, a work-class ROV is being hand-deployed in the water from a pier. On the right, an inspection-class ROV is being deployed by a crew with a crane attached on the surface vehicle. The image on the right is courtesy of http://polartrec.com. 4

Figure 5: NEPTUNE Canada Folger Passage site map. In this particular node, the instruments are positioned in two clusters; namely, Folger Passage Deep and the Folger Passage Pinnacle with the depth ratings of 100m and 25m, respectively. Image courtesy of [5]. 6

Figure 6: A snapshot from ROPOS's on-board camera. In this picture, ROPOS vehicle is deploying a bottom pressure recorder at the Folger Passage with its hydraulic arm. Image courtesy of http:// neptunecanada.ca. 7

Figure 7: Inspection-class SeaEYE Falcon™ ROV systems. Due to their one to one ratio between its weight and trust, these systems are commonly used in industry. This system was used as a test-bed for the experimental portion of the current work. Image courtesy of [43]. 24

Figure 8: HLK 43000™ 4-DOF hydraulically powered underwater manipulator produced by Hydrolek Ltd. This arm is designed specifically for inspection class ROVs and can be easily mounted to the Falcon ROV. 24

Figure 9: On the left is the Falcon™ HCU that allows the human pilot to operate the vehicle. On the right is the HCU of the HLK 43000 underwater manipulator. By pushing/pulling the corresponding joystick on the manipulator HCU, individual joints can be moved. Currently, the manipulator HCU allows commanding only one joint motion at a time. 25

Figure 10: RSI Parallel-architectured joystick was used to generate reference values for the end-effector motion. In the ROVM coordination work, this joystick replaces the

HCU of the Falcon™ ROV and the HLK 43000 shown in Figure 9. Image courtesy of [45]. 27

Figure 11: A wet-test facility was constructed at Van Isle Marina in Sidney BC. This facility hosts a wide range of equipment necessary for the experiment. 44

Figure 12: A series of water-proof, pressure proof sensor canisters for each sensor were designed. The modular design allows the location optimization of each sensor on the new navigation skid in order to minimize magnetic interferences mostly due to the onboard thrusters. 45

Figure 13: The high-level human pilot sets the desired end-effector motion using a 6-DOF joystick while the low-level artificial pilot determines the optimal posture with respect to the secondary objectives in response to the human-pilot input. The ROVM System image is courtesy of http://seaeye.com. 58

Figure 14: The 3D display provides visual clues to the human pilot. (a) third party view of the ROVM system. (b) First person (camera view) of the end-effector (EE). The display tool can also be used for pilot training. 66

Figure 15: Snapshot of the proposed GUI. The developed tools of Appendix-B, Appendix-F, and Appendix-H were integrated into the GUI for the coordinated control of ROVM systems. 68

Acknowledgments

First, I would like to express my deepest gratitude to my supervisors Dr. Bradley J. Buckham and Dr. Ronald P. Podhorodeski for their excellent guidance and support during the completion of this thesis. I feel grateful for the opportunity they have given me to undertake this work, which has allowed me to grow scientifically and personally. They believed in me when I needed them most, and I will remember their support for the rest of my life.

I would like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for providing financial support for this work. Also, I would like to extend my thanks to Suboceanic Sciences Canada Ltd. in Duncan, BC for generously providing access to their Falcon™ ROV.

I also thank friends and fellow researchers for generously providing help with various technical aspects of the research, such as Alison Proctor, John Zand, Flavio Firmani, Dennis Zakopcan, Scott Beatty, Mark Mosher, Mark Butowski, Baris Ulutas, Kerem Karakoc and Karolien Swaak.

My final words of acknowledgment go to my family and my life partner, Karolien. My family has encouraged me greatly throughout my life in all my decisions and has never deprived me of their support. Finally, my dear Karolien, thank you for your love and for the support that you have generously showed me. Knowing that you are in my life has given me the strength that I need to overcome any challenges.

CHAPTER 1. INTRODUCTION

1.1 Background

Remotely operated underwater vehicles (ROVs) equipped with robotic manipulators play an important role in a number of shallow and deep-water missions for marine science, oil and gas extraction, exploration and salvage [1]. The combined system is referred to as an underwater remotely operated vehicle-manipulator (ROVM) as shown in Figure 1. In these applications, the ROV is used as a mobile platform that delivers a robotic manipulator to a subsea work site. The motions of the ROV and the manipulator are guided independently by a human pilot on a surface support vessel through a long slender tether that provides power and telemetry. For detailed surveys, a precise description of a desired ROV motion can be accomplished by an on-board ROV controller that uses ROV state feedback provided by an acoustic and inertial positioning system. [2]. In most cases, the ROV controller relies on a dynamic model to intelligently command conventional propeller type thrusters arranged on the ROV chassis. Manipulator units are generally add-on technologies produced by independent manufacturers, and hence the manipulator most often has an independent control system.

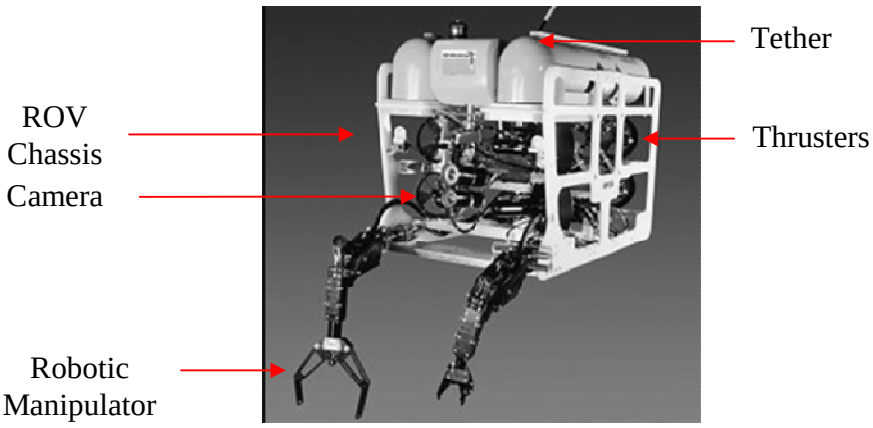


Figure 1: A typical ROVM System In such systems, the major components are tether, ROV chassis, thrusters, on-board cameras and robotic manipulators. Image courtesy of http://seaeye.com).

In many ROV applications, the teleoperated master-slave arm configuration is used to execute underwater tasks. In this configuration, the human pilot sets the desired end-effector position and orientation using a kinematically similar, but passive, master arm whose individual joint motions are duplicated by the submerged slave arm. When the ROV mobility is not needed, this strategy relies on the ROV's ability to hold station in order to emulate the stationary setting that the human pilot is situated within. In the case when the ROV mobility is also required, a second pilot, who is in charge of commanding the ROV motion whilst the first pilot drives the end-effector using the master-arm, is placed in the operation scheme. During such operations, the two pilots operate the ROV and the manipulator separately, thereby requiring vigorous coordination between the two pilots. To command the ROV and the manipulator, the pilots use two-dimensional video images, provided by the ROV's on-board cameras, to judge the required arm motions in the coming moments. A typical such teleoperation system is demonstrated in Figure 2.

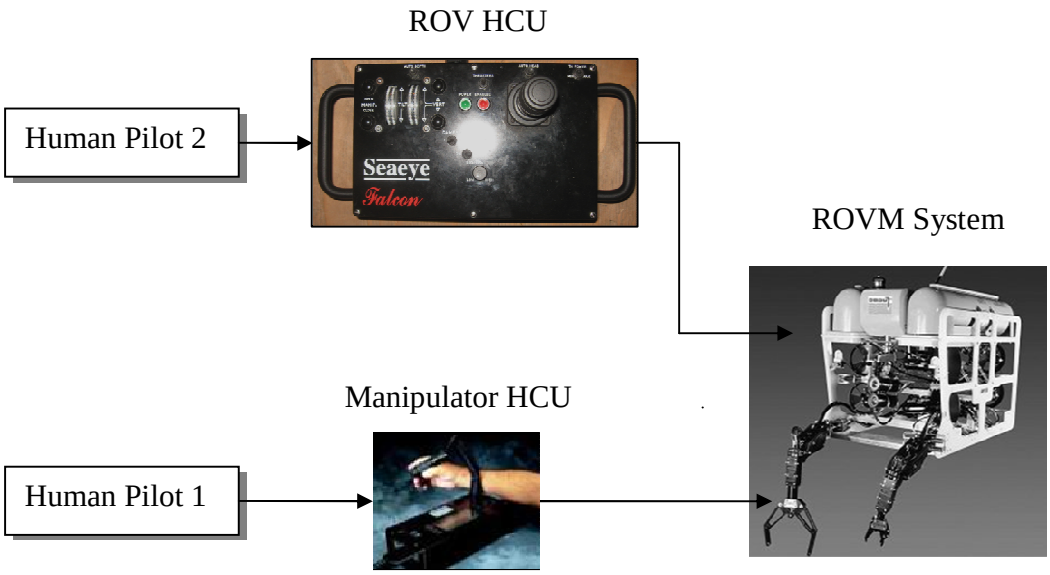


Figure 2: A typical ROVM teleoperation scheme for ROVM systems. In this scheme, there are two pilots operating the system; first pilot drives the ROV and the second pilot commands the manipulator with their respective hand control units (HCU). The ROVM System and the Manipulator HCU images are courtesy of http://seaeye.com.

1.2 Motivation

ROV systems are in general categorized into two groups based on their size and functionality; namely, work class and inspection class ROVs shown in Figure 3. Work class ROVs are very heavy and large in size, and therefore, requires specific expensive logistical arrangements for their deployment such as the rent/purchase of a surface vessel, a crane attached to this surface vessel and a large highly-qualified crew operating and supervising the ROV operations – see Figure 4 right image. Inspection class ROVs, on the other hand, are light-weight, small, and hence, highly-portable. They can be easily deployed from a wide range of inexpensive deployment settings such as the back of a small boat or a wharf without needing a large crew– see Figure 4 left image. As such, the

Figure 3: The difference in size can be seen between a work-class ROV and an inspection- class ROV. Image courtesy of http://oceanscan.net.

smaller inspection class ROVs are most desirable for shallow-depth tasks in which work-class ROVs cannot be deployed due to their size, or for tasks in which the added complexity and expenses of work-class ROVs cannot be justified due to the nature of the subsea mission and a lack of budget.

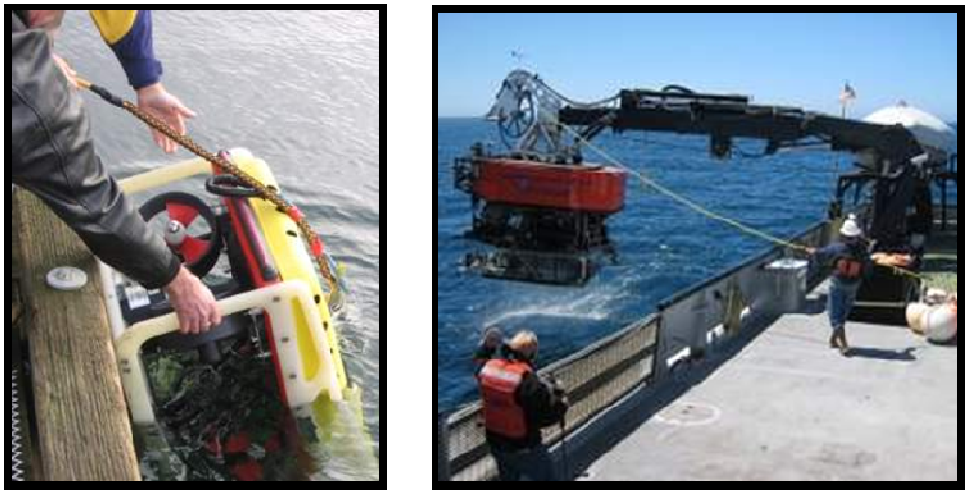


Figure 4: On the left, a work-class ROV is being hand-deployed in the water from a pier. On the right, an inspection-class ROV is being deployed by a crew with a crane attached on the surface vehicle. The image on the right is courtesy of http://polartrec.com.

Current submerged science endeavours such as VENUS [3] and NEPTUNE [4] off the Canadian Pacific coast are creating a surge of interest in underwater remotely operated vehicle-manipulator systems (ROVMs) – particularly small easily deployed variants of this class. This facility is made up of a large, widespread, network of interconnected underwater nodes that is connected to a wide range of underwater sensors. Within the scope of NEPTUNE project, one particular node where the need for small ROVM systems is apparent is the Folger Passage located at the seaward end of Barkley Sound, British Columbia – see Figure 5. Connected to this node, there are various ocean floor instruments positioned at the depth of 100m and 25m [5]. Currently, a work-class, 5-ton ROPOS vehicle, operated by Canadian Scientific Submersible, is deployed for construction, service and maintenance on these sensors – see Figure 6. However, at this shallow depth, having the option of deploying a small, inexpensive, highly-capable ROV, as an alternative to large, expensive work-class ROV would significantly reduce the mission planning and operation time as well as the prohibitive cost of deployment.

Demonstration of ROVM capabilities on large scale underwater observatories is also creating an appetite for ROVM technology in other applications such as aquaculture and ocean energy where infrastructure is installed in relatively shallow waters (< 300 m). For these shallower applications, smaller and more economic ROVMs could play a significant role in servicing submerged equipment. Unfortunately, existing small ROV platforms are inspection class vehicles, and are not equipped to complete complex and interactive submerged tasks; there are very few manipulators that are produced specifically for smaller ROV systems, and those that exist are typically under-actuated in the context of six degree of freedom (**DOF**) tasks. When an under-actuated manipulator

and the small ROV are coupled using the conventional operation scheme, one in which the ROV motion is eliminated by a second human pilot, the end-effector workspace is compressed beyond useful limits. Even if a particular task is kinematically admissible, vigorous coordination of the two pilots is required.

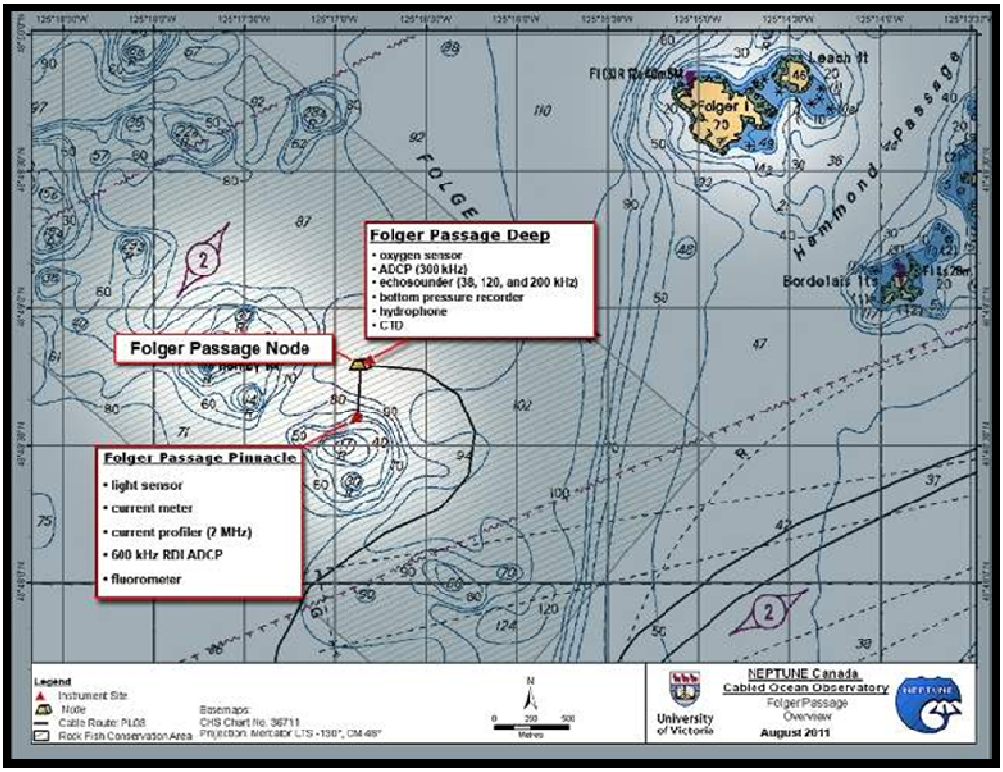


Figure 5: NEPTUNE Canada Folger Passage site map. In this particular node, the instruments are positioned in two clusters; namely, Folger Passage Deep and the Folger Passage Pinnacle with the depth ratings of 100m and 25m, respectively. Image courtesy of [5].

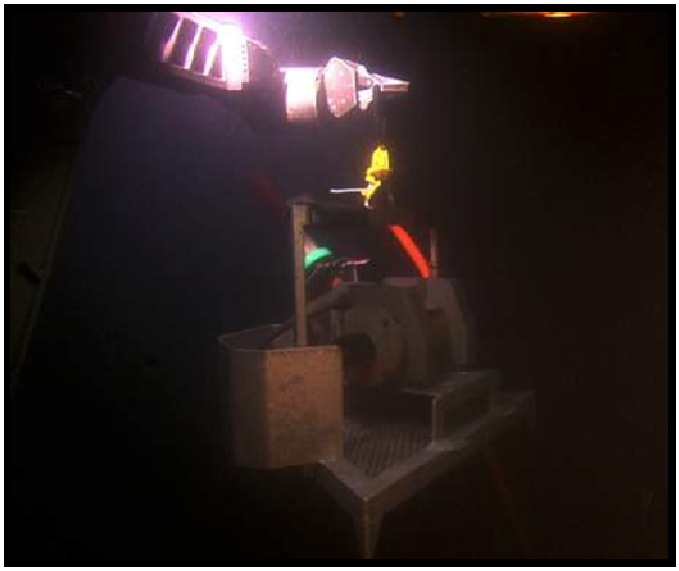


Figure 6: A snapshot from ROPOS's on-board camera. In this picture, ROPOS vehicle is deploying a bottom pressure recorder at the Folger Passage with its hydraulic arm. Image courtesy of http:// neptunecanada.ca.

To expand the range of possible tasks for small ROVM systems, the ROV and manipulator motions must be coupled so that the ROVM becomes a mobile redundant manipulator. Such automatic coordination of an ROV and a manipulator is hampered by a variety of factors including: disparate response times experienced over the different actuators in a ROVM system, reaction loads that develop at the base of the manipulator and disturb the ROV motion, and the limited feedback available to the human pilots.

During coordinated control of an ROVM, the ROV thrusters and the manipulator joints must be combined to realize a desired end effector motion. However, the ROV response will, in general, trail that of the manipulator since the ROV thrusters rely on momentum transfer to a fluid and have an inherent lag in their response to pilot inputs. This problem is more apparent for smaller ROVM systems, as the extent of the dynamic coupling is more prominent. In a conventional operation scheme, the ROV thrusters are used to

counter slowly varying tether and current disturbances, and the manipulator joints are relied upon for detailed interaction with the subsea environment. However, when the ROV DOFs are needed for an end-effector task that division is no longer possible.

When the thrusters and manipulator joints are combined properly to drive the end effector, inertial and hydrodynamic drag associated with the manipulator link motion creates reactions at the manipulator-ROV junction. The reaction loads are disturbances on the ROV motion that subsequently disturb the placement of the end-effector. As such, the reaction loads, albeit internal forces within the ROVM, can elicit a correction from the pilot that can exacerbate the problem [1], [3-6].

Moreover, the ROV motion control during the arm manipulation is compromised by the limited visual and navigational feedback available to the pilot, and the inability to sense significant disturbances being exerted by the ROV tether, the manipulator and any ocean currents. As a result, the human pilot response will change suddenly during ROVM operation, and it will be important to maintain the mobility of the manipulator so that the joints are always available to help reproduce commanded end-effector corrections.

1.3 Objectives

In order to address the difficulties outlined in the previous section, and create a viable coordinated control system for small ROVMs, the current research aims to:

1) Develop control algorithms for small ROVs that provide station keeping, waypoint following and trajectory following capabilities; using state feedback from an off-the-shelf and affordable sensor suite;

2) Develop kinematics algorithms that can automatically dissect a commanded end-effector motion into coordinated ROV and manipulator joint motions;
3) Develop control algorithms that can accurately pursue ROV and manipulator motions, commanded by the kinematic algorithms, simultaneously by properly accounting for the dynamic coupling of the ROV and manipulator;
4) Develop a novel human-machine interface (**HMI**) that can facilitate various modes of ROV/ROVM operation including direct control of the end-effector position and orientation.

To appreciate the choice of objectives for the research program, one needs to envision a representative small ROVM system performing a representative task at the ocean floor – sediment sampling for example.

In this hypothetical scenario, the task starts with the hand deployment of the ROVM system from the back of a small boat. Knowing the coordinates of the location of interest, the pilot first enters these coordinates and subsequently enables the automatic control mode using the HMI. During the *automatic-motion mode*, the ROV needs to fly, and at the same time, avoid obstacles mostly due to the uneven nature of the ocean-floor. To this end, the pilot uses the combination of *way-point* and *trajectory following modes*. First, she enters a series of waypoints defining the path and sets the mode to *way-point tracking* for obstacle-free regions and to *trajectory tracking* for sections with obstacles. When set to the *way-point tracking* mode, the ROV automatic controller guides the vehicle through way-point destinations without posing any constraints on the type of motion. When set to the *trajectory tracking* mode, however, the ROV automatic controller makes the vehicle follow a predefined trajectory, set by the pilot to avoid

known obstacles, as the vehicle traverses through the waypoints. Having arrived to the desired destination, the ROV now needs to perform the sedimentary sampling task within the vicinity of its current location.

To this end, first the pilot needs to pick up the core sampler fixed on the tool-sled and, at the same time, keeps the vehicle stationary, as the strong ocean currents push the vehicle from its current location. The pilot can achieve this by simply enabling the *station-keeping mode* in which the controller intelligently commands the on-board thrusters to combat against disturbance forces caused by the tether, manipulator and ocean current.

While the vehicle motion is fixed, the pilot now can pick up the core sampler with the manipulator. With the core sampler being in the end-effector, the ROV pilot enables the *coordinated ROV/manipulator motion mode* and starts to guide the end-effector to the sampling location by using a 6-DOF joystick.

Receiving the end-effector commands from the pilot, the *coordinated ROV/manipulator motion mode* dissects this motion into the vehicle and the arm motion in a coordinated manner while working to achieve second-level objectives that are preset by the pilot. With the coordinated control-mode being on, as opposed to the conventional methods, the pilot is no longer concerned about commanding the individual ROV and the manipulator's DOFs.

To achieve this level of functionality, the coordinated control of ROVM systems is required, which, in the context of this research, is defined as the collection of hardware and software that allows the aforementioned functionalities. Therefore, the primary focus

of this dissertation is to propose various technical building blocks that ultimately lead to the realization of such a coordinated control system for small ROVMs.

Coordinated control of small ROVMs draws upon research from four thematic areas; namely, *ROV Dynamics and Control*, *ROV Thrust allocation*, *ROVM kinematics*, ROVM *Dynamics and Control*. The following sections summarize existing work in these fields and the developments necessary to realize a coordinated control system for a small ROVM.

1.4 Literature Survey

1.4.1 Survey on ROV Dynamics Modelling and Control

An ROV controller must, using an estimate of the ROV state and a given desired state, calculate a motion that will drive the ROV towards the desired state in a stable manner and intelligently operate the on-board thrusters to obtain that motion. In the ROV paradigm, the control problem is challenged by unique factors such as:

- Inaccuracies in the ROV dynamic model caused by poor knowledge of the hydrodynamic parameters;
- A void of knowledge of some significant disturbances such as tether force, and end-effector payloads.

These constraints call for a robust controller: one that is insensitive to inaccuracies in the dynamic model of the ROV and has the strong disturbance rejection capabilities to handle the dynamic effect of the umbilical, manipulator, the ocean current, etc. Therefore, the primary focus in existing ROV dynamics and control research is on the estimation of these unknown dynamics and incorporation of these estimates into the final control law.

Traditional linear controllers cannot provide high precision motion control ability when directly used, as their performance degrades significantly in the face of highly non-linear ROV hydro dynamics and disturbance forces due to the tether, ocean current and any external effect caused by the unpredictable nature of underwater operating conditions. Therefore, more advanced control techniques are needed for reliable, high precision, effective ROV operations that can perform well under the presence of such conditions. To this end, several nonlinear control techniques have been proposed to address the control problem of ROV systems.

A scheduling of linear H_∞ controllers was applied in [6]. To this end, the work of [6] linearized the nonlinear ROV dynamics around a particular operating state vector with various payload configurations. Although shown to be a robust strategy in the presence of disturbances, this controller suffers greatly when the ROV is required to work outside its preset operating state vector. To address this issue, another level of scheduling can be incorporated into the scheme, which consists of a set of H_∞ controllers linearized around various operating vectors. However, this approach is not practical, as it is extremely difficult to cover all operating conditions especially in the ROV paradigm in which the human pilot input is indeterministic by nature.

Adaptive control techniques were also implemented to the control problem of ROV systems in [7]-[12]. However, the majority of adaptive controllers use a regressor matrix that relates the hydrodynamic coefficients to nonlinear system dynamics in a linear fashion. To obtain this linearity, many assumptions are made in the system model leading to degradation in the quality of the modeled system dynamics and in turn poorer controller performance. Those that do not use a regressor matrix do not utilize the

available knowledge of the system dynamics which could be otherwise used to obtain higher controller performance. In addition, adaptive controllers require the persistency of excitation condition to be satisfied for a reliable implementation. This condition indicates that the content of input signal is rich enough to excite all parameters in the system so that the adaptation algorithm can converge to the true values of the estimated parameters. When this condition is not satisfied, the "parameter drift" problem most often occurs, leading to instability in the controller behaviour [13]. In the aforementioned adaptive ROV control works, this problem was not properly addressed, or even reported.

Another common ROV controller draws upon the sliding-mode control theory, which was applied to the ROV paradigm in works [14]-[16]. As demonstrated in these works, the sliding-mode control is indeed an effective way of controlling ROV systems due to its insensitivity to imprecisions in the system dynamics model. However, this insensitivity comes at the expense of discontinuous, high frequency control activity cited as "chattering" in the relevant literature. The chattering phenomena makes this approach less desirable since it could not only excite the unmodeled system dynamics, in turn causing instable system motion, but also it could lead to premature wear-out in the thrusters. In addition, this type of control activity causes high heat dissipation in power circuits that could be harmful to the electrical circuits. To reduce chattering, the common approach is the boundary layer approach [14]-[17]. The boundary layer approach makes the control activity continuous within the boundary layer and discontinuous outside of the boundary layer. Time-varying version of the boundary layer approach was proposed in

[18]. Although effective, the boundary layer approach requires a trade-off between the controller performance and the degree of reduction in chattering.

Fuzzy-logic controllers were also adapted to the ROV paradigm in [19] and [20]. The fuzzy logic works based on a set of linguistic rules that reflect the knowledge of a human pilot and could be effective approach. That being said, this approach alone is not adequate for tasks where precise and accurate ROV motion is required and needs to be used with other control methods.

Neural network methods were also used for the control problem of ROV systems such as in [21]-[23]. Neural-networks are very effective tools in estimating unknown dynamics due to their universal approximation and learning capabilities. However, when used alone, the neural network controllers require extensive amounts of training data to achieve the desired estimation performance, which is not practical due to the random nature of ROV applications.

1.4.2 Survey on ROV Thrust Allocation

A control law produces a measure of the generalized force, a force and a moment, that is needed at the ROV center of mass. The generalized force must then be realized by the arrangement of on-board thrusters. To ensure manoeuvrability, the thruster arrangement is redundant; there are more thrusters than there are active vehicle degrees of freedom. Due to the excess number of thrusters, there are an infinite number of ways to allocate the controller's command. The field of thrust allocation deals with the generation of mathematical criteria that can be applied to automate the selection of one particular solution

A prominent approach to the thrust allocation problem is the 2 norm (l_2 norm) based solution in which the sum of the squares of the individual thruster forces is minimized. In [11], a pseudo-inverse solution, and in [24]-[25] , a weighted pseudo-inverse is used to generate an optimal distribution of a commanded generalized force. The pseudo-inverse method has the advantage of being relatively simple to compute. Pseudo-inverse solutions correspond to the minimization of either the l_2 norm or a weighted l_2 norm of the thrust manifold. The pseudo-inverse solution was also used for the thruster allocation problem in [26]. To generate reference thruster values that do not exceed the saturation limit of each thruster, in [26] a dynamic state feedback method was employed.

However, the pseudo-inverse method (l_2 norm minimization) does not afford easy implementation of thruster saturation limits [25]. It was reported in [27] that, even if thruster saturation is implemented, the pseudo-inverse solution is not guaranteed to satisfy the saturation constraints. Furthermore, the l_2 norm-based solution does not necessarily minimize the magnitudes of the individual thrusts, and can generate thrust demands that may exceed an individual thruster's saturation point. In addition, there could be an unequal distribution in the thrust manifold leading to a relatively high thrust demand for a particular thruster. Also, there exist solutions where minimizing the l_∞ norm gives feasible solutions whereas minimizing the l_2 norm does not. This is due to the fact that the l_∞ norm provides the exact representation of the feasible thrust solution space whereas the l_2 norm provides an approximation of the feasible solution space. In such cases, there exists a potential for a loss of manoeuvrability on subsequent control steps.

1.4.3 Survey on ROVM Kinematics

ROVM systems are an example of a mobile redundant manipulator. Redundant systems possess more degrees of freedom (**DOFs**) than those required to execute a given task. In a typical ROVM system, in addition to the manipulator's DOFs, the ROV itself contributes 6 active DOFs, including surge (forward), sway (lateral) and heave (vertical) translations, and a yawing rotation about the vertical axis. Due to the extra DOFs, the ROVM system admits an infinite number of joint-space solutions for a given end-effector position and orientation. The available redundant DOFs can be used to achieve additional, or secondary, objectives while the given end-effector motion, the primary task, remains uncompromised. The study of ROVM kinematics deals with the creation of secondary objectives that make the solution for the joint motions deterministic and also make the pilot's job easier in the coming moments.

The implementations of redundancy resolution methods to the ROVM systems have been documented in only a few existing works as pointed out in [28]. The singularity robust task-priority redundancy resolution [29], which was originally proposed in [30] and [31], was shown to be useful for a ROVM in [32] due to its multitask capabilities. In [32], the secondary task was set to hold the vehicle stationary, and the manipulator singularity avoidance was realized by constraining joint motion ranges. However, this limits the available workspace of the manipulator. In [33], the kinematic redundancy is utilized to minimize the total hydrodynamic drag forces experienced by a ROVM system in an effort to reduce energy consumption. However, the different dynamic characteristics of the ROV and the manipulator were not addressed. In [34], the singularity robust task-priority redundancy resolution is merged with a fuzzy technique to resolve the ROV-manipulator coordination. It was shown that the fuzzy method provides

a versatile tool to handle multiple secondary tasks. However, the proposed scheme does not provide a means to hold the vehicle motionless when the commanded end-effector location is within the reach of the manipulator. A unified dynamics-based motion planning method that can produce both kinematically admissible and dynamically realizable joint-space solution was proposed in [35]. The proposed scheme takes into account the different dynamic characteristics of the ROV and manipulator, and resolves the redundancy accordingly. A screw theory method, combined with the Davies method to represent the differential motion, was implemented to solve the inverse kinematics of ROVM systems in [36]. The minimization of the ROV motion was realized by imposing kinematic constraints on velocity magnitudes. However, the singularity problem was not addressed in the same paper. The problem of eliminating unnecessary ROV motions while avoiding manipulator singular configurations was addressed in [37], where a fuzzy hybrid system was proposed as a solution to this problem. However, the method proposed in [37] requires complex fuzzy rules for guiding the ROV motion in a manner consistent with the predefined "hysteresis" behaviour.

In order to fulfill the primary and secondary objectives, the redundancy resolution scheme must allow the incorporation of multiple criteria. Among all these works, only [34] addresses the issue of multiple redundancy resolution criteria in ROVM systems in detail. However, the task-priority redundancy resolution approach used in [34] requires predefined secondary task values at each sampling time, which may not be available for on-line underwater tasks. Furthermore, [34] did not account for faulty joints, and excessive joint velocities. These deficiencies make the scheme short of being complete for complex ROVM applications.

1.4.4 Survey on ROVM Dynamics and Control

While the redundancy resolution scheme translates the pilot's intent into ROVM joint rates and provides reference values, it falls on a separate robust control strategy to realize these reference joint rates. The following challenges are introduced by the presence of the manipulator:

- Increased complexity of the mathematical model used in the calculation of the generalized force needed at the ROV center of mass and the manipulator joint torques;
- Uncertainty in the dynamic model of the manipulator, mainly due to the poor knowledge of the hydrodynamic parameters;
- Dynamic coupling between the vehicle and the manipulator;
- Differences in the bandwidth between the ROV and the manipulator actuators.

Similar to ROV controllers, ROVM controllers must perform well in the presence of inaccuracies in the ROVM dynamic model. Research in ROVM control is focussed on on-line techniques to estimate the unknown dynamics and use these estimations in the calculation of needed actuator inputs.

Controller development has been largely applied to ROVs, and it is rare that the manipulator degrees of freedom are considered. The unified control of ROVM systems has been addressed in [38]. In [38], an adaptive controller for an ROVM was designed. A robust control technique based on the singular perturbation method was demonstrated in [39]. The computed torque control was applied in [33] on an ROVM system. A model-based sliding mode controller was proposed in [11]. In [40], a sliding-mode controller with a Fuzzy logic based tuning of the control gains was designed.

In these methods, as mentioned in Section 1.4.1, the adaptive control methodology requires deriving a regressor matrix. This step requires linear parameterization of the hydrodynamic coefficients, which is difficult to realize due to the nonlinear nature of the system dynamics. Due to this difficulty, further assumptions are necessary in the system model leading to further degradation in the quality of the modeled system dynamics and in turn poorer controller performance. In terms of the computed torque method, it requires exact knowledge of ROVM system, which is extremely difficult to obtain.

It has been shown that the model-based sliding-mode approach is an effective means of controlling a ROV, largely due to its ability to tolerate imprecision in the dynamics model [14], [15], and [41]. As mentioned in Section 1.4.1, the direct implementation of a sliding-mode controller causes chattering. As with ROV systems, to eliminate or reduce chattering, the boundary layer method could be used. However, as noted before, when the boundary layer approach is used, the robustness property of the sliding mode control is compromised for the sake of chattering elimination since sliding-mode controllers act similarly to PD controllers within the boundary layers.

1.5 Methodology

In this section, low-level objectives in each thematic field are defined. These low-level objectives are determined based on the shortcomings of each thematic field mentioned in Section 1.4 and concurrently outlines the necessary steps towards the realization of the high level objectives listed in Section 1.3.

1.5.1 ROV Dynamics and Control

In ROV control literature, the adaptive control and the sliding-mode control strategies are widely used due to their robust nature to unknown dynamics. As mentioned in

Section 1.4.1, the adaptive control requires the derivation of a regressor matrix which requires the assumption of linearity in the hydrodynamic parameters. This assumption is hard to justify since the existing models contains nonlinearity in the hydrodynamic parameters. Therefore, existing adaptive controllers do not allow the use of a detailed dynamics model. With regards to the sliding-mode approach, the existing methods utilize a boundary layer approach to eradicate the chattering problem. However, this approach eliminates the robustness property of the sliding-mode control approach, as the controller acts as a PD controller within the boundary layers. In addition, the controller to be designed must be robust to the parameter drift problem that often occurs when the persistency of excitation condition is not satisfied. Therefore, the following technical objectives will be pursued for the ROV dynamics and control:

1) Develop ROV simulation platform(s) to be used as a test bed for the motion controllers to be designed;
2) Develop a ROV control that can blend the advantages of the adaptive theory with the advantages of the sliding-mode theory while eliminating their pertinent disadvantages with robustness property to the parameter drift problem;
3) Implement the most promising controller on an inspection-class Falcon™ ROV. This implementation involves the following steps:
    a) Designing a new modular navigation skid that uses a blend of Doppler Velocity Log, compass, inertial measurements and acoustic data;
    b) Designing and tuning Extended Kalman Filter for use in the controller;
    c) Extracting the dynamic model of the Falcon with field experiments;
    d) Implementing the controller and collecting field data.

1.5.2 ROV Thrust Allocation

As a survey of the literature in Section 1.4.2 has revealed that the prominent approach to the thrust allocation scheme is the l_2 norm solution. However, there are some disadvantages of the l_2 norm solution. For instance, it may cause excessive thrust values, which may lead to controller failure due to the thruster saturation limits. Also, the l_2 norm method does not afford easy implementation of thruster saturation limits. Furthermore, since the l_2 norm seeks for a solution within the space defined by the l_2 norm, it does not cover all feasible solution points. Also, the scheme must be fault-tolerant for reliable thrust allocation applications. Thus, significant contributions can be made by:

1) Developing an effective thrust allocation scheme that utilizes the redundancy in the thruster lay-out for better manoeuvrability;
2) Making the derived scheme fault-tolerant and capable of eliminating the complications associated with the l_2 norm solution.

The proposed research objectives will foster the development of future improvements and modifications to the ROV systems.

1.5.3 ROVM Kinematics

Since the redundancy resolution replaces the human intervention in the vehicle and manipulator motion, it is vital for the redundancy resolution to accommodate all possible scenarios that might occur during ROVM applications. As a survey of the literature in Section 1.4.3 has revealed, the existing literature is far from being complete in this regard. For instance, there is no redundancy resolution scheme for ROVM systems that

allows the inclusion of multiple objectives with a fault-tolerant property. Also, the existing schemes do not consider velocity limits on the joints. Moreover, there is no work reported related to the interfacing of the redundancy resolution technique with a 6-DOF hand control unit with real-time implementation methodology. Furthermore, there is no detailed user-machine interface scheme for detailed ROVM operations in the existing literature. Thus, this research aims for:

1) Developing a redundancy resolution scheme that accommodates multiple objectives;
2) Orchestrating these competing secondary objectives in an optimal, coordinated manner without generating any excessive joint velocities and fault-tolerant property;
3) Interfacing the derived redundancy resolution with a 6-DOF joystick performed to evaluate the performance of the proposed redundancy resolution schemes for a broad range of ROVM end-effector inputs;
4) Developing a user-machine interface for effective ROVM operations using the proposed redundancy scheme.

1.5.4 ROVM Dynamics and Control

In terms of ROVM control literature; there is only a few works that directly deal with the ROVM control. Similarly to ROV systems, the widely accepted approaches to the control problem of ROVM systems are the adaptive control and the sliding-mode control strategy due to their success in dealing with inaccuracies in the system model, and hence were chosen for the current work. Since the adaptive control and the sliding-mode control methodologies will be employed, the problems defined in Section 1.5.1 applies to

the ROVM control problem tackled in this work. However, given the existence of the manipulator in ROVM system, the following research aims are added to those of Section 1.5.1:

1. Deriving computationally efficient dynamic modelling strategy for the ROVM dynamic modelling that includes the manipulator.
2. Deriving unified control methodologies for effective ROVM system control that addresses the challenges outlined in Section 1.4.4.

The proposed research objectives will foster the development of future improvements and modifications to the ROVM systems.

1.6 The Experimental ROVM platform

The experimental ROVM platform used in the experimental portion of this dissertation is an inspection class, Saab-Seaeye Falcon™ ROV provided by Suboceanic Sciences Ltd with a 4-DOF Hydraulic HLK43000 manipulator produced by Hydrolek Ltd as shown in Figure 7 and Figure 8, respectively. In terms of the Falcon™ ROV, its main components are the vehicle, tether, and surface station as shown in Figure 7. The vehicle is designed to dive up to 300m deep and has 75kg mass. The Falcon™ ROV has 5 electric thrusters; the first four thrusters are for the vehicle's horizontal plane motion, the last one is for the vertical motion in the water column. The forward, lateral and heave motion of the vehicle are controlled by moving the joystick on the Falcon ™ HCU as shown in Figure 9 (left). In terms of the on-board navigational components, the system is equipped with compass for heading measurement, rate gyro for pitch and roll angles, and finally a pressure sensor for depth measurement. In addition, the system has passive sonar that is used to detect nearby objects in the vehicle's horizontal plane.

Figure 7: Inspection-class SeaEYE Falcon™ ROV systems. Due to their one to one ratio between its weight and trust, these systems are commonly used in industry. This system was used as a test-bed for the experimental portion of the current work. Image courtesy of [43].

Figure 8: HLK 43000™ 4-DOF hydraulically powered underwater manipulator produced by Hydrolek Ltd. This arm is designed specifically for inspection class ROVs and can be easily mounted to the Falcon ROV.

Figure 9: On the left is the Falcon™ HCU that allows the human pilot to operate the vehicle. On the right is the HCU of the HLK 43000 underwater manipulator. By pushing/pulling the corresponding joystick on the manipulator HCU, individual joints can be moved. Currently, the manipulator HCU allows commanding only one joint motion at a time.

Additionally, the Falcon™ ROV employed in this work is equipped with a 350m long fiber-optic tether that is used to transmit power and bi-directional telemetry between the surface station and the vehicle. To communicate with each on-board device, the Falcon™ system implements RS485 multi-drop network in which each node contains a micro controller that receives/sends commands/replies from/to the master node over the network.

Finally, the surface station contains power, surface control unit (**SCU**) and monitoring devices for the pilot to operate the vehicle safely from a surface vessel. The power unit transmits 2KW power to down the vehicle. The SCU contains a PC104 stack by which the master node is programmed. The master node receives commands from the HCU and transmits the commands to the targeted nodes one at a time over the RS485 network and receives replies from the corresponding nodes. The HCU allows the pilot to control the on board thrusters, lights and the tilt motor of the on-board camera through the surface

control. The sonar display indicates the nearby objects and finally the video screen provides visual information to the pilot as well as the video overlay indicating the vehicle's orientations.

As for the HLK 43000™ underwater manipulator, it provides additional 4DOF to the Falcon™ ROV. It has a hydraulic pump mounted on the manipulator skid that provides pressurized hydraulic fluid to each hydraulic cylinder in the system. Through the on-off solenoid valves located in the on-board valve-pack, the flow of the hydraulic fluid can be controlled via a HCU. When a small joystick on the HCU is pushed up/pulled down, the hydraulic fluid starts to flow/discharge into/from the corresponding hydraulic cylinder; causing the relevant joint to move. With the existing HLK 43000™, only one joint can be commanded at a time. The new version of this arm that allows the simultaneous motion capability could never be delivered within the timeline of this research, and thus, the evaluation of the ROVM kinematic coordination schemes and controllers was contained to numerical simulations in this dissertation.

To provide reference end-effector motion for the evaluation of the coordinated control of ROVM systems, the RSI joystick [42] shown in Figure 10a was used. The RSI joystick has a parallel architecture that is composed of two platforms (the base and the handle), which are connected with three symmetric branches. Platforms and links are sequentially connected with joints, six in total as shown in Figure 10b. The minimum number of transducers that the RSI hand controller requires is six. Nevertheless, there would be up to 16 solutions to the forward displacement problem. To eliminate this uncertainty, Notash and Podhorodeski [44] analyzed the forward displacement problem of the RSI joystick and concluded that with three additional sensors, there is a unique

solution to the forward displacement problem. As a result, the joystick is instrumented with nine transducers, namely rotary Midori™ CP-2FB potentiometers. To digitalize the output voltages of the potentiometers, the DAQ card is employed.

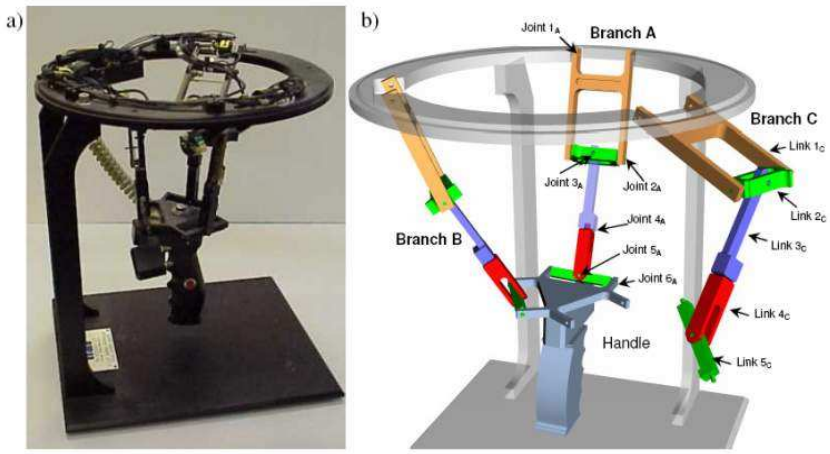


Figure 10: RSI Parallel-architectured joystick was used to generate reference values for the end-effector motion. In the ROVM coordination work, this joystick replaces the HCU of the Falcon™ ROV and the HLK 43000 shown in Figure 9. Image courtesy of [45].

1.7 Thesis Outline

The remainder of the dissertation proceeds as follows: Chapter 2 summarizes the motivation, methodology and the contribution of each peer-reviewed publication and submitted work completed in the field of ROV Dynamics and Control with emphasis on the relation between each work and the objectives outlined in Section 1.3. Chapter 3 summarizes the motivation, methodology and the contribution of each peer-reviewed published and non-published work done in the field of ROVM Kinematics, ROVM Dynamic and Control and ROVM operation. Finally, Chapter 4 provides a brief summary of the overall contributions and enumerates avenues for future work for development.

CHAPTER 2. SUMMARY OF CONTRIBUTIONS TO ROV DYNAMICS MODELLING AND CONTROL

2.1 Chapter Review

In this chapter, technical building blocks toward the completion of Objective #1 of Section 1.3 are summarized in the context of the coordinated control of the Saab-Seaeye Falcon™ ROV. As detailed in Section 1.3, the coordinated control of ROVM systems requires the following ROV functionalities: automatic *Station Keeping*, *Way-Point Tracking* and *Trajectory Following*. While the objective of a station keeping control is to stabilize the vehicle around a desired set point, including position and orientation, the objective of a way-point controller is to track position commands by making the position error vector converge to zero. With regards to the trajectory following controller, the objective is to follow time-dependant, desired positions and velocities of a trajectory. With these functionalities, the human pilot can not only guide the ROV motion with high precision but also can make the ROV automatically follow a preset route with precise position and velocity directives.

In Section 2.2, different station-keeping controllers developed in the pursuit of Objective #1 of Section 1.3 are summarized. Likewise, in Section 2.3, methodologies developed for the way-point and trajectories following tasks are summarized.

2.2 ROV Control: Station Keeping

The major goal in station keeping controller design for ROV systems is to develop a controller that stabilizes the vehicle around a desired set point while compensating for the disturbance forces acting on the ROV. In the ROV paradigm, these disturbances often arise from the dynamic interactions between the vehicle and the on-board manipulator as well as the tether. As such, estimating these disturbances and subsequently integrating them into the controller design are at the core of any station keeping controller design.

To devise an effective station keeping controller, a methodology is summarized in Section 2.2.1. This method provides a means to intelligently compensate for the disturbance forces whilst holding the vehicle stationary. In the development phase of the station keeping controller, the dynamic model of an ROV was used without explicitly defining the dynamics of the disturbance forces. Instead, these disturbances were accounted for through weighting functions, which impose closed-loop performance specifications based on the expected disturbances defined in the frequency domain. This controller may fail, however, if disturbances do not agree with the predicted frequency domain specifications.

In pursuit of a controller that can work for a broader range of disturbance conditions, instead of frequency domain projections of the manipulator and tether disturbances, explicit force domain expressions were incorporated into the controller summarized in Section 2.2.2. To this end, the ROV, manipulator and tether were modelled together as a

whole. However, once the ROVM plus tether model were used, the system dynamics could no longer be linearized around a specific set point, as the manipulator and tether motion and their influence on the ROV dynamics are highly nonlinear. Thus, a new control methodology that does not require the linearization of the overall dynamic model was proposed in Appendix-B and is summarized in Section 2.2.2.

2.2.1 Station Keeping without Explicit Disturbance Knowledge [A]

In this section, the work entitled "*MIMO Sliding-Mode and H∞ Controller Design for Dynamic Coupling Reduction in Underwater-Manipulator Systems*" is summarized. The controller developed in this work uses the expected disturbance knowledge defined in the frequency domain in its control law derivations. For a detailed presentation of the technical developments, the reader is referred to Appendix-A.

Motivation: In roughly 35% of ROV underwater missions, the ROVs are required to hold station [12]. In such applications, ROVs provide a stationary platform while the manipulator performs a required task. However, during task execution, the torques commanded at the manipulator joints lead to reactions at the junction point of the manipulator that significantly disturb the vehicle. Whether the ROVM is being piloted by one individual or two, these manipulator induced disturbances are difficult to anticipate and compensate. Therefore, these reaction forces must be mitigated automatically and on-the-fly, using the on-board thrusters to achieve high performance station keeping.

Methodology: In Appendix-A, the Articulated Body Algorithm (**ABA**) [46]-[47] was used to model the dynamic coupling effect of the manipulator. In order to mitigate the dynamic coupling effect for an effective station-keeping task, a new approach was

proposed that used the combination of two model-based controllers. The two model-based controllers employed therein were a multi-input multi-output **(MIMO**) sliding-mode controller and an H∞ controller [48], respectively. These two controllers were arranged in a cascading structure, as shown in Figure 8-A[1]. To design a sliding-mode control, the work of Antonelli [11] was implemented based on a generic ROV system dynamic model defined in Eq. (1)-A. Using this model, the Lyapunov theory yielded Eq. (7)-A as the control law equation. To design an H∞ controller, Eq. (1)-A was linearized around the operation point of **0** for the station keeping task, and the state-space formula of the linearized system, Eqs. (3-4)-A, was obtained for use in the H∞ controller design. The closed-loop controller performance was defined in the frequency domain using the shaping filter of Eq. (14)-A for lower frequencies. Likewise, the degree of uncertainty due to the disturbances as well as the parametric inaccuracies in the system model was quantified through the shaping filter of Eq. (15)-A for higher frequencies. Finally, using the linearized model and the shaping filters, the mixed sensitivity function [48] was defined and the optimization of this function was performed using Matlab™'s *LMI Control Toolbox* [49].

Results: To evaluate the performance of the proposed controller, time-domain numerical simulations were performed. Although different than the targeted Falcon™ ROV system in this dissertation, the Canadian Scientific Submersible Facility (**ROPOS**) vehicle [50] was considered in the case studies since the dynamic parameters of the ROPOS vehicle were available at the time of the current research, unlike those of the Falcon™ ROV.

[1] The notation, "Figure 8-A" refers to "Figure 8" of the article contained in " Appendix-A"

In the simulation case studies, the station keeping performance of the sliding-mode control, the H∞ control and the combined H∞-sliding-mode control, were reported. In these case studies, the presence of the dynamic coupling effect on the ROV due the manipulator motion as well as the 20% parametric uncertainties in the ROV dynamic model were considered. It was found that the sliding-mode controller provides slightly better performance in compensating for the dynamic coupling effect in comparison to the H∞ control. It was also found that a significant improvement over the sliding-mode and H∞ controller can be obtained when both approaches are combined in a cascading structure. The case study revealed that the combined H∞-sliding-mode controller improves the position and orientation error as much as 35% and 47%, respectively, compared to the sliding-mode controller alone. Compared to the H∞ controller alone case, the improvements were found to be 49% and 49%, for the position and orientation respectively, as demonstrated in Figure 10-A. This improvement can be attributed to the fact that the low-level sliding-mode controller worked to drive the state of the nonlinear ROV system towards the equilibrium state of **0.** As such, the high-level H∞ controller acted on a system whose dynamics model was more consistent with the linearized model used in the design of H∞ controller.

Contributions: The article presented in Appendix-A contributed:

1) Two independent robust controllers that are each capable of reducing disturbances on ROV motion.
2) A new cascading control strategy that wraps an H∞ controller around a sliding-mode controller to yield a significant increase in station-keeping performance.

2.2.2 Station Keeping with Explicit Disturbance Knowledge [B]

In this section, the work entitled "*Dynamics and Control of Tethered Underwater-Manipulator Systems*" is summarized. In this work, a station-keeping controller was proposed that utilized the explicit tether and manipulator disturbance knowledge. For a detailed presentation of the technical developments, the reader is referred to Appendix B.

Motivation: The controller presented in Appendix-A could lead to unstable controller behaviour if disturbances do not agree with the shaping filters. To overcome this shortcoming, explicit knowledge of the most prominent disturbances is needed.

To provide power and telemetry to the vehicle and the robotic manipulator, an ROV is physically connected to the surface support vessel by a tether. Unfortunately, the tether can dominate the system dynamics and compromise the ROV task at hand [51], which often relies on station keeping the ROV. Likewise, the manipulator motion also has a dominant disturbance effect on the ROV behaviour at the ROV-manipulator interface.

To account for these dominant disturbances, the tether and the manipulator dynamics need to be included in the dynamics model. With the inclusion of the manipulator's dynamic effect, however, there is no longer a valid single operating point about which the overall system dynamics can be effectively linearized. This is because the manipulator states are constantly changing with the pilot operating the manipulator. As for the tether disturbances, they can perturb the system dynamics from the linear model and thus make the linear assumption invalid due to their. Thus, the controller of Appendix-A cannot be used with the extended system dynamics, and therefore, a new controller strategy is needed. In light of this, the primary objectives of Appendix-B were:

- To include the tether dynamics as well as the manipulator dynamics in the ROV model and to explore the extent of these forces on the station-keeping task;

- To design a controller for the station-keeping task that can accommodate the new nonlinear model.

Methodology: In Appendix-B, a complete simulation scheme, including the tether, ROV, manipulator and tether dynamics was presented. A tether dynamic model of Eq. (2)-B, originally developed in [52], was implemented. This model uses a lumped mass approach in which the ROV tether is considered to be a series of point masses connected by linear, massless, visco-elastic springs. In addition, the model accounts for the tether bending and twisting effects, which are crucial to be able to accurately simulate the ROV system during instances of low-tension. The dynamics of the tether on the ROV were defined through Eqs. (2-7)-B.

The ROV and manipulator dynamics of Eqs. (5-6)-B were obtained using the ABA presented in [53]. The ABA was used due to its computational superiority over other existing dynamic modelling approaches. Using the ABA approach, the dynamics of the ROV and manipulator were unified. To link the tether dynamics both with the ROV and the manipulator dynamics, Eq. (8)-B was used that represents the resulting tether forces and moments at the ROV's center of mass.

For the station-keeping task, a series of Single-Input Single-Output (**SISO**) sliding mode controllers were designed. To reduce the complexity brought by the detailed tether and the manipulator modeling, the SISO approach was preferred over the MIMO approach in the controller design problem. Using the sliding-mode theory outlined in [54], the control law signal of Eq. (24)-B was derived. The resulting control law has controller gains that are a function of the time-varying articulated inertias, and, as a result, it provides a means to update otherwise static controller gains.

Results: To evaluate the efficacy of the proposed controller in the station-keeping task, a number of time-domain numerical simulation case studies were performed. The tethered ROVM considered in this work was the Falcon™ ROV with the Hydrolek™ HLK43000 manipulator. In order to model uncertainties in the controllers, the drag and added mass coefficients of the FALCON™ and the Hydrolek™ manipulator differed by 40% from the real system.

In the second case study, the controller was commanded to keep the vehicle stationary while the tethered ROVM system was pushed by the ocean current. During the simulation, only the manipulator disturbance effect was accounted for in the controller signal. As Figure 10-B revealed, the largest errors were found to be approximately 0.15m and 0.08m along the X and Z axis, respectively. In the third case study, unlike the second, the tether disturbance knowledge was directly incorporated into the controller signal, and the largest errors were recorded 0.063m and 10^{-3}m along the X and Z axis, respectively, as illustrated in Figure 11-B. In conclusion, the addition of the tether disturbance knowledge into the controller reduced the maximum Euclidian-norm of the position error from 0.1655m down to 0.047m, a71% decrease, as can be seen in Figure 12-B.

Contributions: The article presented in Appendix-B contributed:

1) A strategy to incorporate the tether dynamics into the ABA equations that yields a complete computational scheme for the time-domain simulation of a tethered-ROV with manipulator;
2) A series of SISO sliding mode controllers that command the on-board thrusters in such a way that the tether and manipulator-related disturbance forces are

balanced out. The derived sliding-mode controller synthesis differs from conventional sliding-mode controllers in that the control gains are defined as a function of the time-varying articulated inertias instead of the static values obtained through trial and error;

3) A conclusion that states that the controller's performance can be increased by a relatively significant degree through introducing the tether disturbance knowledge to the sliding mode controllers.

2.3 ROV Control: Trajectory Following

In addition to the automatic station-keeping capability, another functionality sought for the coordinated control of ROVM systems is the ROV's ability to automatically traverse a preset route, as detailed in Section 1.3. This functionality would allow the human pilot to perform tasks that require high-precision position control of the ROV. In addition, this functionality could also be used in assistive mode to complement the human pilot intervention and improve the pilot's ability to perform high precision underwater tasks.

In Section 2.3.1, the work of Appendix-C is summarized. In this work, a chattering-free adaptive sliding-mode controller was presented for the trajectory problem of ROV systems. Conventional sliding-mode controllers have a discontinuous term that provides robustness against nonlinearities in the system as well as inaccuracies in the system dynamics estimation. While effective, these controllers cause high frequency control activity due to the inherent discontinuous control manner of this controller, which is harmful to the system. In order to eliminate chattering, Appendix-C adapted the approach presented in [55], [56] to the ROV paradigm. This approach avoids the chattering problem by replacing the discontinuous term with a continuous, adaptive one,

thereby eliminating the source of this problem. This continuous term constantly estimates the difference between known and unknown dynamics (lumped uncertainty) [57] and acts as a corrector on the final control signal. Numerical simulation of the Falcon ™ ROV showed the viability of this controller for the trajectory control of ROV systems. However, the asymptotic stability of this controller is harmed, especially when the ROV moves suddenly – see Section 3.3.2 for details. In addition, this controller is prone to the parameter drift problem that often occurs when the persistency of excitation condition is not satisfied.

In Section 2.3.2, the work of Appendix-D is summarized. In this work, a methodology that builds on the work of Appendix-C was proposed. The asymptotic stability requirement of Appendix-C were relaxed by using a continuous, adaptive term that constantly estimates the upper bound on the two norm of the lumped uncertainty vector in lieu of the lumped uncertainty vector itself. With this upgrade, the Lyapunov stability is no longer directly dependent on the time derivative of the lumped uncertainty vector and the controller provides asymptotic stability even when the ROV moves suddenly. In addition, the parameter drift problem was eliminated using a leakage term [58] based on a variant of the σ-modification method [59]. However, with this addition, the controller is no longer asymptotically stable and is only uniformly ultimately bounded [60]. Yet, better controller performance is expected compared to the chattering free controller of Appendix-C with the same parameter drift protection since the new controller provides smaller lower bound on the error metric. An adaptive PID layer with a novel adaptation law was also incorporated to further enhance the controller performance. Experimental validation of this approach on the Falcon™ ROV platform was also made via sea-trials.

2.3.1 ROV Control with Estimation of Lumped Uncertainty [C]

In this section, the controller design portion of the work entitled "*Chattering-Free Sliding-Mode Controller for Underwater Vehicles with Fault-Tolerant Infinity Norm Thrust Allocation.*" is summarized. In this work, a trajectory following controller was presented for the trajectory control problem of ROV systems. For a detailed presentation of the technical developments, the reader is referred to Appendix-C.

Motivation: The standard sliding-mode control law is in the form $\boldsymbol{\tau} = \boldsymbol{\tau}_{eq} + \boldsymbol{\tau}_{sw}$ where $\boldsymbol{\tau}$ corresponds to a generalized force acting at the ROV's centre of mass, and $\boldsymbol{\tau}_{eq}$ and $\boldsymbol{\tau}_{sw}$ symbolize the equivalent control law and the switching control law, respectively. The equivalent control law is continuous and model based. In the absence of uncertainties in the system dynamics, this equivalent control alone suffices to realize the desired dynamics. The switching term is a discontinuous feedback component that is in charge of compensating for deviations from the desired dynamics, and therefore is the source of the robustness of the sliding-mode control law. The switching term acts on the system in a bang-bang manner creating chatter in the actuators. This is problematic for the following reasons:

- This type of chatter in the actuators causes the system state to oscillate intensely across the sliding manifold, which in turn leads to high heat losses in electrical power circuits and premature wear in actuators;
- The high control activity may excite unmodelled high frequency dynamics, which in turn may cause controller performance degradation.

Therefore, the main objective of this paper is to eliminate the aforementioned complications regarding the standard sliding-mode control.

Methodology: The controller in this study adopted an adaptive term $\boldsymbol{\tau}_{ad}$ given in Eq. (14)-C with the adaptation law of Eq. (15)-C in place of the conventional, discontinuous switching term to eliminate the chatter of a conventional sliding-mode controller for ROV applications. The adaptation law of Eq. (15)-C was proposed in [61] to estimate the lumped uncertainty vector in the control of linear inductance motors. This adaptation law was also proposed in [55] to eliminate chattering and was implemented to the trajectory control problem of parallel manipulators in [56], [62] and [63].

The adaptive term of Eq. (15)-C compensated for the unknown system dynamics caused by poorly approximated non-linear hydrodynamics or sudden environmental loads in a continuous manner. The justification of the adaptive function of Eq. (15)-C is shown in Figure 10a-C and 10b-C, respectively. In these figures, it can be seen that the estimation function successfully predicted the unknown lumped uncertainty vector, as the difference in real and estimated values became gradually small, as demonstrated in Figure 10c-C.

Results: Numerical simulation studies were performed on the Falcon™ platform. In these studies, the vehicle was commanded to follow a predefined trajectory with the proposed controller to assess the performance of the controller. In terms of eliminating the chatter, the controller demanded continuous forces and moments given the continuous nature of the adaptive term and reduced the chattering significantly, as shown in Figure 8-C.

A comparison study against Slotine's boundary layer approach [54] to avoid chattering was also performed in Appendix-C. As shown in Figure 9-C, the chattering-free controller gave better performance than the boundary layer approach. This can be

attributed to the fact that in the boundary layer approach, there was a loss of robustness whereas in the proposed adaptive method, the robustness was retained. The Lyapunov stability analysis of the control law was given in Appendix-A.

Contributions: The article presented in Appendix-C contributed:

1) A chattering-free adaptive sliding-mode controller for ROV systems that utilizes a simple, lumped adaptive term that is effective at compensating unknown currents, poor estimates of individual ROV dynamic parameters and sudden unknown external impulses. As opposed to the common adaptive approach, the adaptive term does not require the linearity condition on the dynamic parameters and the creation of a regressor matrix;
2) An adaptive control law for ROV systems that does not require a priori knowledge of the upper bounds on the dynamics parameters of the ROV, which is needed for the conventional sliding-mode controller design.

2.3.2 ROV Control with Upper Bound Estimation of Lumped Uncertainty [D]

In this section, the work entitled "*Automatic Navigation and Control of the Falcon™ ROV* is summarized." In that work, a new controller for ROVs was implemented on the inspection-class Falcon™ ROV. The new controller was first developed for ROVM systems and was tested in ROVM simulation studies, which are discussed in Section 3.3.2 and Appendix-H of this dissertation. The reader should note that the Lyapunov stability analysis for the new controller is discussed in more detail in Section 3.3.2 and Appendix H in the context of ROVM dynamics and control. In its application to ROVs, the new controller addresses the shortcomings of the controller discussed in Appendix-C. The unique feature of the new controller formulation is that it pursues an estimate on the

upper bound of the two-norm of the difference between the known and the unknown dynamics, referred to as the lumped uncertainty vector, rather than the actual value of the difference, as was done in Appendix-C. Furthermore, an adaptive PID layer was incorporated into the controller scheme to further improve the performance.

For a detailed presentation of these technical developments, the reader is referred to Appendix-D. The reader should note that reference [25] in Appendix-D is included as Appendix-H in this dissertation.

Motivation: The primary motivation of this work was to implement an automatic control system for the Falcon™ ROV that does not have the shortcomings of the controller presented in Appendix-C. As well, a reliable navigation algorithm was designed in the context of this work. The other objective was to design a series of pressure housings for the underwater sensors as well as a skid that supports these housings. This skid can be easily attached to the ROV for field operations.

Having a reliable navigation system is a challenging problem since the sensor outputs are contaminated with noise. As well, a high sampling rate is not attainable, in particular for acoustic based ocean sensors such as the SBL and DVL devices. In addition, sensor drop-outs may occur in these systems due to a lost acoustic signal transmission, which often occurs in ROV missions. Therefore, a strategy is needed that addresses these issues and provides reliable state estimation in such occurrences.

The asymptotic stability of the controller of Appendix-C relies on a strict assumption on the time rate of change of the lumped uncertainty vector (the difference between the known and the unknown dynamics vector) – see Section 3.3.2 for details. Unfortunately, this condition may not hold throughout a typical ROV operation, especially when sudden

ROV motion is demanded by the human pilot or the automatic guidance control. This may even lead to potential instability of the system depending on the extent of the demanded sudden move. Furthermore, the adaptive component of this controller can lead to unstable when there is not enough excitation within the system, which is needed for the adaptive law to converge to the true values of the estimated parameters (cited as the "parameter drift" problem in existing adaptive control literature). Thus, a new controller is sought for a more reliable trajectory following task.

In the new navigation pod design, there is a need for pressure housings for the underwater sensors that are employed. This design must be waterproof and neutrally buoyant. In addition, it should allow the minimization of magnetic interfaces with the other ROV components such as on-board thrusters.

Methodology: In order to obtain ROV navigation with low noise and high update rate, an EKF was designed based on the work of [43]. The EKF blends the measurements of an acoustic positioning system, a DVL, a low-cost IMU, a compass, and a depth sensor with the predictions of Falcon™ ROV dynamics to further increase the performance of the navigation algorithm. The flowchart of the EKF employed in this work is shown in Figure 2-D.

To obtain the dynamic model of the Falcon™ ROV, first, a series of thruster characterization tests were performed to reveal the mapping from the thruster commands in percentages to thrust values in Newtons. The results are given in Figures 10-D through 13-D. Once the thruster characterization was complete, a series of parameter identification tests were performed to obtain the hydrodynamic parameters of the Falcon™ ROV for each DOF similarly to [64]. These results are tabulated in Table 1-D.

The verification of the derived hydrodynamic model was carried out by comparing the simulated output with the experimental one for a given thrust demand, as illustrated in Figures 14-D through 17-D.

The derived dynamic model of the Falcon™ ROV was incorporated into the EKF scheme and a number of tests were then carried out to generate the corresponding parameters that quantify the confidence in the model. The results are given in Table 6-D. Finally, the accuracy of the EKF in estimating the system states was demonstrated through wet tests using a high precision optical motion capturing system, as shown in Figures 23-D, 24-D and Table 7-D.

A new robust control law, Eq. (20)-D, was implemented in this work that exploited adaptation laws, Eq. (22)-D, which are free of the parameter drift problem. The parameter drift problem was remedied by introducing new terms in the adaptation law that have the functionality of keeping the adaptive terms within a safe range even when the persistency of excitation condition is not satisfied. With its robustness to the parameter drift problem, the new controller provides a more reliable control approach for ROV systems.

The derivation steps of this controller are the same as those given in Appendix-H, although the controller therein accounts for the on-board manipulator as well as the ROV. Again, the reader is reminded that reference [25] in Appendix-D is included as Appendix-H in this dissertation.

Over the course of the control project, a new wet-test facility was successfully constructed at Van Isle Marina in Sidney BC for the experimental validation of the proposed controller approach as shown in Figure 11.

Figure 11: A wet-test facility was constructed at Van Isle Marina in Sidney BC. This facility hosts a wide range of equipment necessary for the experiment.

Results: A number of experiments were performed in the constructed boathouse. To evaluate the step input performance of the controller, first, the Falcon™ was asked to move to a certain point from its current position. These tests were performed individually for X (surge), Y (sway), Z (heave) and ψ (yaw) motion, respectively, and the results are demonstrated Figures 25-D through 40-D. From these results, it was concluded that the controller performs satisfactorily. The controller's ability to follow a desired horizontal motion (X, Y) and heading plus depth (Z, ψ) was also evaluated individually, as demonstrated in Figures 41-D and 42-D. Finally, way-point and trajectory following manoeuvres were carried out, as illustrated in Figures 43-D through 45-D and Figures 46-D though 50-D, respectively. The repeatability of the controller in the most challenging trajectory following task was also tested, as reported in Table 11-D,

and it was concluded that the centimetre precision is attainable with the proposed controller.

A new waterproof navigation pod was designed that uses a blend of Doppler Velocity Log, compass, inertial measurements and acoustic position data. The depth rating of the new navigation pod is 300 meters and the aluminum components were anodized against corrosion. The new navigation pod was designed to be modular – DVL, IMU and Compass have their own housing – allowing us to optimize the position of each sensor in such a way that the magnetic interferences are minimized, as shown in Figure 12 (left). In addition, a new skid was designed for the new navigation pod, as shown in Figure 12 (right). This skid design provides a mounting surface not only for the new navigation pod's components but also for addition tools that might be needed in the future. Starboard and Delrin material were used in the design for neutral buoyancy.

Figure 12: A series of water-proof, pressure proof sensor canisters for each sensor were designed. The modular design allows the location optimization of each sensor on the new navigation skid in order to minimize magnetic interferences mostly due to the onboard thrusters.

Contributions: The article presented in Appendix-D contributed:

1. A new modular navigation pod along with a navigation skid that uses off-the-shelf underwater sensors. The new design has a modular structure allowing the location optimization for magnetic interferences;

2. An EKF that provides an accurate estimate of ROV motion with low noise and a high update rate based on a series of underwater sensors. This EKF is robust to sensor drop-outs owing to the use of the system mathematical model obtained through the performed research;
3. A novel advanced controller that is adaptive and robust with respect to changing underwater conditions or vehicle layout as well as to the parameter drift problem.

2.4 ROV Control: Thrust Allocation [C]

In this section, the thruster allocation portion of the work entitled "*Chattering-Free Sliding-Mode Controller for Underwater Vehicles with Fault-Tolerant Infinity Norm Thrust Allocation*" is summarized. As mentioned in Section 1.5.2, the controllers designed so far generate the forces and moments that need to act at the center of mass of the ROV for the controller design problem, and they need to be mapped into desired thrust values for each on-board thruster. There is an infinite number of ways to allocate the pilot's commanded thrust over the existing thrusters due to the redundancy in the thruster lay-out. In this work, a new method was proposed that minimizes the largest commanded thrust by using an l_∞ norm of the thrust vector. For a detailed presentation of the technical developments, the reader is referred to Appendix-C.

Motivation: To utilize this redundancy, the prominent approach in the literature is the l_2 norm solution. However, the l_2 norm solution allocation problem has some complications with existing practice, as pointed out in Section 1.5.2. Therefore, there is a need for a technique that removes the disadvantages of the l_2 norm solution for the thrust

allocation problem. This paper proposes a method that provides a better alternative to the l_2 norm based solution for the thruster allocation problem.

Methodology: To remove the disadvantages of the l_2 norm solution, an entirely new formulation for the thrust allocation problem was proposed in this paper. The new method minimized the largest commanded thrust by using an l_∞ norm of the thrust allocation in the optimization based on the formulation given in Eq. (21)-C. The l_∞ norm is defined as the absolute value of the largest component of the thrust manifold. By using the l_∞ norm to gauge the optimality of a thrust distribution, the largest single thrust in the distribution was minimized. This formulation was cast as a constrained linear programming problem in Eq. (27)-C that affords easy implementation of thruster saturation limits and fault-tolerance while addressing the two deficiencies of the l_2 norm formulation.

Since the thruster force allocation is a real-time application, the solution to the linear programming problem must be obtained at a real time rate. To obtain this solution rate, a recurrent neural network of Eq. (36)-C was designed in this work.

Results: In order to demonstrate the efficacy of the proposed fault-accommodating thruster force allocation scheme, a number of simulation studies were carried out. To demonstrate the improvement over the conventional l_2 norm solution, comparison studies were performed and the results were illustrated in Figures 12-C and 13-C. The results showed that the largest thrust value obtained from the l_2 norm and l_∞ norm minimizations for the ROV manoeuvre considered were 342.6 and 268.4N, respectively, which translates into a 21.6% reduction in the largest thrust. The average improvement

over the whole manoeuvre was found to be 11.9%. Furthermore, it was concluded in Section 3.2 of Appendix-C that there exist solutions where minimizing the l_∞ norm gives feasible solutions whereas minimizing the l_2 norm does not. Moreover, since the l_∞ norm minimizes the maximum component of the thrust manifold, it allows the thrusters to run within a safer range than the l_2 norm, providing more manoeuvrability for subsequent controller action.

Contributions: The thrust allocation portion of the article presented in Appendix-C contributed:

1) An entirely new formulation for the thrust allocation problem of ROV systems that is based on minimizing the l_∞ norm of the thrust vector was formulated as linear programming. This formulation is fault-tolerant and removes the disadvantages of the l_2 norm solution outlined in Section 1.4.2.
2) A recurrent neural network that solves the linear programming problem and provides real-time solution rates to the thruster allocation problem.

CHAPTER 3. SUMMARY OF CONTRIBUTIONS TO ROVM SYSTEMS

3.1 Chapter Review

In this chapter, technical building blocks towards the completion of Objective #2, #3 and #4 of Section 1.3 are summarized in the context of the coordinated control of the Saab-Seaeye Falcon™ ROV. As detailed in Section 1.3, an important function of the proposed HMI is the ability to translate the human pilot's desired end-effector motion directly from the 6-DOF joystick and distribute this motion over the ROV and manipulator in an optimal, coordinated way. To exploit this coordination, the system needs to be considered as a single mobile manipulator, instead of separate units, as is the case in conventional ROVM operation. This shift renders the system kinematically redundant; there are more DOFs than required. Hence, the extra DOFs can be used to achieve secondary objectives in addition to the primary end-effector following task. The use of these secondary objectives to constrain distribution of the primary task over the joint space of the mobile manipulator is referred to as "redundancy resolution" in the relevant literature.

In Section 3.2, different redundancy resolution algorithms developed in the pursuit of Objective #2 of Section 1.3 are summarized. In these works, the main focus was to find a way to dissect the commanded end-effector motion into separate ROV and manipulator motions while achieving conflicting secondary objectives. The manipulator motions created by the secondary objectives are internal – ones which produce no motion of the end effector but put the ROVM in a pose which eases the burden on the pilot in the coming moments. At times, different secondary objectives request significantly different internal motions that must be intelligently resolved.

In order for the redundancy resolution algorithm to be functional, it must be coupled with a controller and an input device such as a joystick. While the redundancy resolution generates reference state values for the ROV and the manipulator in response to the human pilot's joystick input, the controller works to realize the desired state values by generating a control signal that directs the on-board thrusters as well as the manipulator actuators. In Section 3.3 of this dissertation, work completed in the control and HMI aspects of the coordinated ROVM systems, in pursuit of Objectives #3 and #4 of Section 1.3, are summarized. For the control of the ROVM, different control methodologies from the ROV control work presented in Section 2.3.1 and Section 2.3.2 are extended to include the on-board manipulator, and performance is investigated. In addition, a new comprehensive HMI, needed to achieve Objective #4 of Section 1.3, is presented. This HMI involves a 6-DOF parallel joystick, and a 3D visual display and a graphical user interface (**GUI**) that enables a human pilot to smoothly interact with the Saab-Seaeye Falcon™ ROVM.

3.2 ROVM Kinematics

When controlling kinematically redundant ROVM systems, the main problem is to utilize the inherent redundancy to accomplish secondary tasks while still fulfilling the pilot's end-effector command. Appendix-E summarized in Section 3.2.1 presents a novel methodology in which two secondary objectives are appended to the primary task. While the method works well, its application to the Saab-Seaeye Falcon™ is too constrained – it falls short of covering a wide range of secondary objectives that could ease the burden on the human pilot during a typical ROVM operation. Furthermore, the method does not account for actuator failures that may occur during the ROVM tasks. Moreover, the scheme could demand excessive reference state values that could go beyond the capacity of the joints.

To overcome these shortcomings, the work of Appendix-F, summarized in Section 3.2.2, proposes a different technique that not only allows the inclusion of several (more than two) secondary objectives, but also provides a fault-tolerant property: upper and lower velocity limits are enforced for each joint in the mobile ROVM.

3.2.1 Redundancy Resolution with Two Secondary Objectives [E]

In this section, the work entitled "*Dexterous task-priority based redundancy resolution for underwater manipulator systems"* is summarized. For a detailed presentation of the technical developments, the reader is referred to Appendix-E.

Motivation: In current ROVM practices, the desired manipulator joint motions are created using a teleoperated master-slave arm configuration. This mode of operation depends on the ability of the ROV to hold station and decouples the manipulator and ROV DOFs. However, new initiatives in submerged science are requiring more frequent

deployment of ROVMs and more detailed operation of ROVM end-effectors on each mission. Time below the surface is a valuable commodity, and methods to improve the existing human-machine interaction are sought.

One way of improving the efficacy of the ROVM systems is to automatically coordinate the collective DOFs of the ROVM in response to a single pilot input for the end-effector. When ROVM DOFs are coordinated, the system becomes kinematically redundant; there are an infinite number of joint-spaces solutions for a commanded end-effector position and orientation. The available redundant DOF can be used to achieve additional objectives, besides the primary end-effector task, using a proper redundancy resolution technique for the automatic coordination of the ROVM DOFs. In order to fulfill the primary and the additional objectives, the redundancy resolution scheme must allow the incorporation of multiple criteria. In Appendix-E, a ROVM redundancy resolution with two secondary objectives was proposed and implemented in simulations of a generic ROVM system.

Methodology: In Appendix-E, a new heuristic approach based on the combination of the Task-Priority Redundancy Resolution approach [31], [32], and the Gradient Projection Method (**GPM**) [65] was proposed for the ROVM systems. The primary objective of following the end-effector desired velocities was realized using the right Moore-Penrose pseudoinverse solution that corresponds to the minimization of the two-norm of the joint-space velocity vector [65]. As for the secondary tasks, they were considered to be:

- Avoiding singularity configurations of the manipulator;

- Minimizing the vehicle motion in an attempt to mimic conventional ROVM operations whenever possible.

These two tasks are often in conflict. When the vehicle motion is penalized, the system starts to use the arm more exhaustively to accomplish the desired end-effector task making the system more susceptible to singular, or near singular, configurations. Near-singular configurations are not desirable since they result in undesirably high joint velocities and oscillations that make it difficult to control the arm motion. The proposed approach provided a novel means to resolve this conflict by using the ROV's mobility in an optimal, coordinated manner to avoid singular configurations of the manipulator, resulting in dexterous manipulation of the arm.

In Eq. (19)-E, the task-priority redundancy resolution algorithm was put in charge of realizing minimal ROV motion, and the gradient projection term was put in charge of avoiding near singular configurations of the manipulator. The two methods were blended through the shaping function of Figure 1-E. The shaping function weighs one method over the other based on the mobility index value of Eq. (17)-E. The mobility index value quantified the proximity of the system pose to singular configurations as well as the degree of conflict between the two tasks. The index value was used to supervise the ROV movement and its coordination with the end-effector motion.

Results: Numerical simulations were performed to evaluate the performance of the method in satisfying the primary and secondary objectives. To this end, using a third order polynomial function with zero initial and final velocities, the pilot input was mimicked and fed into the scheme to obtain the desired joint rates. In this case study, a hypothetical ROVM system was considered, but the technique used is directly applicable

to the Falcon™ ROVM, which is the focus of the research program. The obtained state values were reported in Figure 4-E. The results indicated that the method produces kinematically and dynamically realizable ROV-manipulator state values that significantly improve the dexterity of the system. Figures 5-E and 6-E demonstrated that the dexterity measure was brought from 0 (singular arm configuration) to 1.2 (dexterous arm configuration). In conclusion, the method ensures dexterous manipulation by using ROV motions to avoid the singular configurations while accomplishing the primary task.

Contributions: The article presented in Appendix-E contributed:

- A new method for the redundancy resolution problem of ROVM systems that provides a means to avoid singular configurations of the manipulator, and provides dexterous manipulation by using the ROV's mobility in an optimal, coordinated manner;
- A novel combination of two different redundancy resolution techniques using the mobility index, which is the variant of the conventional Yoshikawa's measure of manipulability [31]. The method eliminates unnecessary ROV motions when the demanded end-effector location is within the reach of the manipulator, and hence provides energy efficient control of the ROVM.

3.2.2 Redundancy Resolution with Multiple Secondary Objectives [F]

In this section, the work entitled "*Redundancy Resolution for Underwater Mobile Manipulators*" is summarized. The new redundancy resolution algorithm is fault-tolerant and has the ability to accommodate multiple secondary objectives. In addition, it can impose dynamic joint-velocity constraints, ensuring safer ROVM operation. For a detailed presentation of the technical developments, the reader is referred to Appendix-F.

Motivation: The work of Appendix-E allows only two secondary objectives. Given the diversity and the complexity of today's, and especially tomorrow's, advanced ROVM tasks, a larger set of secondary objectives must be possible in a comprehensive ROVM redundancy resolution algorithm. Moreover, for a reliable redundancy resolution scheme, a fault-tolerance property must be included. Fault-tolerance ensures that the underwater mission is not interrupted in the event of a joint failure, and that unrealistic joint velocities are not requested of the ROVM controller that is applied downstream of the redundancy resolution phase.

Methodology: In Appendix-F, the redundancy resolution equation of Eq. (14)-F consisted of two parts – a homogeneous solution and a particular solution. The particular solution yielded the primary objective, the desired end-effector motion, and was obtained through the right Moore-Penrose weighted-pseudoinverse solution, which corresponds to the minimization of the l_2 norm of the weighted joint-space velocity vector. The weighting was used for setting preferences for certain joints in the solution – for example when motion in a particular joint is more preferable that in others. As for the secondary objectives, they were considered within the homogeneous solution. The homogeneous solution corresponds to the internal ROVM motion and does not affect the state of the end-effector at a position or velocity level. The secondary objectives considered were:

- Avoiding joint displacement limits;
- Avoiding singularity;
- Keeping the end-effector in sight of the on-board camera;
- Minimizing the drag-force resistance;
- Minimizing the vehicle motion.

The scheme was considered at two different levels, as illustrated in Figure 13. The human pilot was considered at a higher level than the fuzzy logic based artificial pilot. The human pilot set the desired end-effector motion using a 6-DOF input device while the secondary artificial human pilot governed the motion of the system by weighing the importance of the secondary tasks based on a set of if-then-type linguistic rules. These linguistic rules reflect the knowledge of a skilled human operator.

Within the proposed scheme, each criterion was defined within the framework of the GPM and was assigned a weight, which determines the relative emphasis on each criterion. The artificial pilot assigned a weight factor for each criterion based on the Mamdani fuzzy inference system [66], which processed the fuzzy logic membership functions given in Figure 2-E and their corresponding set of rules, given on page 331-F. The resulting weight schedule yielded a self-motion (null-space motion) that emulates how a skilled operator would command the vehicle and the manipulator with the full redundant capacity being expended in the vehicle-manipulator coordination.

A fault-tolerant property was then added over top of the artificial pilot calculations in order to accommodate joint failures based on the work of [67]. Joint failure refers to three incidents: first, it occurs when joints break down; second, it happens when a joint has partly lost its driving capacity; and third, it takes place when the redundancy resolution scheme produces joint velocities that are beyond the capacity of the corresponding joints. The method proposed is fault-tolerant and prevents the redundancy resolver from requesting intolerable joint motions in any of the events outlined. To this end, first, a maximum allowable joint rate (0 for a joint that is in complete failure) was assigned to the faulty joint and its end-effector velocity contribution was subtracted from

the desired end-effector velocities, Eq. (29)-F. The corresponding column of the weighted-Jacobian matrix, which relates the associated joint's motion to the end-effector velocities, was removed, as shown in Eq. (30)-F, and the fault-tolerant redundancy resolution equation, Eq. (31)-F, was applied with the new desired end-effector velocity vector and the Jacobian matrix. The final desired state vector was then constructed by entering the preset maximum velocity in the corresponding location in the state velocity vector. The overall scheme is outlined in Figure 3-F.

Results: In Appendix-F, three simulation case studies were performed on the Falcon^{TM} ROV with a 4-DOF Hydrolek^{TM} manipulator. In the first case study, the end-effector velocities were commanded according to Figure 6a-F based on a third order polynomial function with zero initial and final velocities that mimicked the pilot's input. First, only the weighted-pseudoinverse solution was employed to complete the redundancy resolution, and the obtained state values were reported in Figures 7-F and 8-F. The secondary objectives index values, reported in Figure 9-F, indicated that the system performed very poorly in terms of realizing the secondary objectives. In the second case study, the complete fault-tolerant redundancy resolution scheme was implemented for the same end-effector manoeuvre, and the obtained state values and corresponding secondary index values were reported in Figures 10-F, 11-F and 12-F. The index values indicated that significant improvement was obtained with the inclusion of the fuzzy-logic blending of the secondary objectives. The results also indicated that the scheme generated desired joint velocities within their respective limits and handled sudden joints failures very well. In the third case study, incremental differences in system behaviour were investigated with the secondary objectives added in steps, and the results were reported in Fig 19-F.

The results indicated that the proposed method was successful in realizing the secondary objectives.

Contributions: The article presented in Appendix-F contributed:

- A new fault-tolerant redundancy resolution algorithm that allows the inclusion of multiple, meaning more than two, secondary objectives with the consideration of joint velocity limits and joint failure;
- A casting of skilled human pilot tendencies in the form of scalar objective functions;
- A technique that relies only on instantaneous ROVM state information and not preordained knowledge of the ROV and/or end-effector trajectories, making the technique suitable for on-line operations.

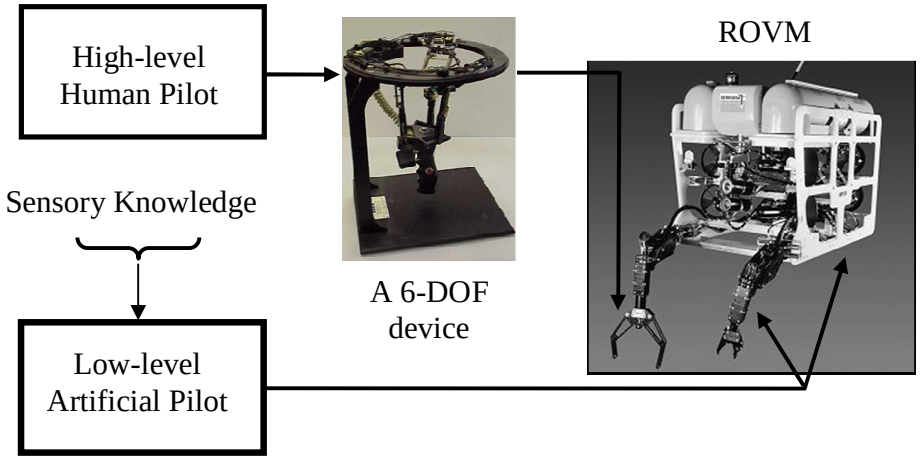


Figure 13: The high-level human pilot sets the desired end-effector motion using a 6-DOF joystick while the low-level artificial pilot determines the optimal posture with respect to the secondary objectives in response to the human-pilot input. The ROVM System image is courtesy of http://seaeye.com.

3.3 ROVM Coordinated Control

While the redundancy resolution generates reference state values for the ROV and the manipulator in response to the human pilot's input, a controller works to realize the desired state values by directing the on-board thrusters as well as the manipulator actuators. As summarized in Section 3.3.1, the work of Appendix-G employed the controller of Appendix-C in the ROVM paradigm to this end. Since the derivation of this controller requires a single dynamic model of the overall ROVM system, a ROVM dynamic modelling strategy was also presented in this work. To overcome the disadvantages of the controller of Appendix-C, as detailed in Section 3.3.2, the work of Appendix-G proposes a new controller that was summarized in Section 3.3.2. Also, proposed in Appendix-G, a new HMI that facilitates the inclusion of real human tendencies in the coordinated control of ROVM systems.

3.3.1 Redundancy Resolution Coupled with a Controller [G]

In this section, the work entitled "*Exploiting Redundancy in Underwater Vehicle-Manipulator Systems*" is summarized. The model-based controller of Appendix-C was extended to include the manipulator dynamics and coupled with the redundancy resolution of Appendix-F. For a detailed presentation of the technical developments, the reader is referred to Appendix-G.

Motivation: To obtain a functional redundancy resolution method, the scheme needs to be coupled with a controller. Control methods generally depend on dynamic models of the system, and hence, there is a need for a reliable dynamic model. This dynamic model also is needed to test out the devised controllers' performance in a simulation environment before the field implementation. Thus, the dynamics model needs to be

computationally efficient for real–time implementation and applicable to the control theory to be used. While the redundancy resolution generates desired ROVM joint, it is the controller's duty to realize these reference joint rates. This requires devising a control strategy that can perform well in the presence of inaccuracies in the system model as well as disturbances.

Methodology: The quasi-Lagrange approach was used to derive the equations of motion for a ROVM system, and Eq. (14)-G was obtained. The quasi-Lagrange approach, originally presented in [68] and [69], was preferred since it allowed the easy inclusion of additional subsystems, such as the articulated collection of manipulator links, and generated the equations of motion in an analytical form – a requirement for the model-based controller used. In addition, the quasi-Lagrange approach used states values in terms of the body-fixed values. Thus, it was a more convenient approach to work with since on-board sensors and actuators measure and create values in the moving local-frames affixed to the ROV and manipulator links.

In terms of the controller, the chattering-free sliding mode controller of the work of Appendix-C was employed as a foundation due to its adaptability to changes in the system dynamic model, as demonstrated in the work of Appendix-C. The control law of Eq. (29)-G and the adaptation law given in Eq. (28)-G were derived for ROVM systems based on Lyapunov theory. Unlike conventional adaptive control theory, the derived control signal does not need parameterization of a regressor matrix and unknown parameter vector, thereby eliminating the need to simplify system dynamics and thus allowing the use of a more detailed dynamic model in the controller.

Results: Numerical, time-domain simulations were performed on the Falcon™ ROVM with a 4-DOF Hydrolek™ manipulator to reveal the performance of the controller for the ROVM systems. To reflect uncertainties present in the ROVM's system dynamics, the exact added mass and drag coefficients in the controller's dynamic model parameters were changed by 40 to 60% in random fashion. In these simulations, the redundancy resolution algorithm of Appendix-F generated the reference joint rates in response to the pilot's end-effector input. As in Appendix-E and Appendix-F, the pilot input was mimicked using a third order polynomial function with zero initial and final velocities in this study. It was found that the proposed scheme provided high performance in the end-effector following task while generating the kinematically and dynamically realizable state values for ROVM systems, as can be seen in Figures 3-G through 5-G. The scheme realized the secondary objectives with success and the controller output was chattering-free, and hence applicable to a real ROVM platform, as demonstrated Figures 6-G and 7-G, respectively. In conclusion, the results illustrate that a complex spatial manoeuvre can be accomplished with a 4-DOF manipulator mounted on a small ROV using the proposed scheme.

Contributions: The article presented in Appendix-G contributed:

- A non-linear controller for ROVM systems that can successfully operate in the presence of inaccuracies in the system dynamic model;
- A unique ROVM operating scheme that can accomplish complex subsea intervention missions. In the scheme, ROV and manipulator motions can be coordinated using a multiple objective redundancy resolution to achieve a desired end-effector motion as defined by a single pilot command.

3.3.2 Redundancy Resolution Coupled with a Controller and a HMI [H]

In this section, the work entitled "*New Operation Scheme for Underwater Mobile Manipulator*" is summarized. In this work, a new controller that builds on the controller presented in Appendix-G was proposed and implemented in hardware-in-the-loop simulation. Secondly, an end-effector input device, a 6-DOF joystick [42] originally developed by RSI Research Ltd. and shown in Figure 10, was interfaced with the redundancy resolution scheme presented in Appendix-F. In addition, an HMI was developed to facilitate the coordinated control of ROVM systems with the most promising tools devised along the course of this research. For a detailed presentation of the technical developments, the reader is referred to Appendix-H.

Motivation: There are two major motivations behind this work, namely obtaining a high performance controller for the ROVM systems, and designing a new HMI that facilitates the use of the developed redundancy resolution and control tools.

The controller of Appendix-G has some shortcomings. For instance, the asymptotic stability of this controller is not guaranteed when:

$$-\dot{\tilde{\boldsymbol{f}}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} > 0$$

and the Lyapunov function is negative definite as long as $\|\boldsymbol{s}\|_2 > \sqrt{b/\lambda_{\min}(\mathbf{K})}$ where b is the positive scalar upper bound, i.e., $-\dot{\tilde{\boldsymbol{f}}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} \le b$, and $\lambda_{\min}(\mathbf{K})$ is the minimum eigenvalues of $\mathbf{K}$. This means that $\|\boldsymbol{s}\|_2$ is bounded and the controller is only uniformly ultimately bounded. The invalidity of the asymptotic stability condition may occur when sudden motions are requested from the vehicle and the manipulator. This can be attributed to high $\dot{\tilde{\boldsymbol{f}}}$ values occurring in those moments where $\dot{\tilde{\boldsymbol{f}}}$ is the time rate of

change between the known and the unknown ROVM system dynamics, or the lumped uncertainty vector. In fact, $\dot{\tilde{\boldsymbol{f}}}$ could be very large, which in turn *b*, and as such the controller could perform poorly or even cause instability. Since sudden motions of the ROVM's are inevitable when a pilot's intent suddenly changes, a method to make the controller of Appendix-G perform better in the presence of such sudden maneuvers was required. Such sudden ROVM maneuvers occur when the objective changes abruptly, for example, from slow picking tasks to fast placing tasks akin to "stop-and-go" type tasks.

In addition, the adaptation law of the previous controller discussed in Appendix-G, Eq. (28)-G, is susceptible to the the parameter drift problem. The parameter drift problem manifests itself in the adaptation results as the unboundness in the estimated values. This is mostly associated with non-parametric uncertainties (disturbance and noise) and unmodelled dynamics and needs to be addressed properly for stable controller behaviour. Especially in ROVM paradigm, this problem is more likely to occur since the existence of the manipulator makes the adaptation quicker due to its faster response, and as a result, the estimated parameters may be rather oscillatory. In particular, when these oscillations move into frequency range of unmodelled dynamics, the unmodelled dynamics may be excited. As such, the adaptation occurs based on these meaningless oscillations in the system, and in turn, could drive the system into instability. The existence of the 6-DOF joystick makes this scenario more likely to happen since possible abrupt changes in the pilot's demand may induce such oscillations in the system. In addition, the 6-DOF joystick introduces noise into the system, which may also lead to the parameter drift problem. These issues are usually ignored in the relevant ROVM literature since the existing research is mostly confined to the controlled conditions of simulation studies.

However, since the 6-DOF joystick is interfaced with the redundancy resolution and control scheme in the current study, this problem can no longer be overlooked and needs to be addressed properly for a reliable operation.

Execution of the redundancy resolution and controller strategies for ROVM systems relies on an improved HMI to gather the desired end-effector motion from the human pilot. In conventional ROVM operations, a master arm that is kinematically similar to the ROV manipulator is used to directly determine the joint rates. For the underactuated manipulators of small ROVM systems, that method is impractical. Thus, an advanced HMI is required that acquires the end-effector motion. The end-effector motion is then passed through the system redundancy and the controller algorithms to set the ROV and manipulator joint rates.

Methodology: A new controller that addresses the potential stability issue outlined in the motivation section was developed. According to Eq. (32)-H, the adaptation law of Eq. (28)-G was changed to estimate the upper bound of the lumped uncertainty vector in lieu of the lumped uncertainty vector itself. This adjustment replaces the strict requirement of converging to the unknown lumped uncertainty vector of the original asymptotic stability condition with a more relaxed requirement of converging to the upper bound on the two norm of the lumped uncertainty vector. Thus, the new strategy provides asymptotic stability, provided that

$$\left\| \tilde{\boldsymbol{f}} \right\|_2 \leq \rho$$

is satisfied where ρ is a scalar positive constant that represents the upper bound on the two norm of the lumped uncertainty vector, i.e., $\left\| \tilde{\boldsymbol{f}} \right\|_2$. The term ρ is adaptive and its

on-line estimation is governed by the adaptation law of Eq. (32)-H. The on-line estimation continuously works to satisfy the stability criterion of $\left\| \tilde{\boldsymbol{f}} \right\|_2 \le \rho$ by relating the estimation to the performance of the controller. With this update, the asymptotic stability is guaranteed as shown in [70] even when $-\dot{\hat{\boldsymbol{f}}}^{\mathrm{T}} \boldsymbol{\Gamma}^{-1} \boldsymbol{w} > 0$ such as when ROV makes sudden manoeuvres.

In order to further enhance the controller performance, a PID layer was also included in the scheme. The PID layer also provided a tuning tool for the controller performance in particular when the robust signal is weak. To make the PID layer adaptive, the adaptation law of Eq. (29)-H was derived, again based on Lyapunov theory.

The new adaptation laws were designed to be robust to the "parameter drift" problem [54], which is a common issue in most adaptive controllers. In this work, the drift problem was circumvented by adding a leakage term [58] based on a version of the $\sigma-$modification technique [59]. While this update, however, makes the controller uniformly ultimately bounded, the lower bound on the error metric is smaller compared to that of the chattering-free controller, thereby yielding a more robust controller action.

As reported in [70], a work presented by the author of this dissertation, upper bound adaptation yielded significant improvements over the controller discussed in Appendix-G. In fact, in the case study of [70], it was shown that the maximum end-effector trajectory following error was reduced by 33% with the use of the new controller.

The l_∞ norm thruster allocation was performed to exploit the redundancy in Falcon™'s redundant thruster lay-out based on the work of Appendix-C in the proposed scheme due to its advantages over the conventional l_2 norm thruster allocation, as presented in Appendix-C.

A new HMI was also proposed and implemented in hardware in the current work. This HMI involves a 6-DOF joystick produced by RSI Research Ltd. [42], a realistic 3-D graphical display and an advanced GUI, as shown in Figure 14 and Figure 15, respectively. The 6-DOF joystick was interfaced with the proposed redundancy resolution plus the control scheme and was used by the pilot to generate desired joint rates. The realistic 3-D graphical display was used to provide visual information to the pilot regarding how the system responses to the end-effector command. Finally, the GUI allowed the user to set the control mode of operation with their respective settings and to monitor the system response to given motion commands.

(a) (b)

Figure 14: The 3D display provides visual clues to the human pilot. (a) third party view of the ROVM system. (b) First person (camera view) of the end-effector (EE). The display tool can also be used for pilot training.

There were three user modes developed for the GUI, there are three modes namely, *Joystick Control*, *Manual Control* and *Automatic Trajectory Control*. In the *Joystick Control*, the pilot operates the 6-DOF RSI joystick (shown in Figure 10) to generate reference joint rates that are fed into the redundancy resolution algorithm of Appendix-F. In *Manual Control*, the pilot controls the vehicle and the manipulator separately akin to

the conventional methods using HCUs shown in Figure 9. Additionally, the pilot can use a station-keeping option in this mode. When selected, the station-keeping option activates the controller presented in Appendix-B to keep the vehicle stationary while the pilot operates the manipulator using the standard HCU shown in Figure 9 (right figure) – the vehicle acts as a stationary foundation for the manipulator. Without station keeping, it was found to be difficult to run the manipulator in any useful fashion due to the dynamic coupling. The controller presented in Appendix-B was chosen over the controller of Appendix-A for the station-keeping option since the controller of Appendix-B allows the direct inclusion of the manipulator disturbance forces into the controller action.

Finally, in *Automatic Trajectory Control,* the pilot enters the coordinates of the desired final locations of the end-effector. Upon receiving the desired coordinates, the *Automatic Trajectory Control* generates desired end-effector rates based on a third order polynomial function with zero initial and final velocities that are fed into the redundancy resolution algorithm.

Results: Hardware-in-the-loop simulations were performed on an inspection-class Saab Seaeye FALCON™ ROV equipped with a 4-DOF Hydrolek™ manipulator to demonstrate the effectiveness of the proposed controller. In the simulation studies, the model differed by up to 40% to model inaccuracies in the system dynamics equations. In the simulation, 6-DOF RSI joystick was driven to generate desired end-effector inputs. The time history of the obtained state values and error values in response to the RSI joystick input was reported in Figures 11-H through 17-H. Small error values indicated that the proposed controller performs well even under high inaccuracies in the system's

dynamic parameters. Figure 18-H indicated that the secondary objectives were met successfully. Regarding the adaptive terms and their evolution over time, they were reported in Figures 19-H and 20-H. These figures indicated that the adaptation law for the upper bound was successful in predicting the bound on the unknown dynamics and provided smooth values for the bound similar to those of the PID the adaptation law.

In addition, comparison simulation studies were performed to assess the controller performance with respect to one of the most commonly used computed-torque (Feedback Linearization) controller [54]. As shown in Figures 14-H and 17-H, the proposed controller outperformed the computed-torque controller significantly.

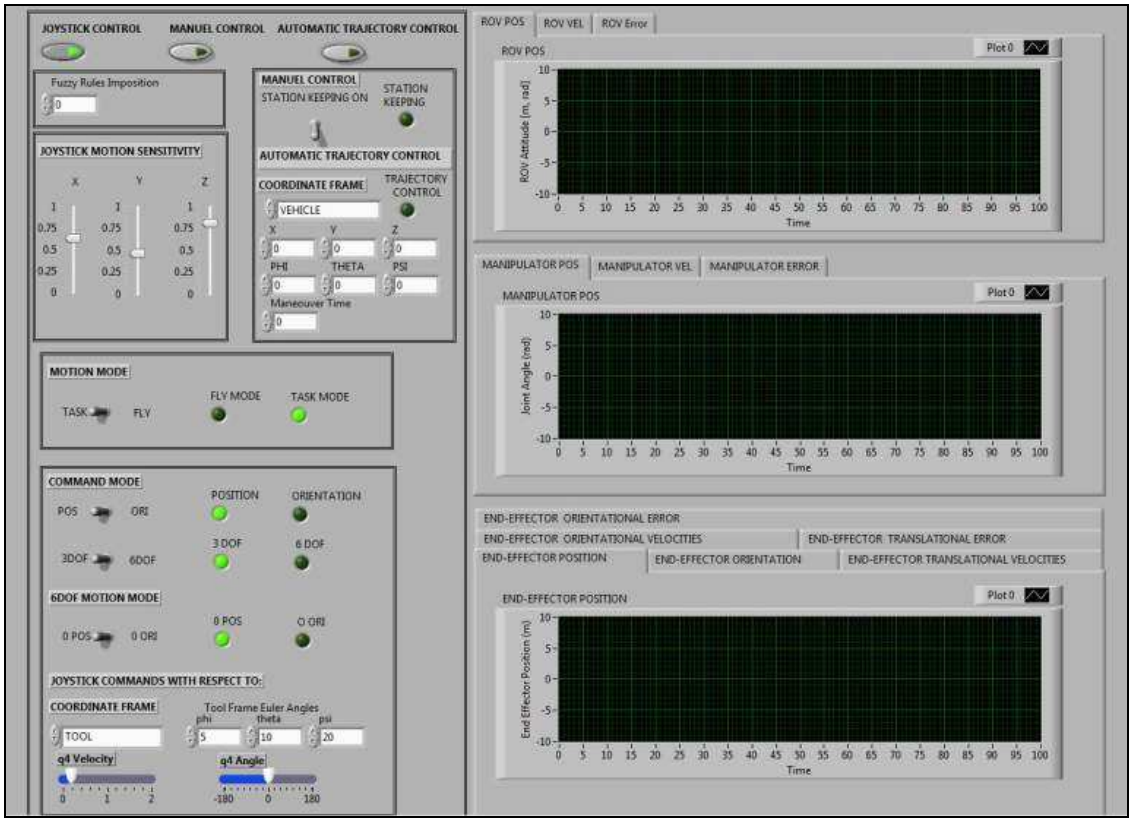


Figure 15: Snapshot of the proposed GUI. The developed tools of Appendix-B, Appendix-F, and Appendix-H were integrated into the GUI for the coordinated control of ROVM systems.

Contributions: The article presented in Appendix-H contributed:

- A new model-based nonlinear ROVM controller. This controller contains two layers of adaptivity; the first layer adjusts PID gains, and the second layer estimates the bound on a lumped uncertainty vector. The proposed adaptation laws are robust to the parameter drift problem that would otherwise lead to instability in the presence of poor system input signals.
- A new HMI that can effectively drive a coordinated controller for ROVM systems. The most promising tools that were developed over the course of the research program were integrated into the HMI. These included the station-keeping controller from Appendix-B, the fuzzy logic based redundancy resolution from Appendix-F, the adaptive sliding-mode controller with the upper bound estimation on the lumped uncertainty vector from Appendix-H.

CHAPTER 4. CONCLUSIONS AND FUTURE WORK

4.1 Conclusions

This dissertation presented a series of technical developments that collectively form the foundation of an advanced small ROVM system based on a Saab-Seaeye Falcon™ ROV. The primary contribution of this research and development project was the transformation of this inspection class ROV into an accessible and economic tool for interacting with submerged coastal infrastructure such as that found on the VENUS and NEPTUNE observatories. In addition to that holistic goal, many novel contributions were made along the course of the research program in the area of coordinated control of ROVM systems. Below, the accomplishments achieved in this field, as well as the potential research avenues for future work, are reviewed with reference to the technical objectives set in Section 1.5.

4.1.1 ROV Research

ROV Station Keeping Control: The design of station keeping controllers was the first step in the development of coordinated control for the ROVM system. In conventional

ROVM operations, human pilots try to hold the ROV stationary while working the manipulator. An automated stabilization of the vehicle around a desired set point that compensates for disturbance forces acting on the ROV, including those from the manipulator, was deemed necessary for the small ROV being considered: the manipulator and tether disturbances would manifest much quicker than a pilot's conditioned response for the smaller, lighter Falcon™ ROV. To this end, two different station keeping controllers were developed, namely, a combined MIMO H_∞-sliding-mode controller and a SISO sliding-mode controller. While the former controller accounted for the disturbances in the form of frequency domain range specifications, the latter directly incorporated them as forces in the controller design. As well, a tethered-ROVM time-domain simulation scheme was developed and implemented to evaluate controller performance in the presence of tether and manipulator disturbances.

The novel aspect of the combined MIMO H_∞-sliding-mode controller was the combination of the two controllers in a hierarchical structure. In this structure, the low-level sliding mode controller worked to drive the state of the nonlinear ROV system towards the equilibrium state of $\mathbf{0}$ while the high-level H∞ controller worked on a system whose dynamics model was more consistent with the linearized model used in the design of the H∞ controller. This unique way of combining two individual controllers culminated in a significant increase in the station-keeping performance. In fact, it was shown that the combined H∞-sliding-mode controller improved the position and orientation error by as much as 35% and 47%, respectively, compared to the sliding-mode controller alone, and by 49% and 49%, respectively, compared to the H∞ controller alone.

With regards to the SISO sliding-mode controller, it provided a novel framework in which the tether and manipulator disturbances could be directly accounted for in the derivation of the control law. In a numerical simulation case study, the hydrodynamic parameters were altered by up to 40% to mimic uncertainties in the system dynamic model of the controller. It was shown that the largest errors were approximately 0.15m and 0.08m along the X and Z axis, respectively, when only the manipulator disturbance was considered. In another simulation study, the tether disturbance knowledge was also incorporated into the controller signal alongside with that of the manipulator, and the largest errors were found to be 0.063m and 10^{-3}m along the X and Z axis, respectively. In conclusion, the inclusion of the tether disturbance knowledge in to the controller reduced the maximum Euclidian-norm of position error from 0.1655m down to 0.047m, a 71% decrease.

In addition, by using this novel strategy, the static control gains of conventional sliding-mode controllers, which are typically obtained through the conventional trial and error approach, were replaced with those that were a function of the time-varying articulated inertias. The level of adaptivity brought by this approach manifested itself as higher controller performance.

ROV Trajectory Following Control: Whether operating in strict ROV mode to traverse a desired search pattern across the seabed or in automatic ROVM mode to guide the end-effector using a joystick, the small Falcon™ ROV had to be able to follow a commanded route no matter if that route was preset or generated on-the-fly. To provide that capability, a chattering-free adaptive sliding-mode controller was employed. Conventional sliding-mode controllers use a discontinuous term that provides robustness

to the inaccuracies in the system dynamic model. The chattering-free adaptive sliding-mode controller removed the high-frequency control action of conventional sliding-mode controllers, which could potentially cause premature wear-out in the actuators, excessive heat dissipation in the motor circuitry or unstable system behaviour, by replacing the discontinuous term with a continuous one. This continuous term estimates the difference between the known and the unknown dynamics and acts on the controller as a corrective term. The ability of this continuous term to converge to the true values and its effectiveness on the reduction of the chattering phenomenon in the ROV paradigm was shown through numerical simulation studies.

The chattering-free adaptive sliding-mode controller had some shortcomings, such as parameter drift and potential instability in the presence of sudden ROV motions. The parameter drift problem leads to unstable growth in the estimation when there is not enough excitation in the system states and considerably limits the applicability of the controller for a broader set of operation scenarios. The latter difficulty was deemed important since ROVM operations involve a human pilot whose intent may change suddenly.

The shortcomings of the chattering-free adaptive sliding-mode controller were overcome by the robust adaptive sliding-mode controller coupled with an adaptive PID layer. This control strategy was model-based and had two layers of adaptivity; the first layer was for adjusting PID gains, and the second layer was for estimating the upper bound on a lumped uncertainty vector. The adaptation laws of this controller were found to be robust to the parameter drift problem that would otherwise lead to instability in the presence of poor system input signals. With this in mind, the robust adaptive sliding-

mode controller coupled with an adaptive PID control layer was chosen for implementation on the inspection-class Saab-Seaeye Falcon™ ROV.

Automated Control of a Falcon™ ROV: The robust adaptive sliding-mode controller coupled with the adaptive PID control layer was implemented on the Falcon™ ROV. Along the course of this implementation, a new wet-lab facility at Van Isle Marina in Sidney, BC, was constructed for ROV navigation and control testing. In addition, a series of pressure housings were designed and manufactured for the following underwater instruments:

- SBL acoustic positioning system;
- Doppler Velocity Log (DVL);
- Inertial Measurement Units (IMU);
- Compass and depth sensor.

For the SBL positioning system, the SouthStar™ SBL was used. This system had its own tether from its surface station to the target transponder on the ROV to provide higher update rates and created tether clutter, which in turn created extra disturbances on the ROV. To eliminate the SBL's tether line, the SBL signals were rerouted through the existing fiber-optic cable by interfacing it with the existing Falcon™ telemetry.

The measurements coming from the on-board sensors for the position tracking of the ROV contained both the true states of the ROV, and the noise. Since accurate, low-noise and high-update state information was crucial for the implementation, an EFK was used that separates the noise from the corrupted signal and provides an accurate estimate of ROV motion with low noise and a high update rate. To further increase performance of the navigation algorithm, especially in the case of sensor drop-outs due to the lost

acoustic signal transmission, the dynamic model of the Falcon™ ROV was integrated into the EKF. The dynamic model of the Falcon™ ROV was obtained through a series of water tests. The designed pressure housings were attached to the navigation skid, which was specifically designed for the implementation project.

Custom, real-time compatible code was generated for the implementation of the robust adaptive sliding-mode controller coupled with the adaptive PID control layer on the Falcon™ ROV. This code was coupled with a Lyapunov based guidance algorithm that generates reference state values to the controller for a given set of desired way-points based on the feedback coming from the navigation algorithm.

Using the facilities in the wet lab, the experimental validation of the designed EKF through a near surface test manoeuvres was performed in order to evaluate the accuracy of the EKF's state estimation. To this end, the EKF's estimates of the horizontal positions were compared to those of a high precision motion capture system and it was found that the RMS estimation error of the EKF was 0.15m for the surge motion and 0.16m for the sway motion. With this capability, high precision geo-referencing becomes possible with the existing Falcon™ ROV. In addition, further experimental tests were performed to demonstrate the performance of the proposed controller. During a rectangular trajectory following task, the average RMS error values of 0.08m, 0.04m, 0.04m, and 2.52degree values were recorded along the X, Y, Z axis and around the Z axis of the North-East-Down reference frame, respectively. In conclusion, centimetre-level precision was achieved with the proposed automatic control scheme, thereby significantly extending the capabilities of the Falcon™ system for high precision tasks

Infinity-Norm ROV Thrust Allocation: Within the two control strategies developed, the generalized force and moment calculated for the vehicle's center of mass had to be translated into a collection of forces on board propeller type thrusters. To this end, a new thrust allocation scheme methodology was developed that utilized the redundancy inherent in the thruster lay-out. This method was based on the minimization of the maximum single thrust components of the thrust manifold and was called the l_∞ norm formulation, as opposed to the conventional l_2 norm solution, which minimizes the two norm of the thrust manifold. In addition, real-time solution rates to the l_∞ norm minimization problem were achieved by designing a recurrent neural-network that inherently solves the l_∞ norm optimization problem.

In one case study, it was shown that the largest thrust values obtained from the l_2 norm minimization and l_∞ norm minimization for the ROV manoeuvre considered were 342.6 and 268.4 N, respectively, which translate to a 21.6% reduction. The average improvement over the whole manoeuvre was found to be 11.9%.

The simulation results showed that the l_∞ norm formulation allows the thrusters to run within a safer range than the l_2 norm and hence provides more manoeuvrability for subsequent controller action. Furthermore, it was concluded that the l_∞ norm solution is capable of finding feasible solutions in situations where the l_2 norm formulation fails to find one; making the l_∞ norm formulation a strong alternative to the conventional l_2 norm approach.

4.1.2 ROVM Research

With regards to the ROVM research, the main focus of the research and development program was to devise novel algorithms that would allow the small, low-cost Falcon™ ROV and a unique 5 function Hydrolek™ manipulator to be combined into an accessible and economical ROVM system. To complete detailed subsea tasks, the current operating procedures of larger work-class ROVMs cannot be downloaded directly to the new small system. A means of automatically coordinating the vehicle and manipulator motions was deemed necessary and this coordination was realized through novel redundancy resolution, extension of the ROV trajectory control to the ROVM paradigm and integration of a HMI that involves a unique spatial joystick, 3D visual display and a GUI.

Redundancy Resolution for Small ROVMs: In standard ROVM operations, the motions of ROV and the manipulator are guided independently by human pilots, and, to further ease the burden on the pilots, the ROV is typically held stationary. This practice not only eliminates the redundancy inherent in the complete ROVM system but also compresses the end-effector workspace beyond useful limits; the workspace is already small due to the under-actuated manipulators that most small ROVs employ. In order to increase the dexterity of the small ROVM systems, one must complement the manipulator DOF with the active ROV DOF. In doing so, the ROVM becomes a unified redundant manipulator that is capable of performing detailed 6-DOF subsea end-effector tasks (primary task) as well as secondary tasks using the redundant DOFs.

As a first step to design such intelligent coordination schemes, the Task Priority and Gradient Projection Method were combined in a novel way to accomplish the secondary objectives, which were defined as the singularity avoidance and the minimal vehicle

motion in the current research. In order to gauge not only the singular configurations but also the need for the ROV motion, a mobility index number was proposed that was derived from a modification of Yoshikawa's measure of manipulability. Using the mobility index, the ROV motions were applied in an optimal, coordinated manner through a simple shaping function. It was shown through numerical studies that effective dexterous manipulation was possible through the proposed approach. In fact, in one case study the dexterity measure was brought from 0 (singular arm configuration) to 1.2 (dexterous arm configuration), demonstrating the effectiveness of the approach in achieving high dexterity.

The drawback of this proposed approach was that it allowed the inclusion of only two secondary objectives, which falls short of fully covering the potential secondary objectives that can be demanded from an ROVM system. Furthermore, it did not account for potential failures in the joints and could generate reference joint velocity values beyond the velocity limits. To address these drawbacks, a novel methodology that had a more complete set of objectives with a fault-tolerant property was later proposed. The methodology employed the GPM for the inclusion of the secondary objectives and dynamically weighted each objective based on the Mamdani type fuzzy algorithm. In this method, the high level human pilot set the desired end-effector motion while the low level artificial pilot determined a weighting factor for each objective based on if-then type fuzzy rules that reflect an expert human pilot's knowledge. The secondary objectives considered in this work were:

- Avoiding manipulator joint and velocity limits;
- Avoiding singularity;

- Keeping the end-effector in sight of the on-board camera;
- Minimizing the vehicle motion;
- Minimizing the drag induced disturbance.

In addition, the proposed methodology prevented the occurrence of excessive joint velocities and ensured that the generated velocities were within the allowable range. These unique features set this work apart from the existing literature in which the possibility of faulty manipulator joints or excessive joint velocities was not considered.

A simulation case study of the Falcon™ ROV with a 4-DOF Hydrolek™ manipulator demonstrated that detailed spatial end-effector manoeuvres that were otherwise not possible could be completed in real-time using the proposed scheme.

ROVM Trajectory Controllers: While the redundancy resolution algorithm generates reference state values for the ROVM system, it falls on the controller to command the on-board actuators to realize the commanded motion. To this end, two different model-based controller approaches were proposed for the control of ROVM systems, namely, chattering-free sliding mode control and robust adaptive sliding-mode controller coupled with adaptive PID control layer, both of which were the extension of the ROV control approaches to ROVM systems. As discussed earlier in the ROV research section, the stringent conditions on the Lyapunov stability of the chattering-free sliding-mode controller were relaxed by using a continuous, adaptive term in the robust adaptive sliding-mode controller version. The new adaptive term constantly estimates the upper bound on the difference between the known and the unknown dynamics as opposed to the "true" value of this difference, as is the case in the chattering-free sliding mode control. This modification makes the controller more robust in case of abrupt ROVM motions and

provides asymptotic stability, provided that the two-norm of the lumped uncertainty vector is upper bounded.

In addition, the robust adaptive sliding-mode controller is robust to the parameter drift problem, which makes it a more suitable option for ROVM paradigm. This is because the parameter drift problem is more likely to occur in such systems due to the existence of the manipulator. The high-bandwidth dynamic characteristic of the manipulator motion may induce oscillations within the system, which in turn may excite the unmodelled dynamics. As such, the adaptation occurs based on these meaningless oscillations in the system, causing the parameter drift, or uncontrolled growth in the adaptive parameters, to occur. Moreover, the existence of the 6-DOF joystick makes this scenario more likely to happen since possible abrupt changes in the pilot's demand may induce such oscillations in the system. In addition, the 6-DOF joystick introduces noise into the system, which may also lead to the parameter drift problem. These issues are usually ignored in the relevant ROVM literature since the existing research is mostly confined to the controlled conditions of simulation studies. In this research, however, this problem is not overlooked since hardware, the 6-DOF joystick, is interfaced with the redundancy resolution and control scheme. Adding the robustness to the parameter drift problem, however, provides only uniformly ultimately bounded controller performance.

For the model-based part of both controllers, the quasi-Lagrange approach was utilized for the dynamic modelling of the ROVM system. The fuzzy based multi-objective redundancy resolution algorithm was coupled with both controllers and numerous time domain numerical simulation studies were performed to assess the performance of each controller. It was concluded that the robust adaptive sliding-mode controller coupled

with adaptive PID control layer provides better controller performance in its state trajectory following capability. In fact, it was shown that the maximum end-effector trajectory following performance was reduced by 33% with the use of the robust adaptive sliding-mode controller coupled with adaptive PID control layer in comparison to the chattering-free sliding mode control.

A New HMI for the Coordinated Control of Small, Inspection-Class ROVM Systems: Execution of any of the strategies relies on an improved HMI to gather the desired end-effector motion. To this end, the RSI 6-DOF joystick, an OpenGL based 3-D graphical display, and a new GUI were incorporated into the proposed HMI. While the joystick generated reference end-effector motion, the 3-D graphical display emulated the visual cues provided to the pilot during actual ROVM operations. As for the GUI, it provided a set of tools for effective use of the proposed redundancy resolution algorithm by giving the pilot a series of operation modes. Furthermore, the new GUI also allowed the pilot to operate the ROVM system using the conventional one-joint-at-a-time with the optional station-keeping mode. Using the developed HMI, hardware–in–the–loop simulations were performed to evaluate the performance of the proposed redundancy resolution schemes for a broad range of ROVM operation scenarios.

Finally, an integrated ROVM operation strategy, comprising a multi-objective fault tolerant redundancy resolver that processes those inputs, a non-linear controller, a fault-tolerant thrust allocator accounting for real limits in ROV manoeuvrability and an advanced HMI has not been presented before. Furthermore, none of the past works include hardware to generate end-effector desired velocities reflecting the tendencies of a

human pilot. Therefore, the proposed coordinated control strategy for ROVM fills the void in the existing literature.

4.2 Future Work

4.2.1 ROV Research

In the experimental ROV control research of the dissertation, the thruster dynamics were neglected. However, it is known for a fact that the thruster dynamics have a significant influence on the closed-loop system behaviour; they limit the achievable closed-loop system bandwidth and leads to a limit cycle. Therefore, a strategy could be applied to compensate for thruster related uncertainties in an effort to improve the controller performance. This strategy may involve a more thorough thruster characterisation approach than the one presented herein in which many assumptions were made including zero effect of incoming water velocity in the thruster shroud.

Furthermore, more research effort could be put into estimating external forces on the ROV such as tether forces. Recently, neural networks have shown great promise in the realm of nonlinear control because of their universal approximation and learning capability and could potentially be used to estimate these disturbances. In particular, radial basis function neural networks could be a good candidate for this purpose, as they can uniformly approximate complex functions as long as a sufficient number of neurons are employed and the approximated functions are smoothly varying. Therefore, such a neural network can be utilized to improve the knowledge of the system dynamics in an effort to achieve better controller performance.

On the guidance and navigation side, the obstacle avoidance ability into the guidance algorithm can be incorporated for a reliable autonomous operation.

4.2.2 ROVM Research

In terms of the ROVM research, the derived fuzzy logic based redundancy resolution scheme could be extended such that the system dynamics can be exploited within the redundancy resolution to ensure the high performance of the controller using the intrinsic redundancy. This step involves linking the system dynamics with the redundancy resolution in order to determine a preferable system pose for the on-board controller.

Currently, the subsea research lab has a 5-function Hydrolek™ HLK 43000 hydraulic manipulator, which is presently under retrofitting for simultaneous joint motion capability since the current system does not have the ability to achieve such motion. Once the retrofitting is complete, an experimental set-up involving the Falcon™ ROV equipped with a 5-function Hydrolek™ manipulator will be established to implement the proposed methodologies, namely, the fuzzy based redundancy resolution algorithm coupled with the robust adaptive sliding-mode controller coupled with adaptive PID controller and the 6-DOF joystick as the input device. In addition, the developed GUI will be used for effective ROVM operation in the implementation phase. Once up and running, the ROVM system will be used as a test-bed to evaluate the proposed approaches.

One future work topic that could substantially increase the ability of the human pilot to effectively perform subsea tasks would be implementing an haptic interface for the end-effector motion. Towards the implementation of such an interface, there are two necessary stages. In the first stage, an haptic device, such as the commercially-available Phantom™, could be interfaced with the existing ROVM simulator obtained from the current research program. While the ROVM simulator provides a computer-generated synthetic environment with which a human pilot can interact to perform virtual tasks, the

haptic device not only allows the pilot to manipulate objects in this virtual world, but also reflects reaction forces to the user. These reaction forces lead to haptic perception of virtual objects in the human brain. Through experiencing the touch and feel of virtual objects, ROV pilots would be better trained for sophisticated subsea missions in a more realistic virtual setting.

In the second phase, in lieu of virtual objects, the pilot would manipulate real objects with force feedback reflected on the haptic device through force sensors planted on the end-effector. As such, the human pilot would be able to develop a feel of the workpiece and perform the task more precisely with the tactile information.

Another future work avenue is the force control of ROVM systems, which involves controlling the end-effector forces exchanged with the underwater environment. Most ROVM applications require the end-effector to move to a specific location and then apply specific forces. During such a task, the end-effector first moves in an unconstrained environment until the first contact occurs with the workpiece and then stays in transient contact until the complete physical contact takes place. Once in full contact, the end-effector is in constrained motion and is required to follow a force trajectory. This is a twofold problem: first, the transient motion must be stable, and second, the end-effector must follow a desired force trajectory while in full contact. To have stable transient contact, an impedance control could be implemented in which the mechanical impedance of the manipulator and the environment are controlled. As for the force trajectory following, a hybrid position/force controller theory would provide a means to guide the forces when the end-effector is in constraint motion due to the physical contact.

Bibliography

[1] M. W. Dunnigan, and G. T. Russell, “Evaluation and reduction of the dynamic coupling between a manipulator and an underwater vehicle,” IEEE Journal of Oceanic Engineering, vol. 23, no. 3, pp. 260-273, July 1998.

[2] G. Grenon, P. An, S. Smith, and A. Healey, “Enhancement of the inertial navigation system for the Morpheus autonomous underwater vehicles,” IEEE Journal of Oceanic Engineering, vol. 26 no. 4, pp. 548-560, October 2001.

[3] K. Kawaguchi, H. Momma, and R. Iwase, “VENUS PROJECT-submarine cable recovery system,” 1998 Int. Symp. on Underwater Technology (Cat. No.98EX101), pp. 448-452, 1998.

[4] J. Delaney, P. Beauchamp, M. McNutt, C. Barnes, A. Chave, and J. Madden, “Project NEPTUNE: an interactive, regional cabled ocean observatory in the Northeast Pacific,” Oceans Conf. Record (IEEE), vol. 2, pp. 1038-1042, 2003.

[5] M. M. R. Best, B. D. Bornhold, S. K. Juniper, and C. R. Barnes “NEPTUNE Canada regional cabled onservotary: Scienceplan, in Proceedings of OCEANS 2007 MTS/IEEE, Sep 29-Oct 4,Vancouver, BC, Canada

[6] G. Conte and A. Serrani, “Robust control of a remotely operated underwater vehicle,” Automatica, vol. 34, no. 2, pp. 193-198, 1998.

[7] T. 1. Fossen and S. 1. Sagatun Adaptive control of nonlinear systems, “A case study of underwater robotic systems,” Journal of Robotic Systems, pp. 392-342, 1991.

[8] L. Hsu, R. R. Costa, F. Lizarrade, and J. P. V. Soares da Cunha, “Dynamic positioning of remotely operated underwater vehicles,” IEEE Robotics and Automation Magazine, vol. 7, no. 3, pp. 21–31, September 2000.

[9] G. Antonelli, S. Chiaverini, N. Sarkar and M. West, “Adaptive control of an autonomous underwater vehicle: experimental results on ODIN,” IEEE Transactions on Control Systems Technology, vol. 9, no. 5, pp. 756–765, 2001.

[10] G. Antonelli, F. Caccavale, S. Chiaverini, and G. Fusco, “A novel adaptive control law for underwater vehicles,” IEEE Transaction on Control Systems Technology, vol. 11, no. 2, pp. 221–232, March 2003.

[11] G. Antonelli, Underwater Robots: Motion and Force Control of Vehicle-Manipulator Systems, Springer-Verlag, 2003.

[12] K. R. Goheen and R. Jeffreys, “Multivariable self-tuning autopilots for autonomous and remotely operated vehicle,” IEEE Journal of Oceanic Engineering, vol. 15, no. 3, pp. 144-151, July 1990.

[13] K. S. Narendra, and A. M. Annaswamy, Stable Adaptive Systems, Prentice-Hall, 1989.

[14] D. Yoerger, and J. J. E. Slotine, “Robust trajectory control of underwater vehicles,” IEEE Journal of Oceanic Engineering, vol. OE-10, pp. 462-470, October 1985.

[15] J. J. Slotine and J. Coetsee, “Adaptive sliding controller synthesis for non-linear systems,” International Journal of Control vol. 43, pp. 1631-1651, 1986.

[16] A. Healey and D. Lienard, “Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles,” IEEE Journal of Oceanic Engineering, vol. 18, no. 3, pp, 327-339, 1993.

[17] H. Elmali and N. Olgac, “Theory and implementation of sliding mode control with perturbation estimation,” In: Proceedings of IEEE International Conference on Robotics and Automation, Nice, France, vol. 3, pp. 2114-2119, 1992.

[18] M.S. Chen, Y. R. Hwang, and M. Tomizuka, “A state-dependent boundary layer design for sliding mode control,” IEEE Transactions on Automatic Control, vol. 47, no. 10, pp. 1677-1681, 2002.

[19] P. Debitetto, “Fuzzy logic for depth control of unmanned undersea vehicles,” In: Proceedings of AUV Symposium, Cambridge, MA, USA, pp. 233-241, 1994.

[20] N. Kato, Applications of fuzzy algorithm to guidance and control of underwater vehicle. Underwater Robotic Vehicles: Design and Control, J. Yuh, (Ed.), Albuquerque: TSI Press, 1995.

[21] K. Ishii and T. Ura, “An adaptive neural-net controller system for an underwater vehicle,” Control Engineering Practice, vol. 8, pp, 177-184, 2000.

[22] V. Kodogiannis, “Direct adaptive control of underwater vehicles using neural networks,” Journal of Vibration and Control, vol. 9, pp. 605-619, 2003.

[23] W. J. Pepijn, F. Colin van de Ven. and T. Daniel, “Neural network control of underwater vehicles,” Engineering Applications of Artificial Intelligence, vol. 18, pp. 533-547, 2005.

[24] T. Fossen, Guidance and Control of Ocean Vehicles, Wiley, 1994.

[25] E. Omerdic, and G. Roberts, “Thruster fault diagnosis and accommodation for open-frame underwater vehicles,” Control Engineering Practise, vol. 12, pp. 1575-1598, 2004.

[26] N. Sarkar, T. K. Podder and G. Antonelli, “Fault-accommodating thruster force allocation of an AUV considering thruster redundancy and saturation,” IEEE Transactions on Robotics and Automation, vol. 18, no. 2, pp. 223-231, 2002.

[27] W. Durham, “Constraint control allocation,” Journal of Guidance, Control and Dynamics, vol. 16, no. 4, pp. 717-725, 1993.

[28] G. Antonelli, and S. Chiaverini, “Fuzzy redundancy resolution and motion coordination for underwater vehicle-manipulator systems, “IEEE Transactions on Fuzzy Systems, vol. 11, no. 1, pp. 109-120, 2003.

[29] S. Chiaverini, “Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators, “IEEE Transactions on Robotics and Automation, vol. 13, pp. 398-410, June 1997.

[30] A. A. Maciejewski, and C. A. Klein, “Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments,” Int. Journal of Robotics Research, vol. 4, no.3, pp. 109-117, 1985.

[31] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, “Task-priority based redundancy control of robot manipulators,” Int. Journal of Robotics Research, vol. 6, no. 2, pp. 3-15, 1987.

[32] G. Antonelli and S. Chiaverini, “Task priority redundancy resolution for underwater-manipulator systems,” 1998 IEEE Int. Conf. Robotic. Autom., pp. 768-733, May 1998.

[33] N. Sarkar and T. K. Podder, “Motion coordination of underwater vehicle-manipulator systems subject to drag optimization,” Proc. 1999 IEEE Int. Conf. on Robotic. Autom., pp. 387-392, 1999.

[34] G. Antonelli and S. Chiaverini, “Fuzzy redundancy resolution and motion coordination for underwater vehicle-manipulator systems.” IEEE Transactions on Fuzzy Systems, vol.11, no 1, pp. 109-120, 2003.

[35] T. K. Podder. and N. Sarkar, “Unified dynamics–based motion planning algorithm for autonomous underwater vehicle-manipulator systems,” Robotica, vol. 22, pp. 117-128, 2004.

[36] C. H. dos Santos, R. Guenther, D. Martins, and E. R. De Pieri, “Virtual kinematic chains to solve the underwater vehicle-manipulator systems redundancy,” J. of the Brazilian Society of Mechanical Sciences and Eng., vol. 28, no. 3, pp. 354-361, 2006.

[37] C. H. dos Santos, G. Bittencourt, and R. Guenther, “Motion coordination for underwater vehicle-manipulator systems using a fuzzy hybrid strategy,” Int. Conf. on Intell. Robot. and Syst., Beijing, China, pp. 3018-3023, October 2006.

[38] G. Antonelli, and S. Chiaverini, “Adaptive tracking control of underwater vehicle-manipulator systems,” In: Proceedings of Conference on Control Applications, Trieste, Italy, pp. 1089-1093, 1998.

[39] De Wit C. Canuscas, Diaz D. Olgun and M. Perrier, “Nonlinear control of an underwater vehicle/manipulator system with composite dynamics,” IEEE Transactions on Control Systems Technology vol. 8, pp. 948-960, 2000.

[40] B. Xu, S. R. Pandian, and F. Petry, “A sliding-mode fuzzy controller for underwater vehicle-manipulator systems,” In Proceedings of 2005 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS’ 2005), Ann Arbor, MI., 2005.

[41] J. J. Slotine and S. Shastry, 1983. Tracking control of nonlinear system using sliding surfaces, International Journal of Control, vol. 38, no. 2, pp. 465-492, 1983.

[42] J. M. Hollerbach and D. M. Lokhorst, “Closed-Loop kinematic calibration of the RSI 6-DOF hand controller,” IEEE Transactions on Robotics and Automation, vol. 11, no. 3, pp. 352-359, 1995.

[43] J. Zand, “Enhanced Navigation and Tether Management of Inspection Class ROVs,” Master’s. Thesis, University of Victoria, 2010.

[44] L. Notash, and R. P. Podhorodeski, “Complete forward displacement solutions for a class of three-branch parallel manipulators,” Journal of Robotic Systems, vol. 11, pp 471-485, 1994.

[45] P. Sobejko, Implementation of Redundantly-Sensed Parallel-Manipulator-Based 6-DOF Joysticks. M.A.Sc. Thesis, University of Victoria, Department of Mechanical Engineering, December 2002.

[46] S. Soylu, B. J. Buckham and R. P. Podhorodeski, “Using articulated-body algorithm within sliding mode control to compensate dynamic coupling in underwater manipulator systems,” CSME Transactions, vol. 29, no. 4, pp. 629-643, 2005.

[47] R. Featherstone, “The calculation of robot dynamics using articulated-body inertias,” Int. Journal of Robotics Research, vol. 2, no. 1, pp. 13-30, 1983.

[48] K. Zhuo, Essentials of Robust Control, Prentice-Hall, 1998.

[49] P. Gahinet, A. Nemirovski, A. Laub and M. Chilali, The LMI Control Toolbox. The MathWorks. Inc., 1995; also in Proc. CDC, 1994, pp. 2038-2041.

[50] B. J. Buckham, Steinke, B., Prabhakar, S., ‘‘Evaluation of the ROPOS liveboating configuration using dynamic simulation,’’ in Proceedings of CANCAM 2005 (20th Canadian Congress of Applied Mechanics), Montreal, QC, Canada.

[51] T. W. McLain and S. M. Rock, “Experimental measurement of ROV tether tension”, in Proceedings of 10th Ann. Subsea Interv. Conf. and Expo., San Diego, 1992.

[52] B. J. Buckham, “Dynamics modelling of low-tension tethers for submerged remotely operated vehicles,” Ph.D. Thesis, University of Victoria, 2003.

[53] S. McMillan, D. E.Orin, and R. B. McGhee, “Efficient dynamic simulation of an underwater vehicle with a robotic manipulator,” IEEE Transactions on Systems, Man, Cybernetics, vol. 25, pp. 1194-1206, August 1995.

[54] J. J. Slotine, W. Li, Applied Nonlinear Control. Prentice-Hall, Englewood Cliffs, NJ, 1991.

[55] M. Zeinali and L. Notash, “New approach to design: a nonlinear robust adaptive controller for manipulators,” in Proceedings of CSME Forum, London, Canada, 2004.

[56] M. Zeinali, and L. Notash, “Fuzzy logic-based adaptive robust control for parallel manipulators,” in Proceedings of 12th World Cong Mechanism and Machine Sci, Besançon, France, 8 pp, 2007.

[57] F. L. Lewis, K. Liu, and A. Yesildirek, “Neural-net controller with guaranteed tracking performance,” IEEE Trans. Neural Networks, vol. 6, no. 3, pp. 703–715, 1995.

[58] F. Sun, Z. Sun, Y. Zhu and W. Lu, “Stable neuro-adaptive control for robots with the upper bound estimation on the neural approximation errors,” Journal of Intelligent and Robotic Systems,” vol. 26, pp. 91-100, 1999.

[59] P. A. Ioannou and P. V. Kokotovic. Adaptive Systems with Reduced Models. New York: Springer-Verlag, 1983.

[60] F. W. Lewis, S. Jagannathan, A. Yesildirek, Neural Network Control Of Robot Manipulators And Non-Linear Systems, Taylor&Francis, 1999.

[61] F. J. Lin, P. H. Shen, and S. P. Hsu, “Adaptive backstepping sliding-mode control for linear induction motor drive,” IEE Proc Electric Power Applications, vol. 149, no. 3, pp 184–194, 2002.

[62] M. Zeinali, and L. Notash, “Robust adaptive neural fuzzy controller with model uncertainty estimator for manipulators,” CSME Transactions, vol 28, no 2A, pp 197–219, 2004.

[63] M. Zeinali, and L. Notash, “Chattering-free sliding mode control design using fuzzy modelling and implementation on robot manipulators,” in Proceedings of the CSME Forum, Calgary, Canada, 2006.

[64] M. Caccia and G. Veruggio, “Modeling and Identification of Open-Frame Variable Configuration Unmanned Underwater Vehicles,” IEEE Journal of Oceanic Engineering, vol. 25, no.2, 2000.

[65] A. Liegeois, “Automatic supervisory control of the configuration and behaviour of multibody mechanisms,” IEEE Transactions on Systems, Man, Cybernetics, vol. SMC-7, pp. 868-871, 1977.

[66] D. Driankov, H. Hellendoorn, and M.Reinfrank, An Introduction to Fuzzy Control, Berlin, Germany,Springer-Verlag, 1995.

[67] W Chen, Q, Zhang, Z, Yang, W. A. Gruver, “Optimizing multiple performance criteria in redundant manipulators by subtask-priority control,” In: Intelligent Systems for the 21st Century, IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 2528–2533, 1995.

[68] L. Meirovitch, Dynamics and Control of Structures, John Willey&Sons, 1990.

[69] L. Meirovitch, and T. Stemple, “Hybrid equation of motion for flexible multibody systems using quasi-coordinates,” Journal of Guidance, Control and Dynamics, vol. 18, no. 4, pp. 678-688, 1995.

[70] S. Soylu, B. J. Buckham and R. P. Podhorodeski, “Development of a coordinated controller for underwater vehicle-manipulator Systems,” In Proceedings of OCEANS 2008 MTS/IEEE, Quebec City, QC, Canada, 2008.

Appendix A. MIMO Sliding-Mode and H_∞ Controller Design for Dynamic Coupling Reduction in Underwater-Manipulator Systems

(2009, CSME Transactions, volume 33, issue 4, pages 731-743)

MIMO SLIDING-MODE AND H∞ CONTROLLER DESIGN FOR DYNAMIC COUPLING REDUCTION IN UNDERWATER-MANIPULATOR SYSTEMS

Serdar Soylu, Bradley J. Buckham, Ron P. Podhorodeski
Department of Mechanical Engineering, University of Victoria, P.O. Box 3055, Victoria, B.C., Canada, V8W 3P6
E-mail: serdar@me.uvic.ca; bbuckham@me.uvic.ca; podhoro@me.uvic.ca

Received October 2009, Accepted November 2009
No. 09-CSME-55, E.I.C. Accession 3141

ABSTRACT

In underwater remote vehicle-manipulator system (**URVM**) applications, it is beneficial to have the underwater remote vehicle (**URV**) hold station using its thrusters while a human pilot operates the serial manipulator. This provides a stable platform for the manipulator and eases the pilot's job drastically when current and/or tether disturbances are present. The contribution of this work is twofold: Firstly, the reduction of dynamic coupling in the URVM systems is realized using two robust control techniques namely Sliding-mode control and H_∞ control, and the performance of both controllers in the dynamic coupling reduction problem is reported. Secondly, a new control scheme is proposed that involves both controllers in the control loop. Numerical case studies are developed to demonstrate the effectiveness of the controllers. It is concluded that sliding-mode and H_∞ controller combined approach provides superior dynamic coupling reduction performance.

Keywords: underwater vehicle manipulator system; nonlinear control; dynamic coupling.

LE DESIGN DU CONTRÔLEUR EN MODE-GLISSANT MIMO ET H∞ POUR LA RÉDUCTION DU COUPLAGE DYNAMIQUE DANS LES SYSTÈMES MANIPULATEURS SOUS-MARINS

RÉSUMÉ

Pour des applications de systèmes de manipulateur-véhicule sous-marin télécommandés (MVST), il est bénéfique que le véhicule sous-marin télécommandé (VST) maintienne sa position avec ses propulseurs tandis qu'un pilote humain actionne le manipulateur sériel. Cela fournit une plate-forme stable pour le manipulateur et facilite grandement la tâche du pilote en présence de perturbations dues au courant et/ou au cordon ombilical. Dans ce travail, une réduction du couplage dynamique dans les systèmes MVST est réalisée en utilisant deux méthodes de commande robustes, soit la commande par mode de glissement et la commande H_∞. La performance des deux systèmes de commande pour la réduction du couplage dynamique est présentée. De plus, une nouvelle approche qui comprend les deux systèmes dans la boucle de commande est proposée. Des études de cas numériques sont développées afin de démontrer l'efficacité des systèmes de commande. Il est conclu que l'approche combinant la commande par mode de glissement et la commande H_∞ offre une performance supérieure pour diminuer le couplage dynamique.

Mots-clés : systèmes de manipulateur-véhicule sous-marin télécommandés; contrôle non linéaire; couplage dynamique.

INTRODUCTION

In many URV applications, a master-slave configuration is used to operate the manipulator. In this configuration, the movement of the smaller master arm driven by an operator on a surface support vessel, is approximately duplicated by the larger slave arm relative to the URV frame of reference [1]. However, when the movement is replicated by the larger-slave arm, a force is exerted on the vehicle through dynamic coupling. Since the end effector's position and orientation are a function of accumulated URV motions, this disturbance of the URV state adversely affects the end-effector position and orientation; the slave arm end-effector does not achieve the same final state as the master arm [1]. Nonlinear hydrodynamic effects dominant the coupled vehicle-manipulator dynamics and the hydrodynamics vary greatly with changes in arm and vehicle orientation [2]. Consequently, reducing the negative effect of the dynamic coupling present in the URV manipulator system through the use of on-board controllers becomes a significant issue in obtaining better system performance [3].

Several control methods have been applied to control URV motion. Yoerger and Slotine [4] applied Slotine's [5] sliding-mode methodology to the control problem of the URV. In [4], a series of single-input single-output controllers were used, and the robustness of the control scheme was demonstrated. Cristi *et al.* [6] proposed a control technique that combines the robustness property of the sliding-mode controller with the adaptivity of an adaptive controller. Yuh [7] demonstrated the application of neural networks to an underwater robotic control system, and concluded that a neural network controller provides a robust control technique in case the exact dynamic knowledge is not available. Dunnigan and Russel [8] addressed the dynamic coupling problem for URVM systems, and proposed the sliding-mode approach in an effort to keep a URV stationary. The sliding-mode control methodology was also utilized to solve the URVM dynamic coupling problem in [8] in which an accurate dynamic coupling prediction required inertial measurement units for monitoring the URV motion. In [9], the Articulated-Body Algorithm (**ABA**) predicted the dynamic coupling force expressions based on the feedback of the URV and the manipulator states. It was shown in the same paper that using the ABA not only eliminates the need for an inertial sensor, but it also produces better dynamic coupling reduction performance.

The H_∞ approach is applied in [10] to the problem of precise trajectory controlling. To this end, the nonlinear system dynamics were linearized around an operating point, and standard techniques of H_∞ theory [11] were applied. Conte and Serrani [12] proposed a scheduling of linear H_∞ controllers, and simulation studies were carried out for a broad range of operating conditions. They showed that the H_∞ approach provides a robust control technique for the control of underwater vehicles. In [13], Conte *et al*. considered the problem of decoupling the effect of the umbilical's traction from the position of a URV. It is shown in [13] that the full disturbance decoupling is theoretically not possible within the H_∞ controller synthesis theory. However, the almost disturbance decoupling was proved to be theoretically realizable in the same work. An H_∞ autopilot design for an autonomous underwater vehicle is presented in [14]. The design of sub-controllers is formulated as the problem of minimizing the mixed sensitivity function, and is solved by the Linear Matrix Inequality (**LMI**)-based H_∞ method. In [15], the same problem was solved using reduced-order H_∞ synthesis that produces reduced-order controllers with slight performance degradation compared to LMI-based controllers.

The primary goal of the current proposed research is to compare the ability of sliding-mode H_∞ controllers to attenuate the dynamic coupling in URVM systems, and to propose a control

scheme that combines the two controller methodology to further improve the controller performance.

1. MODELLING

1.1. Vehicle Modelling with Newton-Euler Dynamics

As shown in Fig. 1, the Z axis of the earth-fixed inertial frame $\{E\}$ is assumed in the gravity direction as is consistent with traditional marine mechanics. The URV is modeled as another manipulator link in the serial chain and numbered as *0*. The body-fixed frame is attached to the centre of mass of the URV as shown in Fig. 1. The URV spatial velocity state vector with respect to (**wrt**) its body-fixed frame is considered to be $\boldsymbol{\mu} = [\mathbf{v}_1 \quad \mathbf{v}_2]^T = [p \quad q \quad r \quad u \quad v \quad w]^T$ and the position and orientation state vector of the URV wrt the inertial frame is given by $\mathbf{x} = [\phi \quad \theta \quad \psi \quad X \quad Y \quad Z]^T$. The spatial transformation matrix between the inertial frame and URV's body-fixed frame is given by $\mathbf{T} \in \Re^{6\times 6}$, which includes the angular velocity transformation matrix and the linear velocity transformation matrix. The term **T** can be obtained by the Euler sequence of rotations [16].

The nonlinear equations of motion of the URV in terms of its body-fixed frame are obtained by applying a Newton-Euler dynamics formulation:

$$\mathbf{M}\dot{\boldsymbol{\mu}} + \mathbf{C}(\boldsymbol{\mu})\boldsymbol{\mu} + \mathbf{D}(\boldsymbol{\mu})\boldsymbol{\mu} + \mathbf{g} = \boldsymbol{\tau},\ \dot{\mathbf{x}} = \mathbf{J}\boldsymbol{\mu} \tag{1}$$

where $\mathbf{M} \in \Re^{6\times 6}$ is the inertia matrix including the added mass effect, $\mathbf{C} \in \Re^{6\times 6}$ is the Coriolis matrix that results from the use of a local frame attached to the URV, $\mathbf{D} \in \Re^{6\times 6}$ is the damping matrix, $\mathbf{g} \in \Re^6$ is the gravity matrix, $\mathbf{J} \in \Re^{6\times 6}$ is the kinematic transformation matrix, and finally $\boldsymbol{\tau} \in \Re^6$ is the force and moment vector acting on the system including thrusters forces, tether forces and the manipulator disturbance forces. The definitions of these terms can be found in [16].

1.2. Linearized Model

To be able to implement H_∞ control theory, the nonlinear system must be linearized around a specific operating point. To this end, the linear model can be obtained by linearization of the general expression given in Eq. (1). Since our main goal is to keep the URV stationary in the presence of the disturbances, the operating point or equilibrium point must be **0**. The

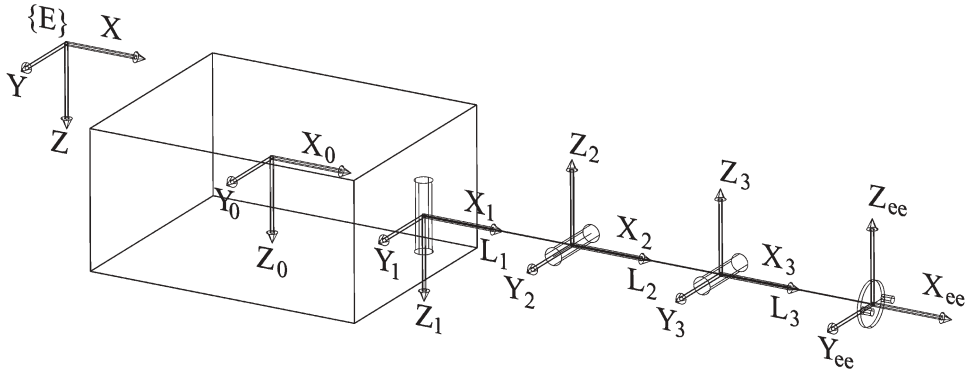


Fig. 1. URVM system.

linearization process around the operating point of **0** yields the following linear time-invariant (**LTI**) system [16]:

$$\begin{bmatrix} \Delta\dot{\boldsymbol{\mu}} \\ \Delta\dot{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} -\mathbf{M}^{-1}(\mathbf{C}+\mathbf{D}) & -\mathbf{M}^{-1}\mathbf{E} \\ \mathbf{J} & \mathbf{0}_{6\times 6} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\mu} \\ \Delta\mathbf{x} \end{bmatrix} + \begin{bmatrix} \mathbf{M}^{-1} \\ \mathbf{0}_{6\times 6} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{0}_{6\times 1} \end{bmatrix} \tag{2}$$

where $\mathbf{E} \in \Re^{6\times 6}$ is the gravity-buoyancy matrix [16]. Equation (2) can be written in a compact form as follows:

$$\dot{\boldsymbol{\xi}} = \mathbf{A}\boldsymbol{\xi} + \mathbf{B}\mathbf{u} \tag{3}$$

To complete the linear model, the output of the system can be given as:

$$\Delta\mathbf{y} = \mathbf{C}\boldsymbol{\xi} \tag{4}$$

2. CONTROL

2.1. MIMO Sliding Mode Control

A multi-input multi-output model-based sliding mode controller allows using one centralized controller instead of many, i.e., one for each controlled axis. The centralized controller coordinates the URV motion in such a way that the desired task is accomplished.

The design of a sliding mode control follows two main steps. First one designs a switching surface **s** to represent a desired system dynamics, then a control law $\mathbf{u} \in \Re^6$ is designed that drives the system states on to the switching surface $\mathbf{s} \in \Re^6$ in a finite time and keeps them on the surface until they hit the desired location on the surface in spite of parameter changes and disturbances [5].

The vector of sliding surfaces **s** is defined as:

$$\mathbf{s} = \Upsilon \begin{bmatrix} {}^{B}_{I}\mathbf{R}\tilde{\boldsymbol{\eta}} \\ \tilde{\boldsymbol{\eta}}_2 \end{bmatrix} - \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \end{bmatrix} = \mathbf{r} - \boldsymbol{\mu} \tag{5}$$

where $\Upsilon \in \Re^{6\times 6}$ is a positive definite matrix containing the control bandwidth value for each sliding surface, ${}^{B}_{I}\mathbf{R} \in \Re^{3\times 3}$ is the rotation matrix from the inertial frame to the body-fixed frame, and the rest of the terms in Eq. (5) are defined as with subscript *d* denoting the desired values:

$$\tilde{\boldsymbol{\eta}}_1 = [\, x_d - x \quad y_d - y \quad z_d - z \,]^T,\ \tilde{\boldsymbol{\eta}}_2 = [\, \phi_d - \phi \quad \theta_d - \theta \quad \psi_d - \psi \,]^T \tag{6}$$

The following control law is implemented based on the work of [17].

$$\mathbf{u} = \mathbf{B}^{\dagger}\left[\mathbf{K}_D\mathbf{s} + \hat{\mathbf{M}}\dot{\mathbf{r}} + \left(\hat{\mathbf{C}} + \hat{\mathbf{D}}\right)\mathbf{r} + \hat{\mathbf{g}} + \mathbf{K}_s sat(\mathbf{s}/\Phi)\right] \tag{7}$$

where $\mathbf{B}^{\dagger}$ is the pseudoinverse of matrix **B**, $\mathbf{K}_D \in \Re^{6\times 6}$ is a positive definite matrix of gains, $\hat{\mathbf{M}}$, $\hat{\mathbf{C}}$, $\hat{\mathbf{D}}$ and $\hat{\mathbf{g}}$ are the estimates of inertia matrix **M**, the vector of Coriolis and centripetal acceleration **C**, the vector of dissipative effects, and gravitational and buoyant force **g**,

respectively. The term $\mathbf{K}_s \in \Re^{6\times 6}$ is a positive definite matrix, and *sat*(**s**) is the saturation function [16].

Theorem 1: *Consider the nonlinear dynamical system described by Eq. (1). If the control law is expressed as Eq. (7) then stability of the closed-loop control system is guaranteed in Lyapunov sense with* $1/2\mathbf{s}^T\mathbf{Ms}$ *being a Lyapunov function candidate.*

Proof: See Antonelli [17].

2.2. H_∞ Control Design

A typical feedback system is demonstrated in Fig. 2 in which **K** is the controller, **G** is the system that is to be controlled, and **d** is a disturbance. The term **r** represents the reference input that is supposed to be tracked by the system, **e** is the error between the reference input and the system output, and finally **u** is the control input produced by the controller **K**.

H_∞ design software programs usually require the system to be expressed in a so-called interconnection system, which is the state-space representation of the augmented system. Such a structure can be easily obtained by adding the performance weighting functions $\mathbf{W_P}$ that captures the closed-loop performance specifications (minimum bandwidth frequency, maximum steady state error, and maximum peak magnitude) and robust weighting function $\mathbf{W_r}$ that characterizes the known frequency dependent knowledge of the modelling uncertainty [11, 18]. The resulting the feedback control loop and the augmented plant **P** are demonstrated in Fig. 3.

So the augmented plant **P** from $[\mathbf{w},\mathbf{u}]^T$ to $[\mathbf{z},\mathbf{v}]^T$ is

$$\begin{bmatrix}\mathbf{z}\\ \mathbf{v}\end{bmatrix} = \begin{bmatrix}\mathbf{P}_{11}(j\omega) & \mathbf{P}_{12}(j\omega)\\ \mathbf{P}_{21}(j\omega) & \mathbf{P}_{22}(j\omega)\end{bmatrix}\begin{bmatrix}\mathbf{w}\\ \mathbf{u}\end{bmatrix},\ \mathbf{u}=\mathbf{K}(j\omega)\mathbf{v} \tag{8}$$

where $\mathbf{z}=[\,\mathbf{z}_1 \quad \mathbf{z}_2\,]^T$ with $\mathbf{z}_1 = w_p\mathbf{u}$ and $\mathbf{z}_2 = w_r\mathbf{Gu}$, $\mathbf{w}=[\,\mathbf{r} \quad \mathbf{d}\,]$, $\mathbf{v}=\mathbf{w}-\mathbf{Gu}$ and

$$\mathbf{P}_{11}(j\omega) = \begin{bmatrix} w_r\mathbf{I} & -w_p(j\omega)\mathbf{G}(j\omega)\\ \mathbf{0} & w_r(j\omega)\mathbf{G}(j\omega)\end{bmatrix}\begin{bmatrix} w_p(j\omega)\mathbf{I}\\ \mathbf{0}\end{bmatrix},\ \mathbf{P}_{12}(j\omega) = \begin{bmatrix} -w_p(j\omega)\mathbf{G}(j\omega)\\ w_r(j\omega)\mathbf{G}(j\omega)\end{bmatrix} \tag{9}$$

$$\mathbf{P}_{21}(j\omega) = [\,\mathbf{I} \quad -\mathbf{G}(j\omega)\,]\mathbf{I},\ \mathbf{P}_{22}(j\omega) = -\mathbf{G}(j\omega)$$

The major goal of the H_∞ design method is to minimize the effect of the worst-case disturbance **w** on the output **z**. The cost function to be minimized is defined as the transfer function between **w** and **z**, and is expressed as $\mathbf{T}_{zw} = \mathbf{P}_{11} + \mathbf{P}_{12}(\mathbf{I}-\mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{KP}_{21}$ yielding:

$$\begin{aligned}\mathbf{T}_{zw} &= \begin{bmatrix}\mathbf{0} & -w_p(j\omega)\mathbf{G}(j\omega)\\ w_r(j\omega) & w_r(j\omega)\mathbf{G}(j\omega)\end{bmatrix} + \begin{bmatrix}-w_p(j\omega)\mathbf{G}(j\omega)\\ w_r(j\omega)\mathbf{G}(j\omega)\end{bmatrix}\mathbf{K}(\mathbf{I}+\mathbf{GK})^{-1}[\,\mathbf{I} \quad -\mathbf{G}(j\omega)\,]\\ &= \begin{bmatrix}(\mathbf{S}(j\omega)w_p(j\omega))\\ (\mathbf{T}(j\omega)w_r(j\omega))\end{bmatrix}\end{aligned} \tag{10}$$

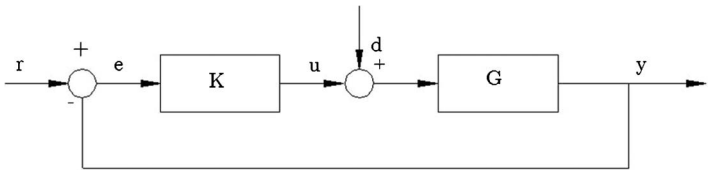


Fig. 2. Feedback control loop. The input disturbance force is given by d.

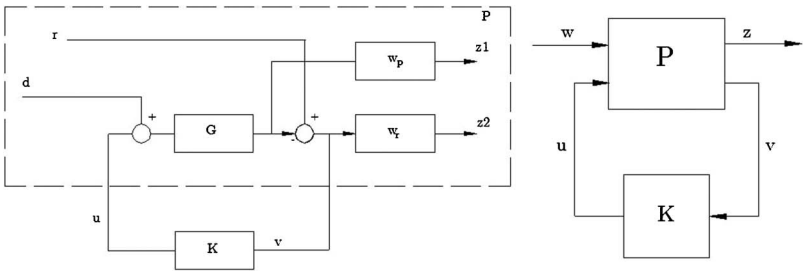


Fig. 3. Feedback control loop (left) and Augmented plant P (right).

In Eq. (10) γ is a constant called nominal performance bound that represents the desired performance level of the closed-loop system and the terms **S** and **T** are called the sensitivity function, and complimentary sensitivity function, respectively [18]. The H_∞ design process seeks a controller **K** such that the closed-loop gain $\mathbf{T}_{zw}$ is minimized in the sense of the H_∞ norm while maintaining internal stability. That is, ensure $\|\mathbf{T}_{zw}\|_\infty < \gamma$ where the term γ represents the achievable performance level of the closed-loop system for a given set of controller design constraints, and is the result of the optimization problem of $\min \|\mathbf{T}_{zw}\|_\infty$ [11].

Consequently, the problem definition can be given as $\|\mathbf{T}_{zw}\|_\infty = \sup_\omega \bar{\sigma}(\mathbf{T}_{zw}(j\omega))$ where $\|\mathbf{T}_{zw}\|_\infty$ is called the mixed-sensitivity cost function, $\bar{\sigma}(.)$ is the maximum singular value and $\sup_\omega$ implies supremum, or least upper bound [11].

In the H_∞ framework, the desired controller performance is given as frequency domain specifications on the various closed-loop transfer functions. These frequency domain specifications form design constraints on the gain response of the closed-loop transfer functions within the H_∞ control problem. The procedure of defining the frequency domain specifications such as tracking performance, disturbance rejection, roll-off, robustness to model uncertainty is called loop shaping [11]. The performance specification and the robustness specification require the following inequalities to hold:

$$\bar{\sigma}\big(\mathbf{S}(j\omega)w_p(j\omega)\big) \le 1, \quad \bar{\sigma}(\mathbf{T}(j\omega)w_r(j\omega)) \le 1 \tag{11}$$

These performance and robustness specifications can be lumped into the following infinity norm inequality:

$$\|\mathbf{T}_{zw}\|_\infty = \left\| \begin{matrix} \big(\mathbf{S}(j\omega)w_p(j\omega)\big) \\ (\mathbf{T}(j\omega)w_r(j\omega)) \end{matrix} \right\|_\infty \le 1 \tag{12}$$

as a result of the following inequalities:

$$\max\left\{\bar{\sigma}\left(\mathbf{S}(j\omega)w_p(j\omega)\right),\ \bar{\sigma}(\mathbf{T}(j\omega)w_r(j\omega))\right\} \le \bar{\sigma}\left(\begin{bmatrix}\mathbf{S}(j\omega)w_p(j\omega)\\ \mathbf{T}(j\omega)w_r(j\omega)\end{bmatrix}\right) \le \left\|\begin{matrix}\mathbf{S}(j\omega)w_p(j\omega)\\ \mathbf{T}(j\omega)w_r(j\omega)\end{matrix}\right\|_\infty \quad (13)$$

2.3. Weighting Function Selection

In the current work, the performance weighting function for the control system was chosen based on the work of [14] as:

$$w_p = 10(l+10)/1000l+1 \quad (14)$$

with l indicating the Laplace domain. This choice leads the steady state error to be below 1%.

The robustness weighting function w_r represents the bound on this uncertainty term. It is assumed that the neglected high frequency dynamics cause as much as 40 dB deviation between the frequency response of the linearized model and the actual system for frequencies over 0.5 Hz. This requires 40dB/dec attenuation after 0.5Hz. The robustness bound satisfying the said conditions was chosen to be:

$$w_r = l^2/10 \quad (15)$$

Note that although w_r is improper, the augmented plant is proper since $w_r(j\omega)\mathbf{G}(j\omega)$ is proper, and that it does not lead to an ill-conditioned state matrix in the realization of the augmented plant **P** [11].

2.4. Solution Procedure

For synthesis of the H_∞ control, transfer matrix **G** in Eq. (10) is replaced by $\mathbf{C}_l(\mathbf{sI}-\mathbf{A}_l)^{-1}\mathbf{B}_l$, which is the transfer matrix of the linear model given in Eqs. (3) and (4). The weighting functions are defined based on the design specifications. There are two common approaches to solve the problem of the minimization of the cost function, i.e., $\min \|\mathbf{T}_{zw}\|_\infty$; the Riccati-Based and the LMI-based. The LMI-based H_∞ control theory [19] will be utilized to synthesize the controller since it applies directly to singular plants. This is the case for our system since the eigenvalue analysis reveals that the linearized model of the vehicle has poles at the origin.

3. SIMULATION RESULTS

3.1. System Overview

These two controllers were applied to the full nonlinear plant, and their performances in the attenuation of the dynamic coupling were evaluated. The URVM system considered in this work is Canadian Scientific Submersible Facility (ROPOS) [20], and all the associated parameters used in this simulation study are given in [20]. The current ROPOS system does not involve manipulators, and hence manipulators from [21] were used instead; these manipulators were chosen due to the similarities in dimensions of their vehicles and the vehicles used in the ROPOS system. In the simulation, a station keeping task is considered for the URV whose initial orientation and position vector is zero. Likewise, the initial configuration of the manipulator is also taken as zero. A randomly selected torque vector of $\tau = [40 \quad 0 \quad 0]$ Nm, which corresponds to the pilot's true desire during task execution and could be any value, is applied for 60 seconds. No joint displacement limit is considered in the current simulation

study. The applied torque vector creates constant disturbance to the URV motion through reaction forces and moments at the junction point between the URV and the manipulator, and in turn causes changes in the position and orientation of the URV. These effects leads to the reduction of task effectiveness, and must be compensated for in order to obtain better system performance in underwater robotic applications. The compensation of the reaction forces and moments will be given by controllers. In underwater applications, these compensation forces and moments are applied to the system by means of thrusters. For the thruster dynamics, a simplified first-order model of the thrusters was implemented. The first-order dynamics successfully model the lag between the controller's commanded thrust and the actual thrust through a time-constant term. The dimensionless time constant term is considered to be 0.07. This value successfully reflects the time lag behaviour of conventional thrusters and is determined based on trial and error. In addition, it will be assumed that commanded thrusts are applied directly to the center of mass of the URV. In order to model uncertainties in the dynamics parameters, drag and added mass coefficients of the URV differed from the real system by 20%. Success of the controller in the presence of these parameter perturbations will partly address the issue of controller robustness. *MatLab Simulink* software was used to perform the simulation. As an integrator, *ode15s* is used.

To predict the reaction forces and moments at the junction point, the ABA [22] algorithm is used in the simulation studies.

3.2. Sliding-Mode Control

MIMO sliding-mode controller has been synthesized based on Section 2.1. The control law parameters were chosen by trial and error to be: $\Upsilon = [0.5 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.5]^T$ and $\mathbf{K}_D = [10 \quad 10 \quad 10 \quad 10 \quad 10 \quad 10]^T$. Differently from the general approach in which control parameter $\mathbf{K}_S$ is taken as constant, in the current implementation of the MIMO sliding-mode controller, the term $\mathbf{K}_S$ is updated using **SVD** (Singular Value Decomposition) so that the convergence condition given in [17] is guaranteed to be satisfied by the sliding-mode controller during task execution. Boundary layer parameter Φ has been set to 0.2 to prevent the chattering problem.

Dynamic simulations were performed for 60 seconds, and the reaction forces and moments occuring at the junction point between the vehicle and the manipulator are demonstrated in Fig. 4. The MIMO sliding-mode controller tries to compensate these disturbance forces and

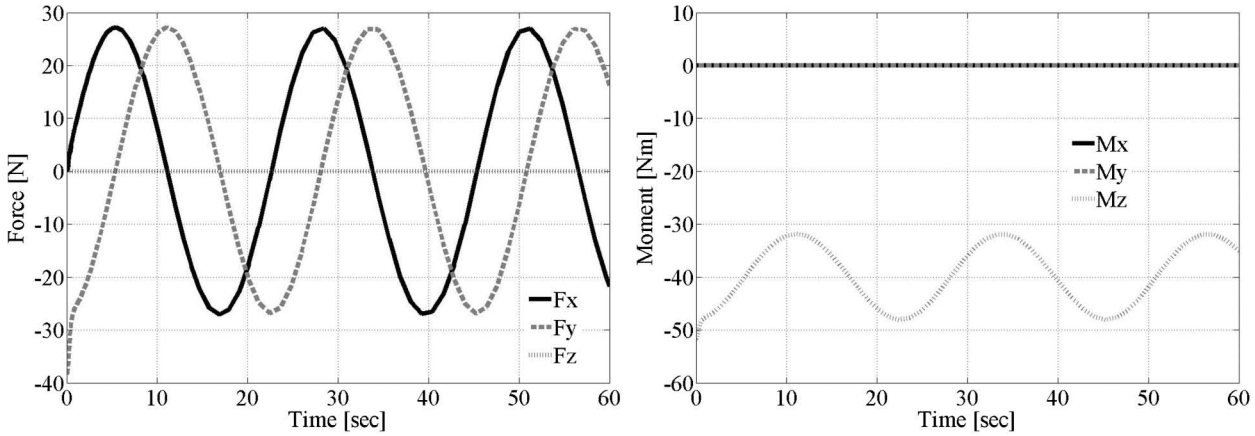


Fig. 4. Forces and moments occur at the connection point due to the manipulator motion.

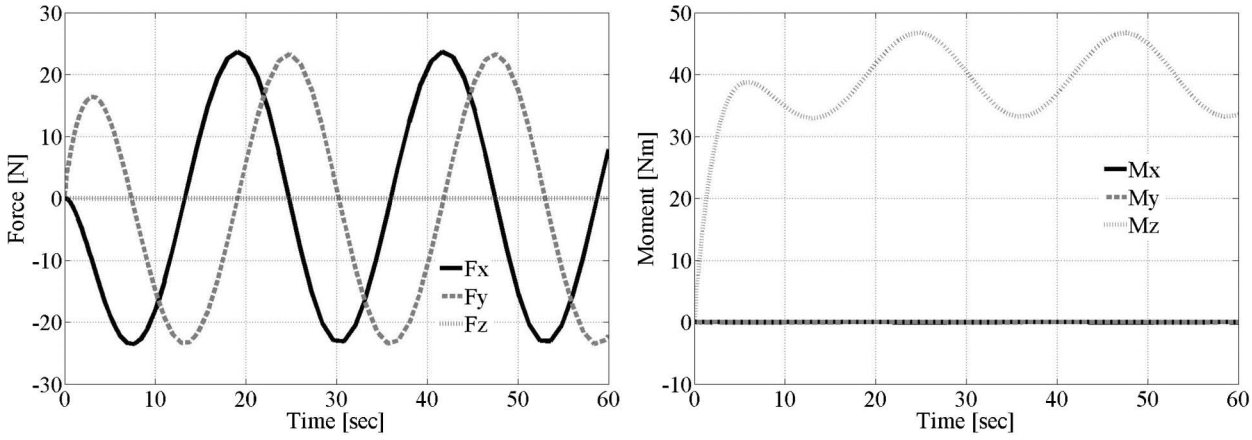


Fig. 5. The compensation forces and moments.

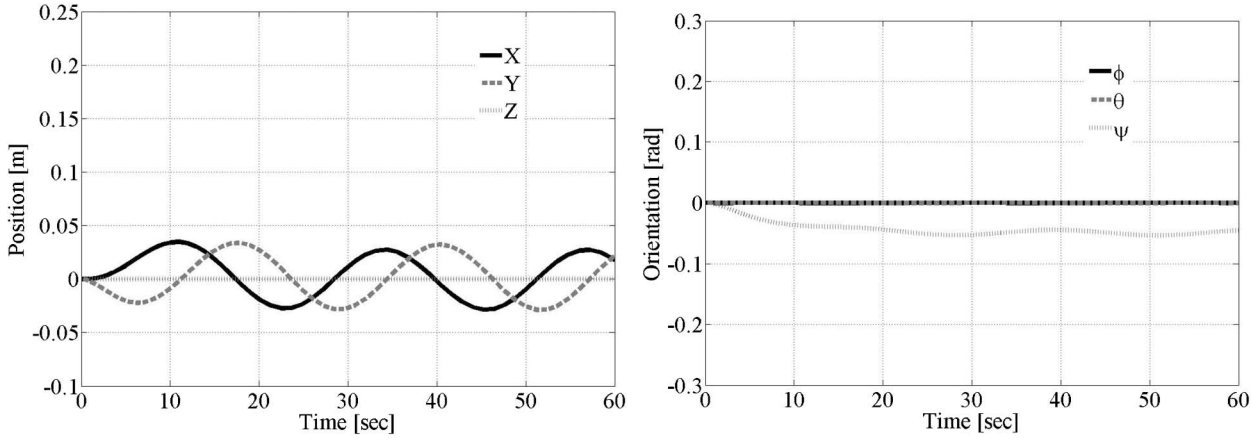


Fig. 6. Above two plots demonstrate the URV's translational (left) and rotational (right) motion. These two graphs are also equivalent to the position and orientation error of the URV, as zero is the reference value for each degree of freedom of the URV.

moments by commanding the thrusters. Figure 5 demonstrates the commanded forces and moments at the center of mass of the URV.

Figure 6 reveals the controller performance in keeping the vehicle still. From Fig. 6, it can be concluded that the MIMO sliding-mode control technique provides satisfactory station-keeping performance since the position and orientation error of the URV under the disturbance of the manipulator motion is small.

3.3. H_∞ Controller Synthesis

The H_∞ control technique was applied to the dynamic coupling reduction problem under the simulation parameters as the previous section. To this end, the dynamic coupling term is included into the general problem setting of H_∞ design as outlined in Section 2.4. MatLab's *LMI Control Toolbox* [23] was used to solve the derived LMI equation, Eq. (10), through the augmented plant given in Eq. (8). The performance specifications were enforced through the weighting functions defined in Section 2.3. The *LMI Control Toolbox* found an optimal 6-input/

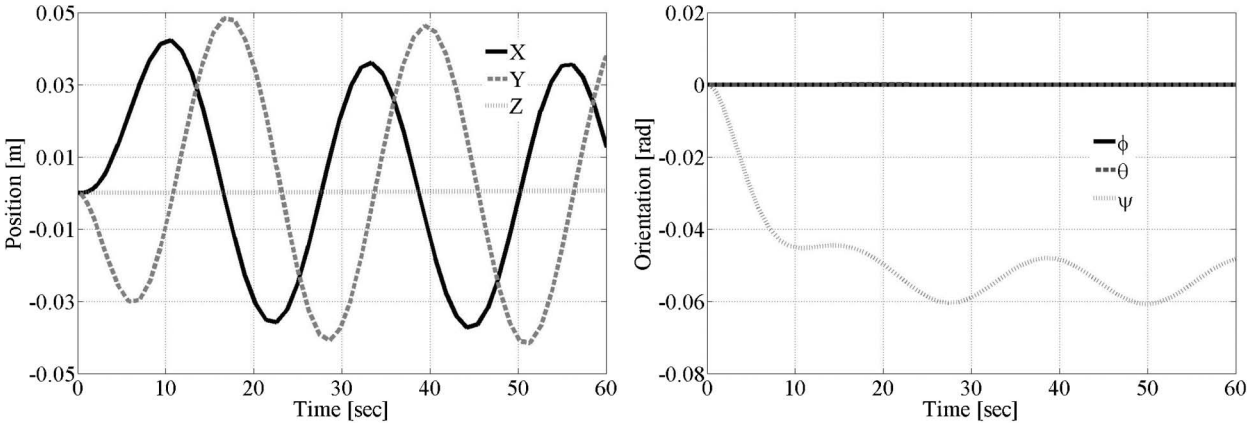


Fig. 7. Position and orientation error of the URV due to the manipulator disturbance forces and moments.

6-output controller of 17^{th} order for given design specifications with nominal performance bound $\gamma = 0.2955$.

Figure 7 shows the translational and rotational position error for the H_∞ controller. Figure 7 reveals that H_∞ control synthesis yields a controller that provides satisfactory station-keeping performance. From the figure, it can be concluded that the sliding-mode controller gives slightly better performance in the attenuation of the dynamic coupling effect compared to the H_∞ controller.

3.4. Sliding-Mode - H_∞ Controller

In order to further improve the disturbance rejection capability, the two approaches can be combined. The proposed feedback control scheme is given in Fig. 8.

In this proposed scheme, there are two loops. The inner loop consists of the sliding mode controller, the disturbances and the URV dynamics and the outer loop consists of an H_∞ controller. In the inner loop, the low-level sliding mode controller works to drive the state of the nonlinear URV system towards the equilibrium state of $\mathbf{0}$. Therefore, the high-level H_∞ controller works on a system whose dynamics model is more consistent with the linearized model used to synthesize the H_∞ controller.

Figure 9 demonstrates the position and orientation error of the URV under the manipulator disturbance. As can be seen from the figure, the combined H_∞-Sliding-Mode controller gives very satisfactory results in keeping the URV's location at zero.

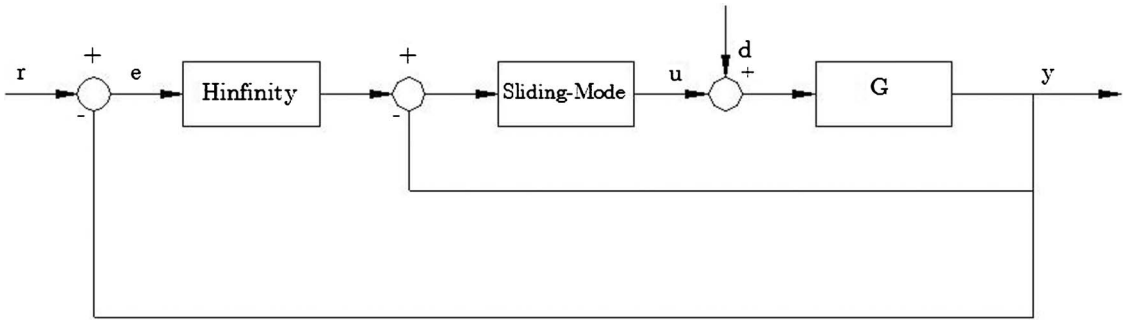


Fig. 8. Proposed H_∞ -Sliding-Mode control scheme.

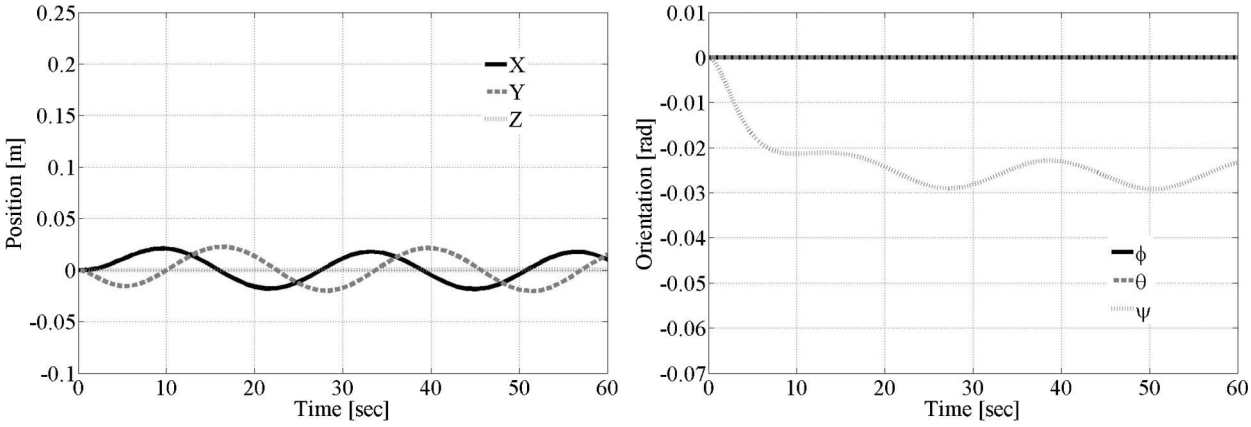


Fig. 9. Position and orientation error of the URV due to the manipulator disturbance forces and moments for the combined H_∞ -sliding-mode controller.

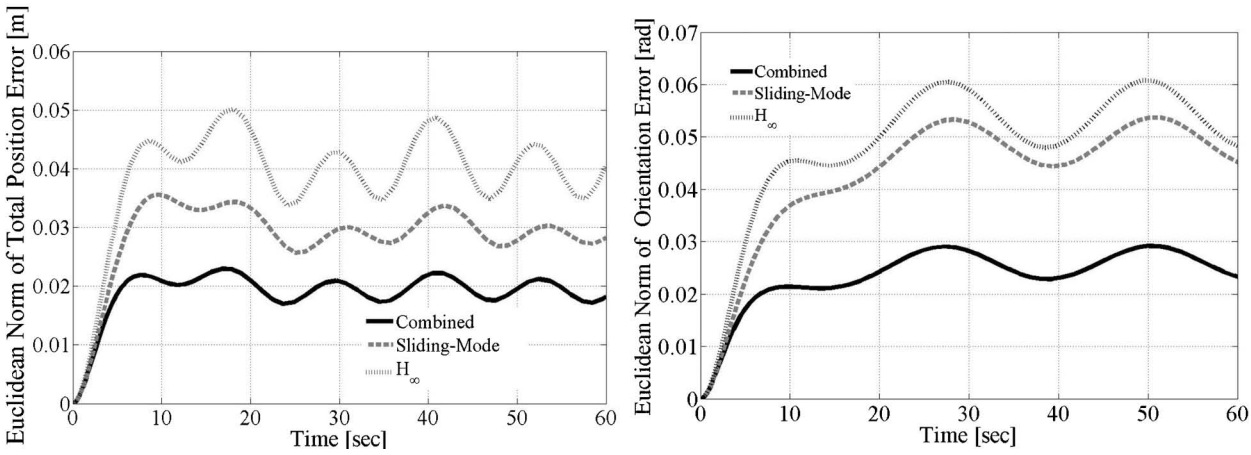


Fig. 10. Euclidean norm of position and orientation error for each controller. As can be seen form the figure that the combined H_∞-Sliding-Mode controller outperforms the other two controllers.

In order to evaluate each controller performance with respect to each other, one should evaluate the Euclidean norm of position and orientation error for each controller. Figure 10 reveals that the combined H_∞-Sliding-Mode controller outperforms the other two controllers. When the area underneath each curve in Fig. 10 is considered, the combined H_∞-Sliding-Mode controller improves the position and orientation error as much as 35% and 47%, respectively, compared to the sliding-mode controller alone, and 49% and 49% compared to the H_∞ controller alone.

4. CONCLUSION

The contribution of this work is twofold. Firstly, the implementation of H_∞ technique to the problem of dynamic coupling reduction in URVMs, and its application to the disturbance mitigation problem is considered to be a unique contribution. Secondly, a new control scheme consisting of a combined sliding-mode and H_∞ controller has been presented. A comparison

case study has been developed to evaluate the performance of each controller in the dynamic coupling reduction problem. Results have demonstrated that the combined H_∞-Sliding-Mode controller outperforms the other two controllers. This improvement can be attributed to the fact that, the high-level H_∞ controller works on a system whose dynamics model is more consistent with the linearized model used to synthesize the H∞ controller due to the low level sliding-mode controller. Although the performance evaluation of each controller has been performed exclusively for the dynamic coupling reduction problem, the proposed control scheme can be easily extended to other URV problems such as waypoint navigation.

ACKNOWLEDGEMENT

The authors wish to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for providing financial support for this research.

REFERENCES

1. Lane, D.M., Dunnigan, M.W., Knightbridge, P.J. and Quinn, A.W., "Planning and control for co-ordination of underwater manipulators," *Proc. IEE Control 91*, pp. 493–498, 1991.
2. Ryu, J.H., Kwon, D.S. and Lee, P.M., "Control of underwater manipulators mounted on an ROV using base force information," *IEEE International Conference on Robotics & Automation*, pp. 3238–3243, May 2001.
3. Yuh, J., "Modeling and control of underwater robotic Vehicles," *IEEE Trans. Syst., Man, Cybern*, Vol. 20, pp. 1475–1483, 1990.
4. Yoerger, D. and Slotine, J.J.E., "Robust trajectory control of underwater vehicles," *IEEE J. Ocean Eng.,* Vol. OE-10, pp. 462–470, Oct 1985.
5. Slotine, J.J.E. and Weiping, L., *Applied Nonlinear Control*, Prentice-Hall, 1991.
6. Cristi, R., Papoulias, F.A. and Healey, A.J., "Adaptive sliding mode control of autonomous underwater vehicles in the dive plane," *IEEE J. Oceanic Eng*., Vol. 15, No. 3, pp. 152–160, July 1990.
7. Yuh, J., "A neural net controller for underwater robotic vehicles," *IEEE J. Oceanic Eng.*, Vol. 15, No. 3, pp. 327–339, 1993.
8. Dunnigan, M.W. and Russell, G.T., "Evaluation and reduction of the dynamic coupling between a manipulator and an underwater vehicle," *IEEE Journal of Oceanic Engineering*, Vol. 23, No. 3, pp. 260–273, July 1998.
9. Soylu, S., Buckham, B.J. and Podhorodeski, R.P., "Using articulated-body algorithm within sliding mode control to compensate dynamic coupling in underwater manipulator systems," *CSME Trans*., Vol. 29, No. 4, pp. 629–643, 2005.
10. Kaminer, I. and Pascoal, A.M., "Control of an underwater vehicle using H_∞ synthesis," *Proc. 30th IEEE CDC*, pp. 2350–2355, 1991.
11. Zhuo, K., *Essentials of robust control*, Prentice-Hall, 1998.
12. Conte, G. and Serrani, A., "Robust control of a remotely operated underwater vehicle," *Automatica,* Vol. 34, No. 2, pp. 193–198, 1998.
13. Conte, G., Serrani, A. and Perdon, A.M., "Rejecting the disturbances due to the umbilical's traction in ROVs' control," *In Proceedings of IEEE Oceans '98*, Nice, France, 1998.
14. Feng, Z. and Allen, R., "H_∞ autopilot design for an autonomous underwater vehicle, " *In Proceedings of IEEE International Conference on Control Applications,* Glaskow, Scotland, UK, 2002.

15. Feng, Z. and Allen, R., "Reduced order H_∞ control of an autonomous underwater vehicle," *Control Engineering Practice*, Vol. 13, pp. 1511–1520, 2004.
16. Fossen, T., *Guidance and Control of Ocean Vehicles*, Wiley, 1994.
17. Antonelli, G., *Underwater Robots: Motion and force control of vehicle-manipulator systems*, Springer-Verlag, 2003.
18. Doyle, J.C. and Stein, G., "Multivariable feedback design-concept for a classical/modern synthesis," *IEEE Trans. Automat. Control*. Vol. AC 26, No. 1, pp. 4–16.
19. Gahinet, P. and Apkarian, P., "A linear matrix inequality approach to H_∞ control," *International Journal of Robust Nonlinear Control,* Vol. 4, pp. 421–448, 1994.
20. Buckham, B., Steinke, D. and Prabhakar, S., "Evaluation of the ROPOS Liveboating Configuration Using Dynamic Simulation," *in Proceedings of CANCAM 2005*(20th Canadian Congress of Applied Mechanics), Montreal, QC, Canada.
21. Podder, T.K., "Dynamics and control of kinematically redundant underwater vehicle-manipulatror systems. Autonomus System Laboratory Technical Report, ASL 98-01, University of Hawaii, Honolulu, Hawaii Adaptive control of an autonomous underwater vehicle, 1998.
22. McMillan, S., Orin, D.E. and McGhee, R.B., "Efficient dynamic simulation of an underwater vehicle with a robotic manipulator," *IEEE Trans. Syst., Man, Cybern.*, Vol. 25, pp. 1194–1206, Aug. 1995.
23. Gahinet, P., Nemirovski, A., Laub, A. and Chilali, M., *The LMI Control Toolbox. The MathWorks. Inc.*, 1995; also in *Proc. CDC*, pp. 2038–2041, 1994.

Appendix B. Dynamics and Control of Tethered Underwater-Manipulator Systems

(2010, In Proceedings of OCEANS 2010 MTS/IEEE)

Dynamics and Control of Tethered Underwater-Manipulator Systems

Serdar Soylu, Bradley J. Buckham, and Ron P. Podhorodeski
IEEE Conference Publishing
University of Victoria, P.O. Box 3055
Victoria, B.C. V8W 3P6 Canada

Abstract- **In this work, the dynamics modelling strategy of a tethered underwater remotely operated vehicle (ROV) coupled with a and spatial manipulator have been studied. With regards to the cable dynamic modelling, it is considered to be a series of lumped point masses connected by linear, massless, visco-elastic springs. In addition, the model accounts for the tether bending and twisting effects. Regarding the manipulator dynamics, the Articulated-Body Algorithm is employed due to its computational efficiency. In order to control the ROV motion under disturbance forces and moments caused by the tether and the manipulator motion, a series of Model-based SISO sliding-mode controllers are implemented and the ABA is used to predict the dynamic coupling force expressions based on the feedback of ROV and the manipulator states. The control gains of the sliding-mode controllers are defined as a closed function of the articulated inertias of the ABA algorithm; leading to time varying gains as opposed to constant as in conventional methods. As a case study, a Saab-Seaeye FALCON™ ROV with a modified Hydrolek™ HLK 43000 manipulator is presented. Numerical simulations are performed to reveal the extent to which the tether dominates the Falcon-manipulator dynamics. It is shown that disturbance forces and moments created by tether motion must be actively compensated using while the ROV is held stationary during manipulator operation. It is also shown that the use of force sensors at the FALCON™'s tether termination can dramatically improve the performance of the series of SISO sliding mode controllers.**

I. INTRODUCTION.

Remotely operated vehicles (**ROV**) equipped with robotic manipulators play an important role in a number of shallow and deep-water missions for marine science, oil and gas, surveys, exploration recovery and military services. The ROV is used as a mobile platform that delivers the necessary robotic tools to a subsea work site. In this capacity, the ROV serves as a base while the manipulator performs a required task. These systems are also equipped with a tether. The tether supplies power to the ROV and facilitates high bandwidth teleoperation of the ROVs robotic end-effector by a human pilot in the deployer vessel.

It has been documented that the hydrodynamic and gravitational forces acting over the tether create internal forces that dominate the ROV motion [1]. Thus, it becomes necessary for a ROV simulation to accurately model the tether's dynamic behaviour under certain conditions. Previous work in the Subsea Robotics Laboratory at the University of Victoria has led to the development of a time-domain simulation of the dynamics of the slender ROV tether subjected to large displacement, as presented in [2]. In the simulation, the lumped mass approach is used. In this approach, the ROV tether is considered to be a series of point masses connected by linear, massless, visco-elastic springs. In addition, the model accounts for the bending effects, which is crucial to be able to accurately predict realistic results for low-tension manoeuvres. The assumption of negligible rotational inertia leads to the derivation of bending forces as a function of the tether curvature at any point along the cable.

Several control methods have been applied to control ROV motion. Yoerger and Slotine [3] applied Slotine's [4] sliding-mode methodology to the control problem of the ROV. In [3], a series of single-input single-output controllers was used, and the robustness of the control scheme was demonstrated. Multivariable sliding-mode controller was proposed by Healey and Lienard [5]. Cristi *et al.* [6] proposed a control technique that combines the robustness property of the sliding-mode controller with the adaptivity of an adaptive controller. In Soylu *et al*. [7], the discontinuous term of a conventional sliding-mode controller was replaced with an adaptive estimate of the dynamic uncertainties. Goheen and Jeffery [8] employed an adaptive controller that does not require precise knowledge of ROV dynamics. Yuh [9] demonstrated the application of neural networks to an underwater robotic control system, and concluded that a neural network controller provides a robust control technique in case exact dynamic knowledge is not available. Dunnigan and Russel [10] dealt with the dynamic coupling problem for ROVM systems, and proposed the sliding-mode approach in an effort to keep a ROV stationary. The sliding-mode control methodology was also utilized to solve the ROVM dynamic coupling problem in [11]. In [11], the Articulated-Body Algorithm (**ABA**) was used to predict the dynamic coupling force expressions based on the feedback of the ROV and the manipulator states. It was shown in the same paper that using ABA not only eliminates the need for an inertial sensor, but it also produces better dynamic coupling reduction performance.

The H∞ approach is applied in [12] to the problem of precise trajectory controlling. To this end, the nonlinear system dynamics were linearized around an operating point, and standard techniques of the H∞ theory were applied. Conte and Serrani [13] proposed a scheduling of linear H∞ controllers. Soylu *et a*l. [14] combined sliding-mode control with H∞ control for a station keeping task.

The ROV dynamics are dominated by hydrodynamic loads and it is difficult to accurately measure or estimate the hydrodynamic coefficients. As such, automatic control of the ROVM requires a robust controller: one that is insensitive to inaccuracies in the dynamic model of the ROVM. It has been shown that the sliding mode approach is an effective technique for the control of ROVM systems [3], [11]. In the present work, the sliding-mode approach is chosen.

978-1-4244-4333-8/10/$25.00 ©2010 IEEE

Tether Modelling

It is known that the forces acting on the tether of a ROV disturb the vehicle position and orientation. Depending on the manoeuvre, these forces can dominate over other forces, including the thrusters' forces, and can hence become governing forces. Therefore, in order to have a reliable dynamic simulator of an ROV system, it is vital to have a dynamic model of the ROV that includes a realistic dynamic model of the tether.

A. Tether Kinematics in a Nutshell

In typical ROV applications, the tether is usually kept in a slack state in an attempt to mitigate the disturbance effect caused by the tether dynamics. Therefore, the tether dynamics to be used in the simulator must accurately capture the dynamics of a slow-tension tether. Recent studies showed that the accurate dynamics of a slack tether must accommodate two important factors: bending and torsional effects. A non-linear dynamic modelling methodology that includes the two aforementioned effects was proposed in [2]. This model is based on a finite-element discretization of the tether continuum. Consider the schematic of the ROV-tether system.

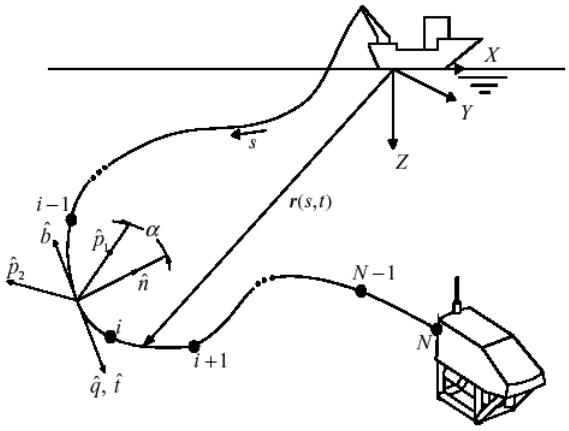


Figure 1: In the finite-element formulation of the cable dynamics, the cable path is discretized by dividing the continuum into N-1 segments with N nodes.

The discretized tether is composed of N-1 segments that are connected to one another through N nodes. The position of the tether with respect to the inertial frame X, Y, Z is defined by the position vector $\boldsymbol{r}$ which is a function of the unstretched curvilinear coordinate $\boldsymbol{s}$ along the cable. The orientation of each visco-elastic tether element is defined by a local coordinate frame $\hat{p}_1, \hat{p}_2, \hat{q}$ aligned with the tether cross-section. The definition of the orientation of this local frame is made through a $Z-Y-X$ Euler angle sequence. The $\hat{p}_1$ axis is normal to the tether element, the $\hat{q}$ axis is normal to the cross-section of the tether and tangent to the increasing tether length, and finally $\hat{p}_2$ completes the right-handed coordinate frame. The orientation of this local reference frame in three-dimensional space varies continuously with the changing location of the cable, and time. Additionally, a frenet frame is defined at the nodes, with the tangential axis $\hat{t}$, aligned with the $\hat{q}$ axis, and the normal axis $\hat{n}$ pointed towards the center of the curvature at the nodes. Term $\hat{b}$ is a bi-normal direction that completes the right-handed coordinate frame. In Figure 2, a differential segment of a continuous tether is seen. Each segment in the tether is a twisted- cubic spline. The state definition of a segment is made via the absolute position vector $\boldsymbol{r}$ and the curvature r'' at the node points on the segment as shown in Figure 2 with the superscript $(\,'')$ denoting absolute second differentiation with respect to $\boldsymbol{s}$. The curvature at the node is defined by the curvature ρ.

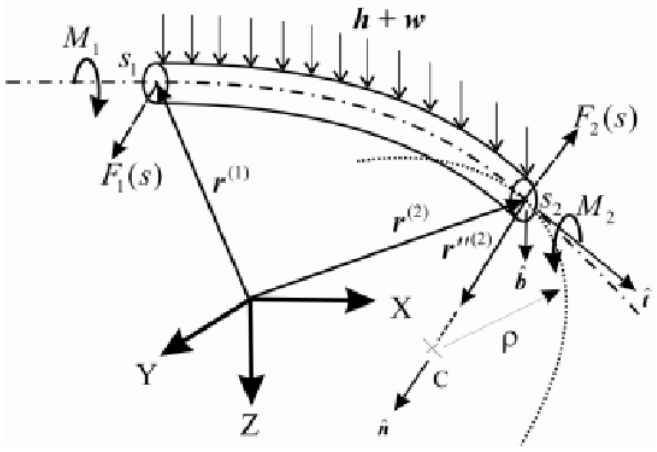


Figure 2: Cable segment between nodes 1 and 2, defined relative to an inertial frame by position vector r. Given the node positions and curvatures, the internal forces and moments shown can be recovered.

B. Tether Solution Process

The segment is subject to internal forces and moments as well as distributed hydrodynamic and gravitational loads. Given the node positions and curvatures, the internal forces and moments can be recovered.

Utilizing the twisted spline geometry and a lumped mass approximation, the node positions along the tether can be defined. The discretized tether is approximated by a series of *N* twisted cubic spline elements. The time-varying state vector defining the cable profile is the concatenation of each node's absolute position and the curvature vectors [2]:

$$Y(t_k) = \left[\left[r_k^{(i)}\right]^T \quad \left[\dot{r}_k^{(i)}\right]^T\right]^T \quad i = 1 \ldots N \qquad (1)$$

For a given initial tether profile (state vector), this vector can be integrated forward in time to obtain the tether profile at t_{k+1}, i.e., $Y(t_{k+1})$. Using the current tether state along with the boundary conditions at node 0 and node N, the set of curvature vectors along the discrete tether, i.e., $\left[r''^{(i)} \ldots r''^{(N)}\right]$, can be obtained.

Using the obtained tether profile, the torsion γ within each element can be calculated. Note that torsion is defined as the

rate of twist of the tether's local frame around the tangential axis incurred by the changing orientation of the osculating plane formed by the $\hat{t}$ and $\hat{n}$ unit vectors at the relevant point. From this definition, it can be said that the torsion corresponds to the twist created by the curvilinear profile of the tether. Having these torsion values combined with the boundary conditions, i.e., one of $\alpha^{(0)}$ or $T^{(0)}$ and one of $\alpha^{(N)}$ or $T^{(N)}$, the torsional deformations through the discrete tether can be calculated. Note that the torsional deformation α and T represent the angle of rotation between the Frenet frame and the principal axes of inertia of the tether cross-section and the applied torque at the boundary nodes, respectively The knowledge of the torsional deformation along the discrete tether allows one to locate the local frame of reference through the discrete tether, i.e., $\left[\alpha^{(i)} \ldots \alpha^{(N)}\right]$.

The global motion equations for the discrete tether are the assembly of each node dynamics and are given by:

$$\left(\left[\mathbf{K}_{\varepsilon}\right]_G + \left[\mathbf{K}_K\right]_G + \left[\mathbf{K}_{\tau}\right]_G\right)\left[\left[r^{(i)}\right]^T \quad \left[r''^{(i)}\right]^T\right]_{i=1\ldots N} + \mathbf{W}_G + \mathbf{H}_G + \mathbf{B}_G = \mathbf{M}_G\ddot{\mathbf{R}} \tag{2}$$

where $R = \left[r^{(0)T} \quad r^{(1)T} \quad \ldots \quad r^{(N)T}\right]^T$ is the set of tether node positions; $\left[\mathbf{K}_{\varepsilon}\right]_G$, $\left[\mathbf{K}_K\right]_G$, and $\left[\mathbf{K}_{\tau}\right]_G$ are rectangular global bending, axial, and torsional stiffness matrices, respectively. The $3\mathrm{N}\times 1$ vectors $\mathbf{W}_G$, $\mathbf{H}_G$, and $\mathbf{B}_G$ are the global hydrodynamic, weight and buoyancy, and external applied loads acting at the node points of the assembly. Finally, $\mathbf{M}_G$ is the assembled lumped mass matrix. Eq. (2) is solved for $\left[\dot{r}^{(i)} \ldots \dot{r}^{(N)}\right]$ to form $\dot{\mathbf{Y}}(t)$, and then the resulting equations are integrated to obtain the tether state vector $\mathbf{Y}(t)$. Details of the structure of the system matrices in Eq. (2) can be found in [2].

Vehicle-Manipulator Modelling

A. ROV Dynamic Modelling within the AB Algorithm

According to the AB algorithm, the nonlinear equations of motion of the ROV are given by [15],[16]:

$$\mathbf{f}_0 = \boldsymbol{I}_0^*\mathbf{a}_0' - \boldsymbol{\beta}_0^*, \tag{3}$$

where $\boldsymbol{I}_0^* \in \mathbf{R}^{6\times 6}$ represents the AB inertia of the whole ROVM, including added mass contributions. The term $\boldsymbol{\beta}_0^* \in \mathbf{R}^6$ is a bias force that contains the hydrodynamic and any potential environmental forces, e.g., drag forces and tether disturbance forces, excluding gravity, acting over the entire ROVM. Details about the calculations of hydrodynamic, drag and fluid acceleration forces can be found in [15]. The components of the vector $\mathbf{f}_0^* \in \mathbf{R}^6$ are the control forces and moments exerted by the thrusters on the ROV. All vectors are expressed in terms of the ROV's body-fixed frame.

Equation (3) can be solved for the biased acceleration of the ROV as:

$$\mathbf{a}_0' = (\boldsymbol{I}_0^*)^{-1}(\mathbf{f}_0 + \boldsymbol{\beta}_0^*) \tag{4}$$

In the simulation, the unbiased acceleration vector of the ROV is needed. The unbiased acceleration, $\mathbf{a}_0$, is:

$$\mathbf{a}_0 = \mathbf{a}_0' + [\mathbf{0}_{1\times 3} \quad {}^0\mathbf{a}_g^T]^T = (\boldsymbol{I}_0^*)^{-1}(\mathbf{f}_0 + \boldsymbol{\beta}_0^*) + [\mathbf{0}_{1\times 3} \quad {}^0\mathbf{a}_g^T]^T. \tag{5}$$

The velocity and unbiased accelerations calculated from Eq. (5) are transformed from body-fixed frame to inertial frame prior to the integration step. This can be done using the following spatial transformations:

$$\dot{\mathbf{x}} = \boldsymbol{T}\mathbf{v}, \quad \ddot{\mathbf{x}} = \boldsymbol{T}\mathbf{a}_0 + \begin{bmatrix} \dot{\boldsymbol{T}}_2 & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix}\mathbf{v}. \tag{6}$$

where $\mathbf{x} = [\phi\ \theta\ \psi\ X\ Y\ Z]^T$ is the position and orientation state vector of the ROV wrt the inertial frame, the term $\boldsymbol{T}$ is the spatial transformation matrix between the inertial frame and ROV's body-fixed frame which includes the angular velocity transformation matrix $\boldsymbol{T_2}$ and the linear velocity transformation matrix $\boldsymbol{T_1}$.The definitions of these matrices can be found in [17].

B. Incorporation of the Tether Dynamics into the AB-Based ROV Dynamics

To couple the ROV and tether dynamics models, the tether bottom boundary position $\mathbf{r}^{(N)}$ is extracted from the vehicle state. The force exerted by the last node N of the tether can be pulled from Eq. (2) The force acting on the ROV at the termination point is the last three components of Eq. (2). Similarly, the moment induced by the tether dynamics on the ROV is given by

$$\mathbf{n}_N = -EI\kappa\big|_{s^{(N)}}\hat{b}^{(N)} - GJ\,\tau\big|_{s^{(N)}}\hat{t}^{(N)} \tag{7}$$

where $\kappa\big|_{s^{(N)}}$ and $\tau\big|_{s^{(N)}}$ are the curvature and the twist at the termination, respectively. The resulting forces and moments on the ROV caused by the tether dynamics are given in [18] as:

$$\mathbf{f}_T = \left[\mathbf{f}_N \quad \mathbf{r}_{N/G}\times\mathbf{f}_N + \mathbf{n}_N\right]^T \tag{8}$$

where $\mathbf{r}_{N/G}$ is the position vector from the center of mass of the ROV to the termination point.

Within the AB dynamic formulation frame, all of the external forces acting on the vehicle can be incorporated into the general scheme through the bias force term $\boldsymbol{\beta}_0^* \in \mathbf{R}^6$ as an additional force [15], [16].

C. Manipulator Modelling

The body-fixed frames are attached to a base joint of each successive link using the modified Denavit-Hartenberg convention [19] and are labelled from n through n. The joint position state vector for the manipulator is defined by

$\mathbf{q} = [q_1 \quad q_2 \quad \dots \quad q_n]^T$ where q_i is the rotation angle of the i^{th} joint. The rotation matrices from link to link can be obtained from homogeneous transforms based on the modified Denavit-Hartenberg parameters.

The force balance equation associating the control force $\mathbf{f}_i$ provided by the manipulator's i^{th} joint with the motion of the outboard articulated links, link i through *3*, is given by:

$$\mathbf{f}_i = \boldsymbol{I}_i^* \mathbf{a}_i' - \boldsymbol{\beta}_i^* . \tag{9}$$

where $\boldsymbol{I}_i^*$ represents the AB inertia of link *i* through *n*. The term $\boldsymbol{\beta}_i^*$ is a bias force that contains the hydrodynamic and environmental forces, excluding gravity, acting on the ROVM. Detailed calculation of $\boldsymbol{I}_i^*$ and $\boldsymbol{\beta}_i^*$ can be found in [15], [16].

The biased acceleration of the i^{th} link is given by:

$$\mathbf{a}_i' = {}^i\boldsymbol{X}_{i-1}\mathbf{a}_{i-1}' + \boldsymbol{\phi}_i \ddot{q}_i + \boldsymbol{\zeta}_i \tag{10}$$

where $\boldsymbol{\zeta}_i$ is the vector of Coriolis and the centripetal accelerations of link i, the term $\boldsymbol{\phi}_i$ maps the contribution of the i^{th} joint motion to the spatial motion of the articulated body, i.e., $\boldsymbol{\phi}_i = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]^T$ for a rotational joint, and ${}^i\boldsymbol{X}_{i-1}$ is a spatial transformation between the *i-1* and *i* coordinate systems. .For simulation purposes, the joint accelerations $\ddot{q}_i, i = 1 \dots n$ are required. To obtain these joint accelerations, Eq. (10) is substituted into Eq. (9) and then force $\mathbf{f}_i$ is projected onto the joint axis according to:

$$\tau_i = \boldsymbol{\phi}_i^T \mathbf{f}_i \tag{11}$$

The resulting equation is solved for the unknown $\ddot{q}_i$ [16]

$$\ddot{q}_i = \left(m_i^*\right)^{-1} \tau_i^* - \left[\boldsymbol{n}_i \left(m_i^*\right)^{-1}\right]^T \left({}^i\boldsymbol{X}_{i-1}\mathbf{a}_{i-1}' + \zeta_i\right), \tag{12}$$

where

$$\boldsymbol{n}_i = \boldsymbol{I}_i^* \boldsymbol{\phi}_i, \quad m_i^* = \boldsymbol{\phi}_i^T \boldsymbol{I}_i^* \boldsymbol{\phi}_i, \quad \tau_i^* = \tau_i + \boldsymbol{\phi}_i^T \boldsymbol{\beta}_i^* . \tag{13}$$

with τ_i being externally applied torque at joint *i*.

D. Dynamic Coupling Expressions

The dynamic coupling effect on the ROV is caused by the reactions forces at the mounting point between the ROV and manipulator as illustrated in Figure 3:

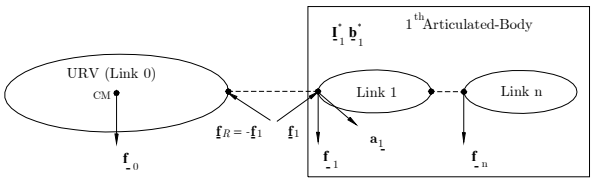


Figure 3: Reaction forces that cause the dynamic coupling effect

As far as the ROV's motion is concerned, this force can be considered to be a force that is exerted by the manipulator on the ROV. Referring to Figure 3the force that is experienced by the link 1 in the ROVM serial-chain can be given according to the articulated-body equations as:

$$\mathbf{f}_1 = \boldsymbol{I}_1^* \mathbf{a}_1^* - \beta_1^* \tag{14}$$

Therefore, the force and moment exerted by the manipulator on the ROV is equal and opposite and can be written as:

$$\mathbf{f}_R = -\mathbf{f}_1 \tag{15}$$

The moment created by this force is found by:

$$\mathbf{N}_R = \begin{bmatrix} 0_{3\times 1} \\ p_{0\to 1} \times \mathbf{f}_{R-f} \end{bmatrix} \tag{16}$$

where $p_{0\to 1}$ is the position vector from the center of gravity to the manipulator mounting point on the ROV, and $\mathbf{f}_{R-f}$ is the force components of $\mathbf{f}_R$. Finally, the spatial force vector acting on the ROV due to the dynamic coupling is given by:

$$\mathbf{D}_{dist} = \mathbf{f}_\mathbf{R} + \mathbf{N}_\mathbf{R} \tag{17}$$

Eq. (17) represents the disturbance effect acting on the ROV. This models the disturbance effect that the controller tries to reject. This effect will be treated as external disturbance by the controllers that are to be synthesized.

III. ROVM SOLUTION PROCESS

For given tether profile and boundary conditions, which are defined by the ROV at node N, the time derivative of tether state vector $\dot{\mathbf{Y}}$ is estimated by solving Eq. (2). This step is followed by finding the reaction forces and moments using Eq. (2) and Eqs. (7) and (8) for the same tether state.

Having obtained the tether disturbance effect on the ROV, the AB algorithm calculates the ROV and joint accelerations. The complete hydrodynamic AB algorithm for the forward dynamics problem is given in [15]. The AB hydrodynamics algorithm consists of three steps. The first step, called forward kinematics, is the computation of the velocity-dependent terms from the ROV to the end effector. The first step also involves the computation of velocity dependent terms of the added mass force, the drag forces, and the fluid acceleration and buoyancy forces using the state vector of the ROVM. The second step, called backward dynamics, involves computing the articulated inertias, from the end effector to the ROV. The third step, called forward acceleration, is the calculation of the accelerations of the ROV, $\ddot{\mathbf{x}}$, based on Eq. (6), of the manipulator, $\ddot{\mathbf{q}}$, based on Eq. (12).

The final step is to concatenate the ROVM state vector $\mathbf{X} = \left[\mathbf{x}^\mathbf{T} \quad \dot{\mathbf{x}}^\mathbf{T} \quad \mathbf{q}^\mathbf{T} \quad \dot{\mathbf{q}}^\mathbf{T}\right]^T$ and the tether state vector $\mathbf{Y}$ in preparation for the integration process. The resulting vector is passed into the integration routine to update the state vector of combined tether-ROVM system.

IV. CONTROL

This section proposes a control law based on [3]-[4]. The control law design method involves breaking the control system up into a set of low-order controllers so that each axis of the ROV has a separate controller. Each controller includes cross-coupling terms due to the multiaxis motion of the ROV and the manipulator dynamic effects on the ROV. These coupling effects are treated as disturbances within each controller.

The dynamic model of the ROV within the controller is derived from Eq.(5). The sliding mode control strategy seeks to control motion of the ROV as seen by an observer in the ROVs body fixed frame. Since Eq. (5) gives the absolute ROV accelerations it is necessary to remove the component of this result that is due to the rotating body fixed frame. As such, the controller model is given by:

$$\mathbf{a} = (\boldsymbol{I}_0^*)^{-1}(\mathbf{f}_0 + \boldsymbol{\beta}_0^*) + \left[\mathbf{0}_{1\times 3} \quad {}^0\mathbf{a}_g^T\right]^T - \left[\mathbf{0}_{1\times 3} \quad (\boldsymbol{\omega}_0 \times \boldsymbol{v}_0)^T\right]^T \quad (18)$$

Eq. (18) can be rearranged as:

$$\mathbf{a} = \left(\boldsymbol{I}_0^*\right)^{-1}\boldsymbol{\beta}_0^* + \left[0_{1\times 3} \quad {}^0\mathbf{a}_g^T\right]^T - \left[0_{1\times 3} \quad (\boldsymbol{\omega}\times\boldsymbol{v}_0)^T\right]^T + \left(\boldsymbol{I}_0^*\right)^{-1}\mathbf{u} \quad (19)$$

Note that in Eq. (19), $\mathbf{f}_0$ is replaced by the control force vector $\mathbf{u}$. The dynamic equation of each degree of freedom of the vehicle can be written based on Eq. (19) as:

$$a_i = f_i + \sum_{j=1}^{6}\left[\boldsymbol{I}_0^*\right]_{ij}^{-1} u_j \;\; i,\; j = 1\ldots 6 \quad (20)$$

where a_i, and u_i are the ith component of vector $\mathbf{a}$, $\mathbf{u}$, respectively. The term f_i is the ith component of the vector defined by the expression $(\boldsymbol{I}_0^*)^{-1}\boldsymbol{\beta}_0^* + \left[\mathbf{0}_{1\times 3} \quad \mathbf{a}_g^T\right]^T - \left[\mathbf{0}_{1\times 3} \quad (\boldsymbol{\omega}\times\boldsymbol{v}_0)^T\right]^T$, and $\left[\boldsymbol{I}_0^*\right]_{ij}^{-1}$ is the ij^{th} component of the inverse of the articulated inertia matrix of the vehicle. For the current work, it is intended to control the motion of the vehicle along the surge, sway and, heave axis and around the yaw axis. This translates into using the subscripts of $i,\; j = 3\ldots 6$ in Eq. (20).

Sliding surfaces s_i are defined for each controlled degree of freedom according to

$$s_i = \left(\frac{d}{dt} + \boldsymbol{\lambda}_i\right)^2 \tilde{x}_i = \dot{x}_i - \dot{x}_{i,d} + \boldsymbol{\lambda}\left(x_i - x_{i,d}\right) \;\; i = 3\ldots 6 \quad (21)$$

where subscript d represents the desired value of the state variable. The term x_i is the controlled degree of freedom with respect to the inertial frame. The term λ is the bandwidth of the closed-loop system. For notational compactness, Eq. (21) can be written as:

$$s_i = \dot{x}_i - x_{ri} \quad (22)$$

The error metric s can be kept zero by constructing a control input u that satisfies the sliding condition

$$\frac{1}{2}\frac{d}{dt}s_i^2 \le -\eta_i |s_i| \;\; \eta_i > 0 \quad (23)$$

As long as the control input satisfies Eq. (23), it drives the state variables onto the switching line and then keeps them on the line in spite of parameter changes and disturbances. Once the system state is trapped on this line, it remains on the switching line provided that the sliding condition is satisfied. As a result, the system behaves consistently with the desired dynamics defined by Eq. (22) when sliding occurs.

The control input is chosen to be of the form [4]:

$$u_i = \sum_{j=3}^{6}\left[\hat{\boldsymbol{I}}_0^*\right]_{ij}\left(x_{rj} - \hat{f}_j - k_j sat\left(s_j/\phi\right)\right) \; j = 3\ldots 6 \quad (24)$$

where the sign '^' denotes the estimate of relevant terms. The *sat* function is defined as:

$$\begin{aligned} sat\left(\frac{s_i}{\phi}\right) &= \frac{s_i}{\phi} \quad if \left|\frac{s_i}{\phi}\right| \le 1 \\ sat\left(\frac{s_i}{\phi}\right) &= \mathrm{sgn}\left(\frac{s_i}{\phi}\right) \quad otherwise \end{aligned} \quad (25)$$

The saturation function, $sat(*)$, is introduced to make the control activity continuous within the boundary layer and discontinuous outside the boundary layer ϕ. This relaxes the control law and reduces control activity at the expense of controller performance [3]. Once the system state is driven into the boundary layers, it remains in that region provided that the uncertainty bounds are not exceeded.

In Eq. (25), the discontinuous term is responsible for compensating the uncertainties in the dynamic model, and it keeps the system state on the line by pushing the system state back on the line in the event of $s > \phi$.

The convergence of s to zero is guaranteed by choosing k_i such that

$$(1 - D_{ii})k_i + \sum_{j\ne i} D_{ij}k_j = F_i + \sum_{j=3}^{6} D_{ij}\left|x_{ri} - \hat{f}_j\right| + \eta_i \;\; i, j = 3\ldots 6 \quad (26)$$

Eq. (26) represents a set of 4 equations with unknown k_i $i = 3\ldots 6$. Equation (26) corresponds to the following linear equation system:

$$\begin{bmatrix} 1-D_{33} & D_{34} & D_{35} & D_{36} \\ D_{43} & 1-D_{44} & D_{45} & D_{46} \\ D_{53} & D_{54} & 1-D_{55} & D_{56} \\ D_{63} & D_{64} & D_{65} & 1-D_{66} \end{bmatrix}\begin{bmatrix} k_3 \\ k_4 \\ k_5 \\ k_6 \end{bmatrix} = \begin{bmatrix} F_3 + \sum_{j=3}^{6} D_{3j}\left|x_{r3} - \hat{f}_j\right| + \eta_3 \\ F_4 + \sum_{j=3}^{6} D_{4j}\left|x_{r4} - \hat{f}_j\right| + \eta_4 \\ F_5 + \sum_{j=3}^{6} D_{5j}\left|x_{r5} - \hat{f}_j\right| + \eta_5 \\ F_6 + \sum_{j=3}^{6} D_{6j}\left|x_{r6} - \hat{f}_j\right| + \eta_6 \end{bmatrix} \quad (27)$$

where $F_i \ge \left|f_i - \hat{f}_i\right|$ is the upper bound on the estimation error on f_i Note that (27) has a unique solution satisfying $k_i > 0$ [4]. The control input is applied to the vehicle through the term $\mathbf{f}_0$ in Eq. (4).

It is assumed that the parametric uncertainty on the inverse of the articulated inertia matrix can be expressed in multiplicative form as:

$$\left(\boldsymbol{I}_0^*\right)^{-1} = \left(1_{6\times 6} + \Delta\right)\left(\hat{\boldsymbol{I}}_0^*\right)^{-1}, \quad \left|\Delta_{ij}\right| \le D_{ij} \quad i = 1\ldots 6 \; j = 1\ldots 6 \; (28)$$

where $\hat{\boldsymbol{I}}_0^*$ is the estimate of the articulated inertia matrix, and D_{ij} is the upper bound of the parametric uncertainty on $\Delta \le 1$.

Note that the sliding-mode controller synthesis presented differs from the one used in [11] in that the control gain is defined as a function of the articulated inertias. In [11], the definition of control gains is made through trial and error.

V. RESULTS

The tethered ROVM system considered in this work a Saab-Seaeye FALCON™ ROV with a modified Hydrolek™ HLK 43000 manipulator whose frame attachment is shown in Figure 4. All the associated parameters used for the FALCON™ in this simulation study are taken from [20].

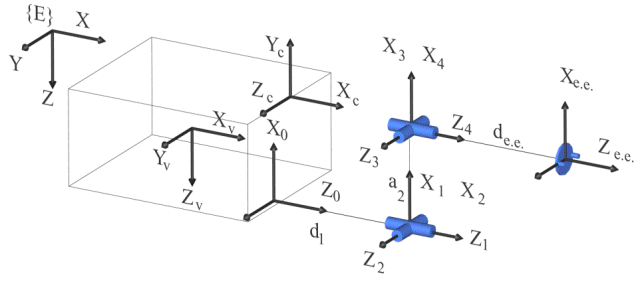


Figure 4: Frame Attachment of ROVM System

Case Study 1

In the simulation, the ROVM system is assumed to be neutrally buoyant, and is connected to a stationary surface vessel via a flexible tether whose length is 1000 meters. The system is suspended 1000 meters straight down from the surface vessel through this tether as in shown in Figure 5. The cable dynamics are kinematically constrained at the interface between the cable and the vehicle meaning the position of the last node of the cable is calculated from the vehicle state. In the case study, a station keeping task is considered. In this task assignment, the goal is to keep the FALCON™ vehicle at $\mathbf{x} = [0\ 0\ 0\ 0\ 0\ 1000]^T\ m$ under disturbances. No joint torque was applied to the joints, as the primary concern of the current work is to explore the tether disturbances on the ROV attitude. However, free manipulator motion still causes disturbance forces and moments to some extent. No joint displacement limit is considered in the current simulation study.

It is assumed that the ocean current at the ROV depth is 0.5 knots, and flows along the X axis of the inertial frame for 100 seconds. The resulting tether profiles at the instants of 0, 50,

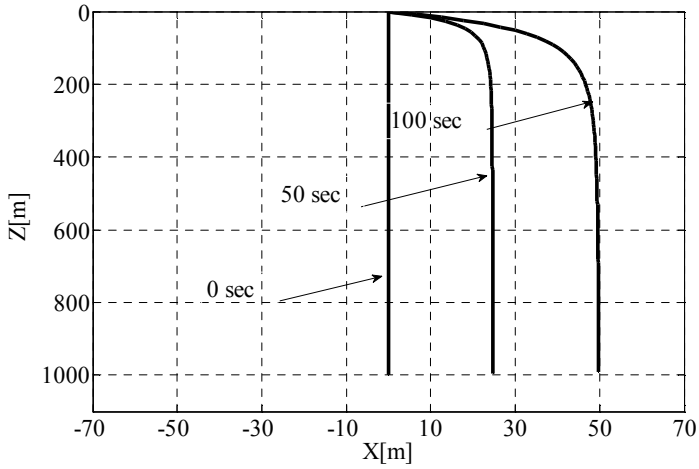


Figure 5: Tether Profile on X-Z plane of the Inertial Frame

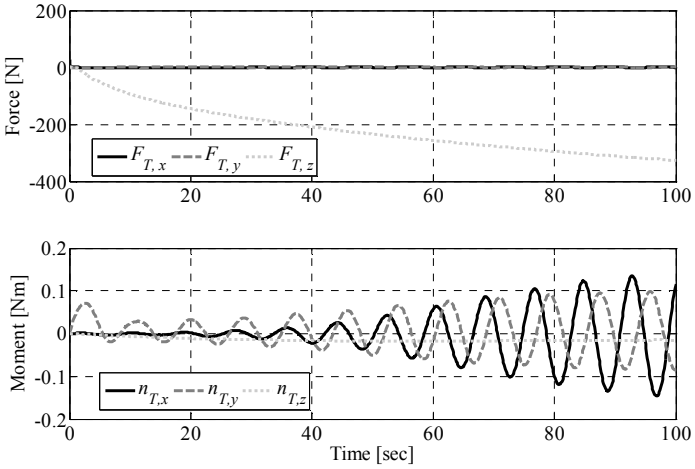


Figure 6: Tether Reaction Forces

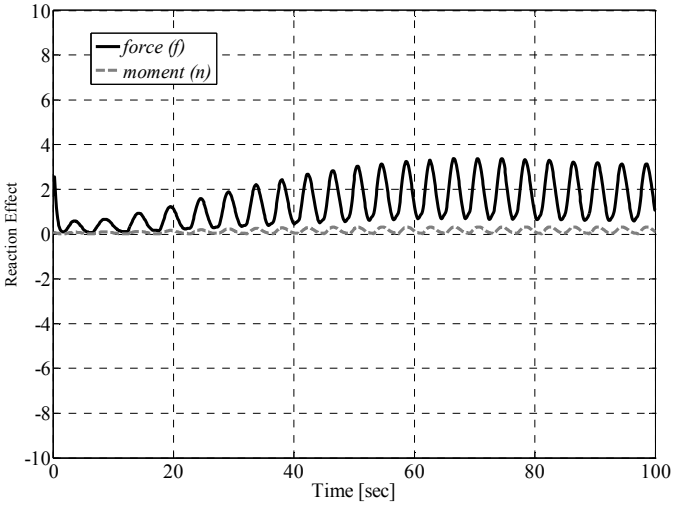


Figure 7: Euclidean norm of Manipulator Reaction Forces

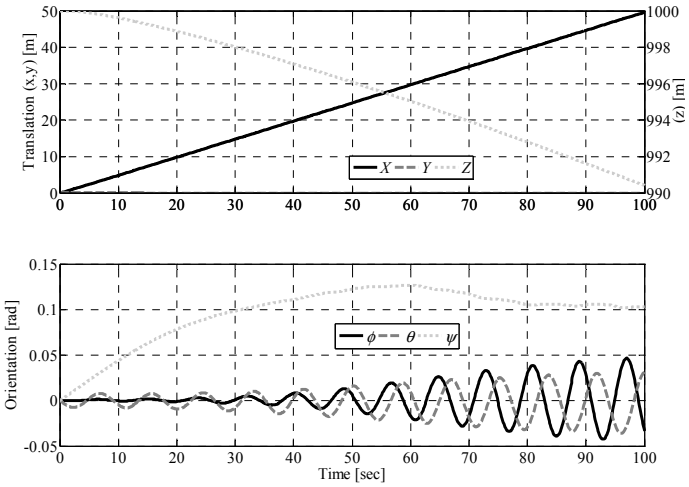


Figure 8: Time History of Vehicle Position and Orientation

100 seconds are shown in Figure 5. The constant ocean current is the triggering factor of disturbance forces and moments on the ROVM system. The tether and arm perturbs FALCON™ at the junction point through reaction forces and moments as shown in Figure 6 and Figure 7. These reaction forces and moments cause changes in the position and orientation of FALCON™ as demonstrated in Figure 8. As the figure reveals, the vehicle moves by about 50 m and 10 m along X and Z axis of the inertial reference frame, respectively. While the surge motion is due to the prevailing ocean current, the heave motion is primarily caused by the tether disturbance force along the Z axis as shown in Figure 6. The results illustrate not only the extent of the tether disturbance disturbances on the attitude of the vehicle but also the need for a controller that mitigates the disturbances due to the drag forces on the vehicle.

Case Study 2

In the second case study, the simulation conditions were kept the same as the previous case. However, this time a series of sliding-mode controllers were used to mitigate the negative effect of the tether-manipulator disturbances. Only the FALCON™s' motions along surge, heave, sway axis, and around yaw axis were attempted to control. The controller design parameters were set as $\lambda_i = 4, \phi_i = 0.3, \eta_i = 5,$ and $D_i = 0.2, i = 3 \ldots 6.$

In order to model uncertainties in the controllers, drag and added mass coefficients of FALCON™ and the Hydrolek™ differed by 40% from the real system. The changes in the dynamic parameters model imprecision in the system dynamic model. Success of the controller in the presence of these large parameter perturbations partly addresses the issue of the controller robustness.

As the water current acts over the cable along the X axis of the inertial frame, the tether buckles as shown in Figure 9. The extent to which the tether buckles is greater compared to the first case since the controller works to keep the vehicle stationary and makes the vehicle act like a fixed point at the bottom end.

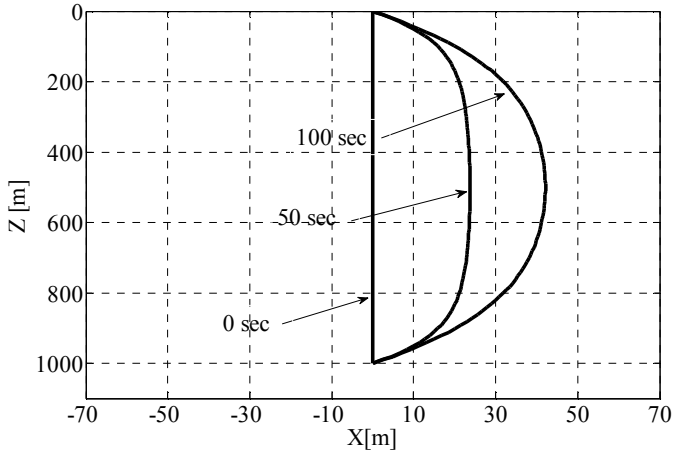


Figure 9: Tether Profile on X-Z plane of the Inertial Frame with Controller on.

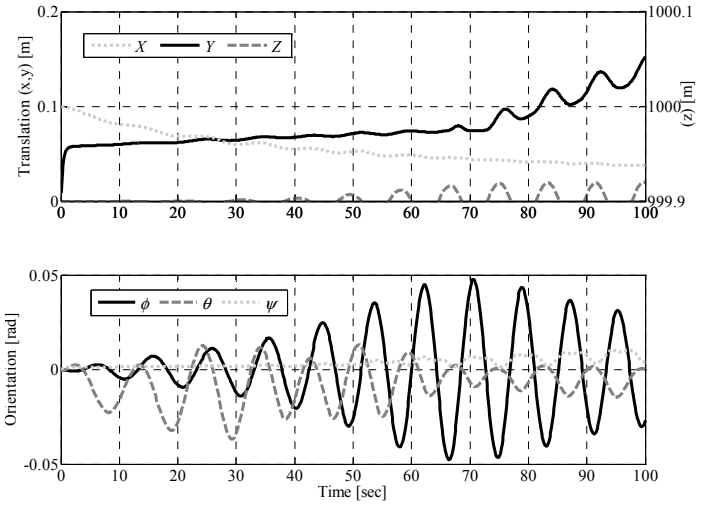


Figure 10:Time History of Vehicle Position and Orientation with Controller on.

Figure 10 is the time history of vehicle position and orientation. Given the desired values, i.e., $X = 0, Y = 0, Z = 1000,$ and $\psi = 0$, the scale of error was greatly reduced by the sliding-mode controllers in comparison to the control-off case, i.e., Figure 8. Along the X axis, the largest error is approximately 0.15m. In regards to the heave motion, the error is approximately 0.08m.

Case Study 3

Differently from the second case study, in the third case, it is assumed that the controllers have the knowledge of the tether disturbances on the FALCON™. In real applications, this knowledge can be obtained by force sensors planted at the tether- junction point. The measured values can then be passed into the controller as a feedforward term. Since the inclusion of the tether disturbance knowledge increases the accuracy of the dynamics model of the system in the controller, the controllers exhibit better performance as demonstrated in Figure 11. Along the X axis, the largest error is approximately 0.063m. In regards to the heave motion, the error is in scale of 10^{-3}.

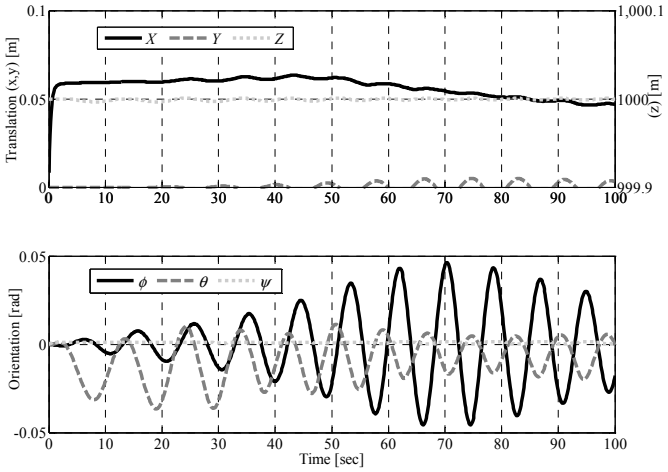


Figure 11: Time History of Vehicle Position and Orientation with tether disturbances in.

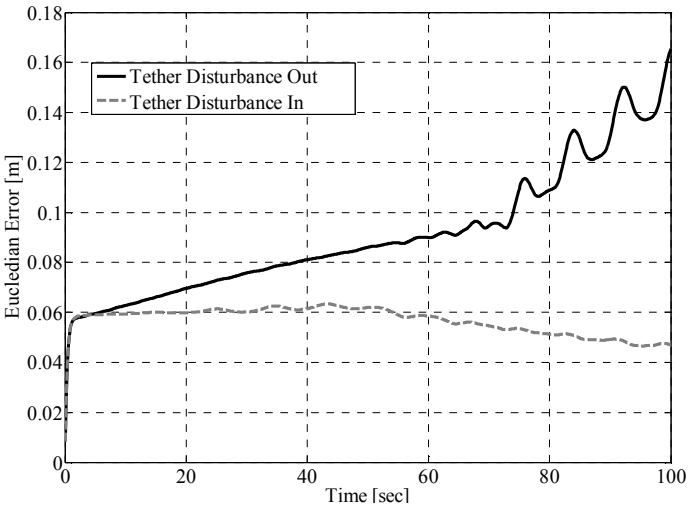


Figure 12: Euclidean norm of the position error with tether force in and out.

Figure 12 is Euclidean norm of position error figure reveals that the inclusion of the tether disturbance knowledge in to the controller reduces the max error from 0.1655m down to 0.047 m. Therefore, the controller performance can be increased significantly by introducing the tether disturbance knowledge to the sliding mode controllers.

VI. CONCLUSION

The dynamics modelling strategy of a tethered underwater remotely operated vehicle coupled with a spatial manipulator have been presented. Regarding the manipulator dynamics, the ABA is employed due to its computational efficiency. It has also been shown that disturbance forces and moments created by tether and manipulator motion must be compensated for in particular for a station keeping task. In order to control the ROV motion under these disturbance effects, a series of sliding-mode controllers have been designed. The control gains have been defined as a function of the articulated inertias as opposed to the conventional trial-and-error approach. It has shown that the controller performance can be increased to a relatively significant degree by introducing the tether disturbance knowledge to the sliding mode controllers.

ACKNOWLEDGMENT

The authors would like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for providing financial support for this work.

REFERENCES

[1] T. W. McLain and S. M. Rock, "Experimental Measurement of ROV Tether Tension", in Proceedings of 10th Ann. Subsea Interv. Conf. and Expo., San Diego, 1992.

[2] B.J. Buckham, "Dynamics modelling of low-tension tethers for submerged remotely operated vehicles," Ph.D. Thesis, University of Victoria, 2003.

[3] D. Yoerger, and J. J. E. Slotine, "Robust trajectory control of underwater vehicles," IEEE J. Ocean Eng., vol. OE-10, pp. 462-470, Oct. 1985.

[4] J. J. E. Slotine, and L. Weiping, Applied Nonlinear Control, Prentice-Hall, 1991.

[5] A. J. Healey and D. Lienard, "Multivariable sliding-mode control for autonomous diving and steering of unmanned vehicles," IEEE Journal of Oceanic Engineering, vol. 18, no. 3, pp. 327-339, 1993.

[6] R. Cristi, F. A. Papoulias, A. J. Healey, "Adaptive sliding mode control of autonomous underwater vehicles in the dive plane," IEEE Journal of Oceanic Engineering, vol. 15, no. 3, pp. 152-160, July 1990.

[7] S. Soylu, B. J. Buckham and R. P. Podhorodeski , "A chattering free sliding-mode controller for underwater vehicles with fault-tolerant infinity-norm thrust allocation," Ocean Engineering, vol.35, no. 16, pp. 1647-1659, 2008.

[8] K. Goheen and E. R. Jeffery, "Mutivariable self-tuning autopilots for autonomous and remotely operated undrewatervehicle," IEEE Journal of Oceanic Engineering, vol. 15, no. 3, pp. 144-151, 1998.

[9] J. Yuh, "A neural net controller for underwater robotic vehicles," IEEE J. Oceanic Eng., vol. 15, no. 3, pp. 327-339, 1993.

[10] M. W. Dunnigan, and G. T. Russell, "*Evaluation and reduction of the dynamic coupling between a manipulator and an underwater vehicle*," IEEE Journal of Oceanic Engineering, vol. 23, no. 3, pp. 260-273, 1998.

[11] S. Soylu, B. J. Buckham and R. P. Podhorodeski,"Using articulated-body algorithm within sliding mode control to compensate dynamic coupling in underwater manipulator systems," CSME Transactions 2005 Special Edition, June 2006.

[12] I. Kaminer and A. M. Pascoal, "Control of an underwater vehicle using H_∞ synthesis", Proc. 30th IEEE CDC, pp.2350-2355,1991.

[13] G. Conte and A. Serrani, "Robust control of a remotely operated underwater vehicle," Automatica, vol.34, no. 2, pp. 193-198, 1998.

[14] S. Soylu, B. J. Buckham and R. P. Podhorodeski, MIMO Sliding-Mode and H∞ Controller Design For Dynamic Coupling Reduction In Underwater- Manipulator Systems," CSME Transactions vol. 33, no. 4, pp. 731-743, 2009.

[15] S. McMillan, "Computational dynamics for robotic systems on land and underwater," Ph.D Dissertation, The Ohio State University, Colombus, Summer 1994.

[16] S. McMillan, D. E.Orin, and R. B. McGhee, "Efficient dynamic simulation of an underwater vehicle with a robotic manipulator," IEEE Trans. Syst., Man, Cybern., vol. 25 pp. 1194-1206, Aug. 1995

[17] T. Fossen, Guidance and Control of Ocean Vehicles, John Wiley & Sons, 1994.

[18] D. Steinke, "Design and Simulation of a Kalman filter for ROV Navigation," Master's. Thesis, University of Victoria, 2006.

[19] J. J. Craig, Introduction to Robotics, 2nd Edition. Addison-Wesley, 1989.

[20] J. Zand, "Enhanced Navigation and Tether Management of Inspection Class ROVs.," Master's. Thesis, University of Victoria, 2010.

Appendix C. A Chattering-Free Sliding-Mode Controller for Underwater Vehicles with Fault-Tolerant Infinity-Norm Thrust Allocation

(2008, Ocean Engineering, volume 35, issue 16, pages 1647-1659)

Ocean Engineering 35 (2008) 1647–1659

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

A chattering-free sliding-mode controller for underwater vehicles with fault-tolerant infinity-norm thrust allocation

Serdar Soylu *, Bradley J. Buckham, Ron P. Podhorodeski

Department of Mechanical Engineering, University of Victoria, P.O. Box 3055, Victoria, BC, Canada V8W 3P6

ARTICLE INFO

Article history:
Received 30 November 2007
Accepted 23 July 2008
Available online 26 July 2008

Keywords:
Nonlinear control
Underwater vehicles
Thruster force allocation
Fault-tolerant systems

ABSTRACT

There are two objectives to this paper. First, a chattering-free sliding-mode controller is proposed for the trajectory control of remotely operated vehicles (ROVs). Second, a new approach for thrust allocation is proposed that is based on minimizing the largest individual component of the thrust manifold. With regards to the former, a new adaptive term is developed that eliminates the high-frequency control action inherent in a conventional sliding-mode controller. As opposed to the common adaptive approach, the new adaptive term does not require the linearity condition on the dynamic parameters and the creation of a regressor matrix. In addition, it removes the need for a *priori* knowledge of upper bounds on uncertainties in the dynamic parameters of the ROV. With regards to the latter, it is demonstrated that minimizing the l_∞ norm (infinity-norm) of the thrust manifold ensures low individual thruster forces. The new control and thrust allocation concepts are implemented in numerical simulations of a work class ROV, and the chattering-free nature of the controller is demonstrated during typical ROV manoeuvres. In the simulation studies, the l_∞ norm-based thrust allocation problem is cast as a linear programming problem that allows direct incorporation of the thruster saturation limits and a fault-tolerant property. To achieve real-time solution rates for the l_∞ norm-based thrust allocation problem, a recurrent neural network is designed. In the simulation studies, the l_∞ norm-based thrust allocation provides smaller maximum absolute value of the largest component of the thrust manifold than that of a conventional l_2 norm (2-norm) minimization, satisfies the saturation limits of each thruster, and accommodates faults that are introduced arbitrarily during the manoeuvre.

1. Introduction

Remotely operated vehicles (ROVs) play an important role in a number of shallow and deep-water missions for marine science, oil and gas extraction, exploration, and salvage. In these applications, the motions of the ROV are guided either by a human pilot on a surface support vessel through an umbilical cord providing power and telemetry, or by an automatic pilot. In the case of automatic control, ROV state feedback is provided by acoustic and inertial sensors, and this state information along with a controller strategy is used to drive several conventional thrusters arranged on the ROV chassis.

In the existing literature, several different automatic control approaches have been applied to control ROV motion such as the H_∞ approach (Conte and Serrani, 1998), adaptive control techniques (Antonelli et al., 2001, 2003), sliding-mode control (Yoerger and Slotine, 1985; Slotine and Coetsee, 1986; Healey and Lienard, 1993; Antonelli, 2003), fuzzy-logic control (Debitetto, 1994; Kato, 1995), and neural network methods (Ishii and Ura, 2000; Kodogiannis, 2003; Pepijn et al., 2005; Van de Ven et al., 2005). It has been shown that the model-based sliding-mode approach is an effective means of controlling an ROV, largely due to its ability to tolerate imprecision in the dynamics model (Yoerger and Slotine, 1985; Slotine and Coetsee, 1986). However, one major drawback of the sliding-mode approach is the high frequency of control action (chattering). This high-frequency control activity causes high heat losses in electrical power circuits and premature wear in actuators. In addition, the high control activity may excite unmodelled high-frequency dynamics, which in turn causes controller performance degradation. To eliminate or reduce chattering, various methods such as the boundary layer method (Yoerger and Slotine, 1985; Slotine and Shastry, 1983) and the disturbance compensation method (Elmali and Olgac, 1992; Zeinali and Notash, 2007) have been presented. The boundary layer approach makes the control activity continuous within the boundary layer and discontinuous outside the boundary layer. In this work, a disturbance compensation method is utilized.

* Corresponding author. Tel.: +1 250 472 4065; fax: +1 250 721 6035.
E-mail addresses: serdar@me.uvic.ca (S. Soylu), bbuckham@me.uvic.ca (B.J. Buckham), podhoro@me.uvic.ca (R.P. Podhorodeski).

0029-8018/$ - see front matter © 2008 Published by Elsevier Ltd.
doi:10.1016/j.oceaneng.2008.07.013

To eliminate chattering, the disturbance compensation approach replaces the discontinuous term of a conventional sliding-mode controller with an estimate of the uncertainties in an adaptive manner. In the current work, a disturbance compensation approach discussed by Zeinali and Notash (2007) for land-base manipulators is extended to mobile-base ROV systems.

To ensure manoeuvrability, the ROV thruster arrangement is redundant: there are more thrusters than there are active vehicle degrees of freedom. Due to the excess thrusters, there are an infinite number of ways to allocate the pilot's commanded generalized force over the existing thrusters. As such, an optimal thrust allocation must be selected by applying criteria to distinguish the various options. In this work, a new definition for optimal thrust distribution is proposed.

A prominent approach to the thrust allocation problem is the 2-norm (l_2 norm)-based solution in which the sum of the squares of the individual thruster forces is minimized. In Antonelli (2003), a pseudo-inverse solution, and in Fossen (1994), Sordalen (1997), and Omerdic and Roberts (2004), a weighted pseudo-inverse are used to generate an optimal distribution of a commanded generalized force. The pseudo-inverse method has the advantage of being relatively simple to compute. Pseudo-inverse solutions correspond to the minimization of either the l_2 norm or a weighted l_2 norm of the thrust manifold. However, the l_2 norm-based solution does not necessarily minimize the magnitudes of the individual thrusts, and can generate thrust demands that may exceed an individual thruster's saturation point. In addition, there could be an unequal distribution in the thrust manifold leading to a relatively high thrust demand for a particular thruster. In such cases, there exists a potential for a loss of manoeuvrability on subsequent control steps. The disadvantages of the l_2 norm-based optimization were reported for land-base manipulators by Arati and Walker (1997).

Furthermore, the pseudo-inverse method (l_2 norm minimization) does not afford easy implementation of thruster saturation limits (Omerdic and Roberts, 2004). It was reported by Durham (1993) that, even if thruster saturation is implemented, the pseudo-inverse solution is not guaranteed to satisfy the saturation constraints. The pseudo-inverse solution was also used for the thruster allocation problem by Sarkar et al. (2002). To generate reference thruster values that do not exceed the saturation limit of each thruster, Sarkar et al. (2002) employed a dynamic state feedback method.

In the current work, it is proposed that the complications associated with the l_2 norm-based solutions be avoided by using the infinity-norm (l_∞ norm) as the criterion in the thrust allocation. (The l_∞ norm is defined as the absolute value of the largest component of the thrust manifold). By using the l_∞ norm to gauge optimality of a thrust distribution, the largest single thrust in the distribution is minimized. The current work shows how the l_∞ norm thruster allocation can be cast as a constrained linear programming problem that allows direct implementation of thruster saturation limits and a fault-tolerant property. As pointed out by Sarkar et al. (2002), the allocation of thruster force problem in the presence of thruster faults and saturation limits for ROV systems has not been studied extensively. To obtain real-time computation rates for the linear programming problem, a recurrent neural network is proposed. An l_∞ norm-based thruster allocation has been preliminarily discussed by the authors of the current work in Soylu et al. (2007).

The efficacy of the proposed fault-accommodating thrust allocation scheme with the chattering-free sliding-mode control is demonstrated through numerical simulation studies on the ROPOS vehicle operated by the Canadian Scientific Submersible.

2. Dynamics and control

2.1. Dynamics of an ROV

The dynamic equations of motion of ROVs in the body-fixed frame can be represented as (Fossen, 1994)

$$\begin{aligned}&\mathbf{M}\dot{\boldsymbol{q}}+\mathbf{C}(\dot{\boldsymbol{q}})\boldsymbol{q}+\mathbf{D}(\dot{\boldsymbol{q}})\boldsymbol{q}+\mathbf{g}(\boldsymbol{\eta})=\boldsymbol{\tau}\\&\dot{\boldsymbol{\eta}}=\mathbf{J}(\boldsymbol{\eta})\boldsymbol{q}\end{aligned}\tag{1}$$

where $\boldsymbol{q}=[u\ \ v\ \ w\ \ p\ \ q\ \ r]^{\mathrm{T}}$ is the ROV spatial velocity state vector with respect to its body-fixed frame, and $\boldsymbol{\eta}=[x\ \ y\ \ z\ \ \phi\ \ \theta\ \ \psi]^{\mathrm{T}}$ is the position and orientation state vector with respect to the inertial frame. The coordinate systems considered are illustrated in Fig. 1.

In Eq. (1), the spatial transformation matrix between the inertial frame and the ROV's body-fixed frame can be defined through the Euler angle transformation (Fossen, 1994), denoted by $\mathbf{J}(\boldsymbol{\eta})\in\mathbb{R}^{6\times6}$. The term $\mathbf{M}(\boldsymbol{q})\in\mathbb{R}^{6\times6}$ is the inertia matrix including the added mass effects, $\mathbf{C}(\boldsymbol{q})\in\mathbb{R}^{6\times6}$ is the matrix of centrifugal and Coriolis terms, $\mathbf{D}(\boldsymbol{q})\in\mathbb{R}^{6\times6}$ is the drag matrix, $\mathbf{g}(\boldsymbol{\eta})\in\mathbb{R}^{6}$ is the vector of gravity and buoyancy forces and moments, and finally $\boldsymbol{\tau}\in\mathbb{R}^{6}$ is the control forces and moments acting on the ROV centre of mass.

Eq. (1) can be represented in the inertial reference frame as (Fossen, 1994)

$$\boldsymbol{f}=\mathbf{M}_\eta(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}}+\mathbf{C}_\eta(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}}+\mathbf{D}_\eta(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}}+\boldsymbol{g}_\eta(\boldsymbol{\eta})=\mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau}\tag{2}$$

where $\mathbf{M}_\eta(\boldsymbol{\eta})=\mathbf{J}^{-\mathrm{T}}\mathbf{M}\mathbf{J}^{-1}$, $\mathbf{C}_\eta(\boldsymbol{q},\boldsymbol{\eta})=\mathbf{J}^{-\mathrm{T}}[\mathbf{C}-\mathbf{M}\mathbf{J}^{-1}\dot{\mathbf{J}}]\mathbf{J}^{-1}$, $\mathbf{D}_\eta(\boldsymbol{q},\boldsymbol{\eta})=\mathbf{J}^{-\mathrm{T}}\mathbf{D}\mathbf{J}^{-1}$ and $\boldsymbol{g}(\boldsymbol{\eta})=\mathbf{J}^{-\mathrm{T}}\boldsymbol{g}$. The dynamics of an ROV are assumed to have the following structural properties (Fossen, 1994):

Property 1. The inertia matrix $\mathbf{M}_\eta$ is symmetric and positive definite, i.e., $\mathbf{M}_\eta^{\mathrm{T}}=\mathbf{M}_\eta$;

Property 2. Matrix $\dot{\mathbf{M}}_\eta-2\mathbf{C}_\eta$ is skew symmetric, i.e., for any vector ζ, $\zeta^{\mathrm{T}}(\dot{\mathbf{M}}_\eta-2\mathbf{C}_\eta)\zeta=0$.

Eq. (2) can be written in a more compact form of

$$\boldsymbol{f}=\mathbf{M}_\eta(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}}+\boldsymbol{h}_\eta(\boldsymbol{q},\boldsymbol{\eta})\tag{3}$$

where $\boldsymbol{h}(\boldsymbol{q},\boldsymbol{\eta})=\mathbf{C}_\eta(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}}+\mathbf{D}_\eta(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}}+\boldsymbol{g}_\eta(\boldsymbol{\eta})$. As mentioned earlier, the ROV dynamics are dominated by hydrodynamic loads, and it is difficult to accurately measure or estimate the hydrodynamic coefficients that are valid for all ROV operating conditions and instrument and tether configurations. As such, the system dynamics are not exactly known. Therefore, the system dynamics given in Eq. (3) can be written as the sum of estimated dynamics $\hat{\boldsymbol{f}}$ and the unknown dynamics $\tilde{\boldsymbol{f}}$:

$$\boldsymbol{f}=\hat{\boldsymbol{f}}+\tilde{\boldsymbol{f}}\tag{4}$$

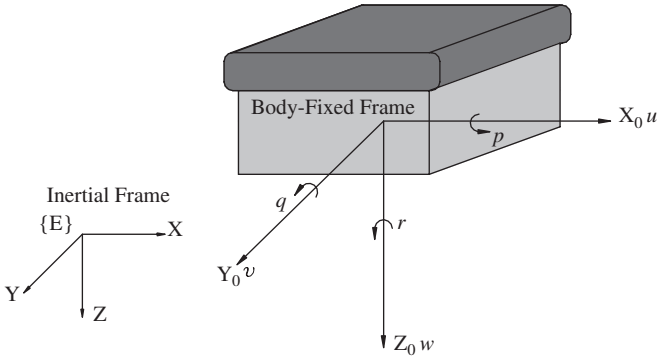


Fig. 1. Coordinate systems.

where the estimated dynamics vector is defined as

$$\hat{\boldsymbol{f}} = \hat{\mathbf{M}}_\eta(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \hat{\boldsymbol{h}}(\boldsymbol{q},\boldsymbol{\eta}) \tag{5}$$

with $\hat{\boldsymbol{h}}_\eta(\boldsymbol{q},\boldsymbol{\eta}) = \hat{\mathbf{C}}_\eta(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \hat{\mathbf{D}}_\eta(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \hat{\boldsymbol{g}}_\eta(\boldsymbol{\eta})$ and the unknown dynamics vector are defined as

$$\tilde{\boldsymbol{f}} = \tilde{\mathbf{M}}_\eta(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \tilde{\boldsymbol{h}}_\eta(\boldsymbol{q},\boldsymbol{\eta}) + \boldsymbol{d} \tag{6}$$

with $\tilde{\boldsymbol{h}}_\eta(\dot{\boldsymbol{q}},\boldsymbol{\eta}) = \tilde{\mathbf{C}}_\eta(\dot{\boldsymbol{q}},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \tilde{\mathbf{D}}_\eta(\dot{\boldsymbol{q}},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \tilde{\boldsymbol{g}}_\eta(\boldsymbol{\eta})$, $\tilde{\mathbf{M}}_\eta = \mathbf{M}_\eta - \hat{\mathbf{M}}_\eta$, $\tilde{\mathbf{C}}_\eta = \mathbf{C}_\eta - \hat{\mathbf{C}}_\eta$, $\tilde{\mathbf{D}}_\eta = \mathbf{D}_\eta - \hat{\mathbf{D}}_\eta$, and $\tilde{\boldsymbol{g}}_\eta = \boldsymbol{g}_\eta - \hat{\boldsymbol{g}}_\eta$. Note that $\boldsymbol{d} \in \mathbb{R}^6$ is added as a disturbance force vector. The unknown dynamics vector is also called lumped uncertainty vector (Lin and Wai, 2002).

Assumption 1. Nonlinear lumped uncertainty vector $\tilde{\boldsymbol{f}}$ given in (6) and its time derivative are bounded.

2.2. Chattering-free adaptive sliding-mode controller design

2.2.1. Sliding-mode controller with a switching term

A controller design based on the sliding-mode methodology involves two main steps. The first step is to define the desired dynamics in the form of a vector of sliding manifolds $\boldsymbol{s} \in \mathbb{R}^6$. The second step is to find a control law $\boldsymbol{\tau} \in \mathbb{R}^6$ such that the system trajectories move toward the sliding manifold, and once they hit the manifold, remain on it in the presence of system uncertainties and disturbances (Slotine and Li, 1991).

For a second-order system such as an underwater vehicle system, reasonable desired dynamics would be a stable first-order system. This first-order dynamics can be defined as (Slotine and Li, 1991)

$$\boldsymbol{s} = \left(\frac{\mathrm{d}}{\mathrm{d}t} + \boldsymbol{\Lambda}\right)^2 \left(\int \boldsymbol{e}\,\mathrm{d}t\right) = \dot{\boldsymbol{e}} + 2\boldsymbol{\Lambda}\boldsymbol{e} + \boldsymbol{\Lambda}^2 \int \boldsymbol{e}\,\mathrm{d}t \tag{7}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{6\times 6}$ is a constant, symmetric, positive definite and diagonal matrix that defines the break frequency of the desired error response. Each component of the sliding manifold represents a time-varying line in the state space that passes through the desired state variables. When $\boldsymbol{s} = 0$, the system states are on the surface meaning the system behaves consistently with the desired dynamics. This implies that the value of $\boldsymbol{s}$ indicates the extent of discrepancy between the desired state and the current state. Any sliding-mode-based controller works to keep the value of $\boldsymbol{s}$ at zero.

The tracking error between the measured state values and the desired state values is given by $\boldsymbol{e} = \boldsymbol{\eta} - \boldsymbol{\eta}_\mathrm{d}$ with the subscript d denoting the desired position and attitude of the ROV produced by a separate trajectory planner. The integral term ensures zero offset error. For notational simplicity, Eq. (7) can also be written as

$$\boldsymbol{s} = \dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_\mathrm{r} \tag{8}$$

where $\dot{\boldsymbol{\eta}}_\mathrm{r} = \dot{\boldsymbol{\eta}}_\mathrm{d} - 2\boldsymbol{\Lambda}\boldsymbol{e} - \boldsymbol{\Lambda}^2 \int \boldsymbol{e}\,\mathrm{d}t$, and the subscript r refers to virtual reference trajectory (Slotine and Li, 1991).

The standard sliding-mode control law is in the form

$$\boldsymbol{\tau} = \boldsymbol{\tau}_\mathrm{eq} + \boldsymbol{\tau}_\mathrm{sw} \tag{9}$$

where $\boldsymbol{\tau}$ corresponds to a generalized force acting at the centre of mass of the ROV, and $\boldsymbol{\tau}_\mathrm{eq}$ and $\boldsymbol{\tau}_\mathrm{sw}$ symbolize the equivalent control law and the switching control law, respectively.

The equivalent control law is continuous and model based. In the absence of uncertainties in the system dynamics, this equivalent control alone suffices to realize the desired dynamics. However, due to model uncertainties, an auxiliary switching term is needed that offsets the difference between the desired dynamics and real dynamics. This switching term is defined as $\boldsymbol{\tau}_\mathrm{sw} = -\mathbf{K}\,\mathrm{sgn}(\boldsymbol{s})$ in conventional sliding-mode control, where $\mathbf{K} \in \mathbb{R}^{6\times 6}$ is the positive definite diagonal gain matrix that is defined based on the upper bounds on the system parameter uncertainties, and $\mathrm{sgn}(\cdot)$ is the nonlinear signum function.

The switching term is a discontinuous feedback component that is in charge of compensating for deviations from the desired dynamics, and therefore is the source of the robustness of the sliding-mode control law. The switching term acts on the system in a bang–bang manner creating chatter in the actuators and causing the system state to oscillate intensely across the sliding manifold (Slotine and Li, 1991). The goal of this section is to replace the discontinuous switching term $\boldsymbol{\tau}_\mathrm{sw}$ with a continuous adaptive term $\boldsymbol{\tau}_\mathrm{ad}$ in an effort to eliminate the chattering problem.

2.2.2. Equivalent control law

The model-based equivalent control law component of the sliding-mode control signal can be derived by assuming that the motion is constrained to the sliding manifold. This implies that $\boldsymbol{s}$ is a constant vector, and thus $\dot{\boldsymbol{s}} = \mathbf{0}$. The time derivative of $\boldsymbol{s}$ can be defined based on Eq. (8), as

$$\dot{\boldsymbol{s}} = \ddot{\boldsymbol{\eta}} - \ddot{\boldsymbol{\eta}}_\mathrm{r} \tag{10}$$

where $\ddot{\boldsymbol{\eta}}_\mathrm{r} = \ddot{\boldsymbol{\eta}}_\mathrm{d} - 2\boldsymbol{\Lambda}\dot{\boldsymbol{e}} - \boldsymbol{\Lambda}^2\boldsymbol{e}$. Multiplying both sides of Eq. (10) by the inertia matrix $\hat{\mathbf{M}}_\eta$, and substituting $\mathbf{M}_\eta\ddot{\boldsymbol{\eta}} = \mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau}_\mathrm{eq} - (\mathbf{C}_\eta\dot{\boldsymbol{\eta}} + \mathbf{D}_\eta\dot{\boldsymbol{\eta}} + \boldsymbol{g}_\eta)$ from Eq. (2) into the resulting equation yields

$$\hat{\mathbf{M}}_\eta\dot{\boldsymbol{s}} = \mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau}_\mathrm{eq} - (\hat{\mathbf{M}}_\eta\ddot{\boldsymbol{\eta}}_\mathrm{r} + \hat{\mathbf{C}}_\eta\dot{\boldsymbol{\eta}} + \hat{\mathbf{D}}_\eta\dot{\boldsymbol{\eta}} + \hat{\boldsymbol{g}}_\eta). \tag{11}$$

Letting $\dot{\boldsymbol{s}} = 0$ and solving the resulting equation for $\boldsymbol{\tau}_\mathrm{eq}$ yield the equivalent control as

$$\boldsymbol{\tau}_\mathrm{eq} = \mathbf{J}^\mathrm{T}\hat{\boldsymbol{f}}_\mathrm{r}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}},\ddot{\boldsymbol{\eta}}_\mathrm{r}) \tag{12}$$

where

$$\hat{\boldsymbol{f}}_\mathrm{r}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}},\ddot{\boldsymbol{\eta}}_\mathrm{r}) = \hat{\mathbf{M}}_\eta\ddot{\boldsymbol{\eta}}_\mathrm{r} + \hat{\mathbf{C}}_\eta\dot{\boldsymbol{\eta}} + \hat{\mathbf{D}}_\eta\dot{\boldsymbol{\eta}} + \hat{\boldsymbol{g}}_\eta \tag{13}$$

As mentioned before, in the absence of uncertainties in the system dynamics, this equivalent control alone can keep the state variables on the sliding surface.

2.2.3. Adaptive control law

With regard to the replacement of the discontinuous term, the following continuous adaptive control law is proposed in place of the switching term:

$$\boldsymbol{\tau}_\mathrm{ad} = \mathbf{J}^\mathrm{T}(\tilde{\boldsymbol{f}}_\mathrm{est} - (\mathbf{K} + \hat{\mathbf{C}}_\eta)\boldsymbol{s}) \tag{14}$$

where $\tilde{\boldsymbol{f}}_\mathrm{est}$ is an adaptive term that estimates the lumped uncertainty vector defined in Eq. (6), and $\mathbf{K} \in \mathbb{R}^{6\times 6}$ is a diagonal positive definite constant matrix that is related to the convergence rate of the controller. The estimation of the lumped uncertainty vector is proposed to follow:

$$\dot{\tilde{\boldsymbol{f}}}_\mathrm{est} = -\boldsymbol{\Gamma}\boldsymbol{s} \tag{15}$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{6\times 6}$ is a positive definite diagonal constant design matrix that determines the rate of adaptation. This adaptive term relates the error metric $\boldsymbol{s}$ function to the dynamic uncertainties, and acts on the controller in such a way that the estimated dynamics reflect the unknown dynamics more closely to the actual dynamics. The assumption of $\tilde{\boldsymbol{f}}$ being bounded ensures that Eq. (15) is bounded as well.

The total control input $\boldsymbol{\tau}$ is defined as the sum of the equivalent control signal and the adaptive control signal, and is given by

$$\begin{aligned}\boldsymbol{\tau} &= \boldsymbol{\tau}_\mathrm{eq} + \boldsymbol{\tau}_\mathrm{ad} \\ &= \mathbf{J}^\mathrm{T}(\hat{\boldsymbol{f}}_\mathrm{r} + \tilde{\boldsymbol{f}}_\mathrm{est} - (\mathbf{K} + \hat{\mathbf{C}}_\eta)\boldsymbol{s})\end{aligned} \tag{16}$$

Assumption 2. The following inequality is assumed to hold:

$$\boldsymbol{s}^\mathrm{T}\mathbf{K}\boldsymbol{s} \geqslant |\dot{\tilde{\boldsymbol{f}}}^\mathrm{T}\boldsymbol{\Gamma}^{-1}\boldsymbol{w}| \quad \text{only when } \dot{\tilde{\boldsymbol{f}}}^\mathrm{T}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} < 0 \tag{17}$$

where $\boldsymbol{w} = \tilde{\boldsymbol{f}}_{\text{est}} - \tilde{\boldsymbol{f}}$ is the difference vector between the estimated lumped uncertainty vector and the exact lumped uncertainty vector.

Remark. Under slow ROV motion assumptions, $\dot{\tilde{\boldsymbol{f}}}$ is relatively small. Also, the term $\boldsymbol{w}$ becomes smaller as the adaptive term acts on the system. Therefore, Assumption 2 is rather realistic. In the worst case scenario, Assumption 2 can be strengthened by increasing the $\mathbf{K}$ and $\boldsymbol{\Gamma}$ entries.

Theorem 1. *Consider the nonlinear dynamical system described by Eq. (2) with Assumptions 1 and 2. If the control law is expressed as Eq. (16), then stability of the closed-loop control system is guaranteed.*

Proof. See Appendix A.

The general scheme of the control law is illustrated in Fig. 2. The control law is given by Eq. (16) and the update law for the adaptive term is given by Eq. (15). The novelty of Eq. (16) is that it replaces the discontinuous term with a term, $\tilde{\boldsymbol{f}}_{\text{est}}$. This term continuously acts on the model-based equivalent control term and works to reduce the discrepancy between the known and the exact dynamics at each sampling time. As the discrepancy is reduced, the state variables stay on the sliding manifold once they are driven on to it.

When the boundary layer approach is used, the robustness property of the sliding-mode control is compromised for the sake of chattering elimination since sliding-mode controller acts similar to PD controllers within the boundary layers. The given adaptive term, however, preserves this robustness property since the unknown dynamics are estimated and added to the equivalent control signal as a corrective term. In addition, as opposed to conventional adaptive control theory, no parameterization of a regressor matrix and unknown parameter vector needs to be found. The regressor matrix is so complex that further assumptions are necessary in the system model leading to further degradation in the quality of the modelled system dynamics (Antonelli et al., 2001). The proposed method does not require this linearization since it estimates the complete accumulation of unknown dynamics. In this regard, the proposed adaptive term blends the advantages of the sliding and adaptive control while eliminating their disadvantages.

3. Infinity-norm thrust allocation

The fault-tolerant, l_∞ norm-based thrust allocation scheme will be formulated in this section.

3.1. Thruster force model

The generalized force vector $\boldsymbol{\tau}_i \in \mathbb{R}^{6\times 1}$ produced by the ith thruster can be expressed as (Omerdic and Roberts, 2004)

$$\boldsymbol{\tau}_i = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{n}_i \end{bmatrix} = \begin{bmatrix} p_i \boldsymbol{e}_i \\ p_i(\boldsymbol{r}_i \times \boldsymbol{e}_i) \end{bmatrix} = \begin{bmatrix} e_{i,x} \\ e_{i,y} \\ e_{i,z} \\ (r_i \times e_i)_x \\ (r_i \times e_i)_y \\ (r_i \times e_i)_z \end{bmatrix} p_i \tag{18}$$

where p_i is the thrust exerted by the ith thruster, and $\boldsymbol{f}_i$ and $\boldsymbol{n}_i$ are the force and moment vectors associated with p_i. The vector $\boldsymbol{e}_i = [e_{i,x} \quad e_{i,y} \quad e_{i,z}]^{\mathrm{T}}$ in Eq. (18) indicates the positive direction of the thrust p_i and the position vector $\boldsymbol{r} = [r_{i,x} \quad r_{i,y} \quad r_{i,z}]^{\mathrm{T}}$ determines the position of each thruster with respect to the ROV centre of mass as illustrated in Fig. 4.

The control input $\boldsymbol{\tau}_i \in \mathbb{R}^{6\times 1}$ calculated in Eq. (16) is realized by superposing the individual generalized force vectors of each thruster:

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_Z \\ \tau_K \\ \tau_M \\ \tau_N \end{bmatrix} = \sum_{i=1}^{n} \boldsymbol{\tau}_i = \sum_{i=1}^{n} \begin{bmatrix} e_i \\ (r_i \times e_i) \end{bmatrix} p_i = \mathbf{E}\boldsymbol{p} \tag{19}$$

where n is the number of thrusters, and $\boldsymbol{p} \in \mathbb{R}^n$ is the collection of individual thrusts, hereafter called the thrust manifold. The thruster configuration matrix $\mathbf{E} \in \mathbb{R}^{6\times n}$ captures the geometry of the thruster layout, and transforms the individual thruster forces into generalized forces experienced at the ROV mass centre.

3.2. Potential advantages of l_∞ norm over l_2 norm

In this paper, it is proposed to minimize the l_∞ norm of the thrust manifold rather than the conventional method of minimizing the l_2 norm. The l_∞ norm provides the exact representation of the feasible thrust solution space whereas the l_2 norm provides an approximation of the feasible solution space. This suggests the l_∞ norm always finds a feasible solution as long as a solution exists within the thrusters' saturation limits whereas the l_2 norm does not. To demonstrate this, consider an ROV with two onboard thrusters whose limits are normalized such that $p_{n,i} = \pm 1$ N, for $i = 1, 2$. In the l_2 norm case, the feasible thrust solutions are enclosed by a circle (an n-dimensional sphere in $\mathbb{R}^n$), i.e., $||\boldsymbol{p}||_2 = (p_1^2 + p_2^2)^{1/2} \leqslant 1$, and the corresponding task space points are enclosed by an ellipse (an n-dimensional ellipsoid in $\mathbb{R}^n$), which is obtained through the

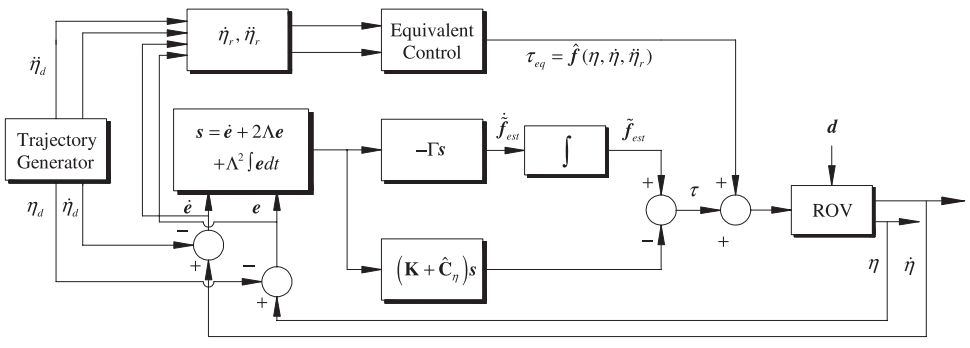


Fig. 2. Adaptive sliding-mode controller.

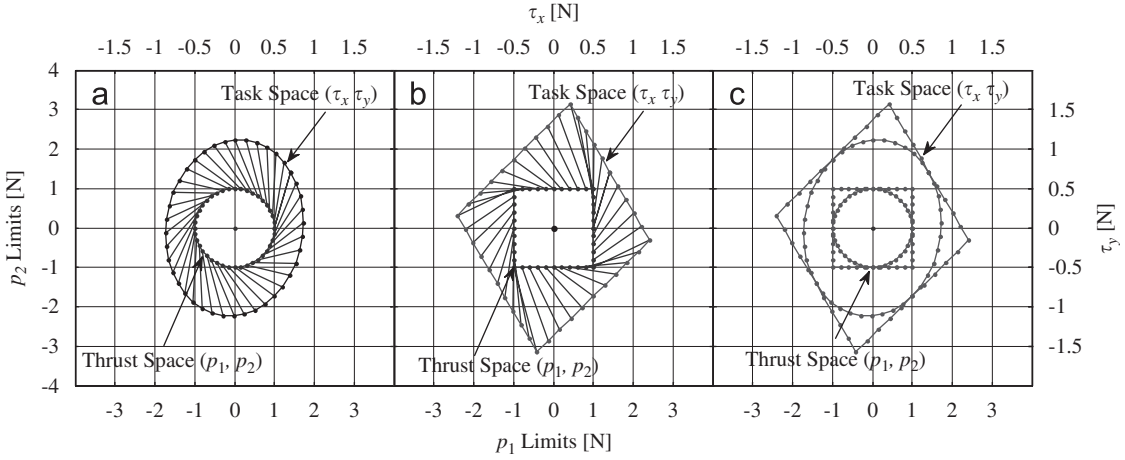


Fig. 3. The lines that connect inner and outer shapes represent the linear transformations.

linear transformation according to $\boldsymbol{\tau} = \mathbf{E}\boldsymbol{p}$ as illustrated in Fig. 3a. In the l_∞ norm case, however, the feasible thrust space points are enclosed by a square $||\boldsymbol{p}||_\infty = \max\{|p_1|, |p_2|\} \leqslant 1$ (an n-dimensional parallelepiped in $\mathbb{R}^n$), and the corresponding task space is enclosed by a polygon (a polytope in $\mathbb{R}^n$) obtained from the same linear transformation as illustrated in Fig. 3b. Now consider a feasible solution point $p_1 = p_2 = 1$ which corresponds to the upper right vertex of the square. While this feasible point is contained by the square, it is excluded by the circle rendering the otherwise feasible point infeasible. Similarly, the associated value of $p_1 = p_2 = 1$ in the task space (a vertex in the polygon) is not contained by the ellipse. The unfeasible regions of the 2-norm solution are illustrated in Fig. 3c where the circle and ellipse are inscribed within the square and polygon, respectively.

To extend this to an example, consider an ROV that is to be driven in horizontal motion (*X*-*Y* plane) with four thrusters. The thruster saturation limit for each thruster is ± 900 N. Assume that a generalized force vector of $\boldsymbol{\tau} = [1800 \quad 0 \quad 0; \quad 0 \quad 0 \quad 2742]^{\mathrm{T}}$ N; N m at the ROV centre of mass is commanded by the on-board controller. If the solution to $\boldsymbol{\tau} = \mathbf{E}\boldsymbol{p}$ is realized based on the minimization of the l_∞ norm, then the thrust vector of $\boldsymbol{p}_{l_\infty} = [900 \quad 900 \quad 900 \quad -900]^{\mathrm{T}}$ N is obtained. This corresponds to a vertex point on the polytope containing all the feasible points. However, if the distribution is realized based on the minimization of the l_2 norm, then the thrust vector of $\boldsymbol{p}_{l_2} = [1292 \quad 671 \quad 508 \quad -671]^{\mathrm{T}}$ N is obtained. This corresponds to a point outside the hypersphere formed by the approximation of the feasible points, and therefore, it is infeasible. This example shows that there exist solutions where minimizing l_∞ norm gives feasible solutions whereas minimizing l_2 norm does not. The practical outcome of this could be disastrous since the first component of $\boldsymbol{p}_{l_2}$ is truncated to the upper limit at the first thruster, the solution vector no longer satisfies $\boldsymbol{\tau} = \mathbf{E}\boldsymbol{p}$. As a consequence, the controller demand cannot be realized, and the controller fails. A similar discussion for the force capability of parallel manipulators is presented by Firmani et al. (2007). In general, since the l_∞ norm minimizes the maximum component of the thrust manifold, it allows the thrusters to run within a safer range than the l_2 norm providing more manoeuvrability for subsequent controller action.

3.3. Infinity-norm formulation

The l_∞ norm of the thruster force manifold $\boldsymbol{p} = [p_1 \quad p_2 \quad \cdots \quad p_n]^{\mathrm{T}}$ is defined as

$$||\boldsymbol{p}||_\infty = \max\{|p_1| \quad |p_2| \quad \cdots \quad |p_n|\} = \max_{1 \leqslant i \leqslant n} \left|\mathbf{I}_i^{\mathrm{T}} \boldsymbol{p}\right| \tag{20}$$

where $|\cdot|$ is the absolute value of each component of the thrust manifold, and $\mathbf{I}_i$ is the ith column vector of the identity matrix $\mathbf{I} \in \mathbb{R}^{n \times n}$. Taking into account the thruster saturation limits, the bounded l_∞ norm thrust allocation problem can be cast as the following constrained optimization problem:

$$\begin{aligned} &\text{minimize} \quad ||\boldsymbol{p}||_\infty \\ &\text{subject to} \quad \boldsymbol{\tau} = \mathbf{E}\boldsymbol{p}, \quad p_{i,l} \leqslant p_i \leqslant p_{i,u}, \quad i = 1, \ldots, n \end{aligned} \tag{21}$$

where $p_{i,\,l}$ and $p_{i,\,u}$ are the lower and upper thruster saturation limits, respectively. Redefining the objective function as

$$l = ||\boldsymbol{p}||_\infty = \max_{1 \leqslant i \leqslant n} \left|\mathbf{I}_i^{\mathrm{T}} \boldsymbol{p}\right| \tag{22}$$

leads to the following equivalent linear programming problem:

$$\begin{aligned} &\text{minimize} \quad l \\ &\text{subject to} \quad \max_{1 \leqslant i \leqslant n} \left|\mathbf{I}_i^{\mathrm{T}} \boldsymbol{p}\right| \leqslant l, \\ &\qquad\qquad \boldsymbol{\tau} = \mathbf{E}\boldsymbol{p}, \quad p_{i,l} \leqslant p_i \leqslant p_{i,u}, \quad i = 1, \ldots, n. \end{aligned} \tag{23}$$

The linear problem of Eq. (23) can be written in the following compact matrix form:

$$\begin{aligned} &\text{minimize} \quad \boldsymbol{c}^{\mathrm{T}} \boldsymbol{y} \\ &\text{subject to} \quad \mathbf{A}_1 \boldsymbol{y} \geqslant \boldsymbol{b}_1, \quad \mathbf{A}_2 \boldsymbol{y} = \boldsymbol{b}_2 \end{aligned} \tag{24}$$

where

$$\mathbf{A}_1 = \begin{bmatrix} -\mathbf{I} & \boldsymbol{I}_n \\ \mathbf{I} & \boldsymbol{I}_n \\ -\mathbf{I} & \boldsymbol{0}_n \\ \mathbf{I} & \boldsymbol{0}_n \end{bmatrix} \in \mathbb{R}^{4n \times (n+1)}, \quad \boldsymbol{b}_1 = \begin{bmatrix} \boldsymbol{0}_n \\ \boldsymbol{0}_n \\ -\boldsymbol{p}_u \\ \boldsymbol{p}_l \end{bmatrix} \in \mathbb{R}^{4n},$$

$$\mathbf{A}_2 = [\mathbf{E} \quad \boldsymbol{0}_m] \in \mathbb{R}^{m \times (n+1)}, \quad \boldsymbol{b}_2 = \boldsymbol{\tau} \in \mathbb{R}^m,$$

$$\boldsymbol{c} = [0 \quad 0 \quad \cdots \quad 0 \quad 1]^{\mathrm{T}} \in \mathbb{R}^{n+1}, \quad \boldsymbol{y} = [\boldsymbol{p} \quad l]^{\mathrm{T}} \in \mathbb{R}^{n+1}$$

with $\boldsymbol{I}_n = [1 \quad 1 \quad \cdots \quad 1]^{\mathrm{T}} \in \mathbb{R}^n$ and $\mathbf{I} \in \mathbb{R}^{n \times n}$ denoting the identity matrix. The term $\boldsymbol{p}_u = [p_{1,u} \quad p_{2,u} \quad \cdots \quad p_{n,u}]^{\mathrm{T}} \in \mathbb{R}^n$ is the upper saturation vector, and the term $\boldsymbol{p}_l = [p_{1,l} \quad p_{2,l} \cdots p_{n,l}]^{\mathrm{T}} \in \mathbb{R}^n$ is the lower saturation vector with $\boldsymbol{0}_n \in \mathbb{R}^n$ and $\boldsymbol{0}_m \in \mathbb{R}^m$ being the zero column vectors.

3.4. Fault-tolerant property

In the current work, joint failure refers to two incidents: first, when a thruster breaks down; second, when a thruster partly loses its driving capacity. A general fault-tolerant scheme detects faulty thrusters, and then applies a strategy in which the

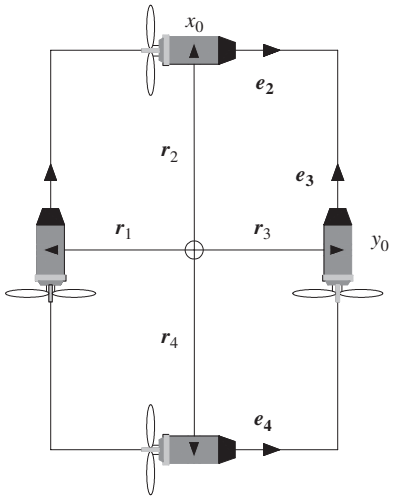


Fig. 4. The redundant thruster layout.

demanded controller output can be reallocated among the functioning thrusters.

In order to account for the faulty thrusters, the linear programming problem given in Eq. (24) must be modified. To this end, a diagonal weighting matrix can be used. The diagonal elements of the weighting matrix penalize the upper and lower saturation limits of faulty thrusters such that demanded thrust does not exceed the available capacity of the faulty thrusters. The diagonal weighting matrix can be defined as

$$\mathbf{W} = \operatorname{diag}(w_1 \quad w_2 \quad \cdots \quad w_n) \in \mathbb{R}^{n\times n}$$
$$w_i = \begin{cases} 0 & \text{if the } i\text{th thruster is in failure} \\ 0<w_i<1 & \text{if the } i\text{th thruster is in partial failure;} \quad i=1,\ldots,n. \\ 1 & \text{if the } i\text{th thruster is not in failure} \end{cases} \tag{25}$$

The fault-tolerant property is incorporated into the optimization formulation by redefining the vector $\boldsymbol{b}_1$ as

$$\bar{\boldsymbol{b}}_1 = \begin{bmatrix} \boldsymbol{0}_n & \boldsymbol{0}_n & -\bar{\boldsymbol{p}}_u & \bar{\boldsymbol{p}}_l \end{bmatrix}^{\mathrm{T}} \tag{26}$$

where

$$\bar{\boldsymbol{p}}_u = \mathbf{W}\boldsymbol{p}_u \quad \text{and} \quad \bar{\boldsymbol{p}}_l = \mathbf{W}\boldsymbol{p}_l.$$

Finally, the fault-tolerant, bounded l_∞ norm thrust allocation problem is formulated as

$$\begin{aligned} &\text{minimize} && \boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} \\ &\text{subject to} && \mathbf{A}_1\boldsymbol{y} \geqslant \bar{\boldsymbol{b}}_1, \ \mathbf{A}_2\boldsymbol{y} = \boldsymbol{b}_2 \end{aligned} \tag{27}$$

3.5. Primal-dual problem

The linear problem of Eq. (27) can be transformed into an equivalent optimization problem in terms of its Lagrange multipliers. In this context, the transformed problem is called the *dual* problem whereas the original problem is referred to as the *primal* problem (Wolfe, 1961). In view of this, the linear programming problem of Eq. (27) will be called the primal problem hereafter.

Theorem 2. *Let $\boldsymbol{y}^* \in \mathbb{R}^n$ be the solution of the primal problem given in Eq. (27). Consider $\boldsymbol{\mu} \in \mathbb{R}^{4n}$ and $\boldsymbol{\lambda} \in \mathbb{R}^m$ to be the associated Lagrange multipliers (also called dual decision variables) related to the inequality and equality constraints of the primal problem. If $\boldsymbol{y}^*$ is a regular point of the constraints (i.e., it satisfies the equality constraints and A2 has full row rank), then $\boldsymbol{y}^*$, $\boldsymbol{\mu}^*$ and $\boldsymbol{\lambda}^*$ solve the dual problem:*

$$\begin{aligned} &\text{maximize} && L(\boldsymbol{y},\boldsymbol{\mu},\boldsymbol{\lambda}) \\ &\text{subject to} && \nabla_y L(\boldsymbol{y},\boldsymbol{\mu},\boldsymbol{\lambda}) = 0 \\ & && \boldsymbol{\mu} \geqslant 0 \end{aligned} \tag{28}$$

where $L(\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\lambda})$ is the Lagrangian function and $\nabla_y L(\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\lambda})$ is the gradient of the Lagrangian function. In addition, $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y}^* = L(\boldsymbol{y}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$.

Proof. See Antoniou and Lu (2007).

According to Theorem 2, the objective functions of the primal and dual problems reach to the same value at their solution point. The difference is called duality gap, i.e., $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} - L(\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\lambda})$, which will be used in Section 4.2 in the derivation of an energy function.

In an attempt to obtain the dual problem of the primal problem, the Lagrangian function of Eq. (27), $L(\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\lambda})$, is defined as

$$L(\boldsymbol{y},\boldsymbol{\mu},\boldsymbol{\lambda}) = \boldsymbol{y}^{\mathrm{T}}(\boldsymbol{c} - \mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} - \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda}) + \bar{\boldsymbol{b}}_1^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{\lambda} \tag{29}$$

The gradient of Eq. (29) with respect to the primal variable $\boldsymbol{y}$ is

$$\nabla_y L(\boldsymbol{y},\boldsymbol{\mu},\boldsymbol{\lambda}) = \boldsymbol{c} - \mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} - \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda} \tag{30}$$

Applying Theorem 2 to Eq. (27) with the definitions of Eqs. (29) and (30) yields the dual problem as

$$\begin{aligned} &\text{maximize} && \boldsymbol{y}^{\mathrm{T}}(\boldsymbol{c} - \mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} - \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda}) + \bar{\boldsymbol{b}}_1^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{\lambda} \\ &\text{subject to} && \boldsymbol{c} - \mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} - \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda} = 0 \\ & && \boldsymbol{\mu} \geqslant 0 \end{aligned} \tag{31}$$

Using the equality constraint directly within the objective function, Eq. (31) can be simplified to

$$\begin{aligned} &\text{maximize} && \bar{\boldsymbol{b}}_1^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{\lambda} \\ &\text{subject to} && \mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} + \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda} - \boldsymbol{c} = 0 \\ & && \boldsymbol{\mu} \geqslant 0, \quad \boldsymbol{\lambda} \text{ unrestricted} \end{aligned} \tag{32}$$

Eq. (32) corresponds to the dual problem of the primal problem.

4. Solution with recurrent neural network

4.1. Neural network design preliminaries

The solution of the primal problem, Eq. (27), and the dual problem, Eq. (32), is the vector $\boldsymbol{y}$ containing the thrust manifold $\boldsymbol{p}$. Since real-time updates to the thrust manifold are required, the linear programme in the previous section must be solved in real time. However, the existing sequential algorithms such as simplex methods (Gill et al., 1991) and interior point methods (Wright, 1992) are often not capable of delivering solutions at this rate. One promising approach to obtaining real-time solutions is a recurrent neural network on the basis of an analogue circuit. A neural network can satisfy real-time requirements due to its fast parallel processing nature. In addition, as long as there is a solution to the primal problem, a properly designed recurrent neural network will converge to the optimal solution without the requirement of a feasible initial point.

In what follows, a recurrent neural network will be designed to solve the primal problem following the gradient model derivation outlined by Xia and Wang (1998). The premise of the gradient model derivation is to design a dynamic system whose state variables naturally converge to the solution of the optimization problems of Eqs. (27) and (32).

4.2. Energy function derivation

In an attempt to design a recurrent neural network, an energy function can be defined using the synergy between the primal problem and its dual problem. In view of this, the following continuously differentiable, convex, positive definite, energy-like function $E(\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\lambda})$ is proposed for the neural network design:

$$E = \tfrac{1}{2}(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} - \bar{\boldsymbol{b}}_1^{\mathrm{T}}\boldsymbol{\mu} - \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{\lambda})^2 + \tfrac{1}{2}\left\|\mathbf{A}_2\boldsymbol{y} - \boldsymbol{b}_2\right\|_2^2 + \tfrac{1}{2}\left\|\mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} + \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda} - \boldsymbol{c}\right\|_2^2 + \tfrac{1}{4}(\mathbf{A}_1\boldsymbol{y} - \bar{\boldsymbol{b}}_1)^{\mathrm{T}}(\mathbf{A}_1\boldsymbol{y} - \bar{\boldsymbol{b}}_1 - |\mathbf{A}_1\boldsymbol{y} - \bar{\boldsymbol{b}}_1|) + \tfrac{1}{4}\boldsymbol{\mu}^{\mathrm{T}}(\boldsymbol{\mu} - |\boldsymbol{\mu}|) \quad (33)$$

The first term in Eq. (33) is the squared difference (duality gap) between the objective functions of the primal and dual problems. The second and the third terms are the equality constraints associated with the primal and dual problems. Finally, the last two terms are associated with the inequality constraints of the same primal and dual optimization problems.

4.3. Gradient model derivation

Since (33) is a convex function whose minimum is at zero, its solution corresponds to the solution of the following optimization problem (Xia, 1996; Xia and Wang, 1998):

$$\begin{aligned} &\text{minimize} \quad E(\boldsymbol{v}) \\ &\text{subject to} \quad \boldsymbol{v} \in \mathbb{R}^k \end{aligned} \quad (34)$$

where $\boldsymbol{v} = [\boldsymbol{y} \quad \boldsymbol{\mu} \quad \boldsymbol{\lambda}]^{\mathrm{T}}$.

To transform the minimization problem (34) into an associated system of ordinary differential equations that govern the recurrent neural network, the gradient of the energy-like function can be used:

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} = -\kappa \nabla E(\boldsymbol{v}) \quad (35)$$

and $\kappa \in \mathbb{R}$ is a strictly positive number defining the convergence rate of the optimization problem of (34) (i.e., greater values provide faster convergence rates), and $\nabla E(\boldsymbol{v})$ is the gradient of the energy-like function. The equations given in Eq. (35) can be written more explicitly as

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} &= -\kappa \nabla_y E(\boldsymbol{v}) = -\kappa[\boldsymbol{c}(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} - \bar{\boldsymbol{b}}_1^{\mathrm{T}}\boldsymbol{\mu} - \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{\lambda}) \\ &\quad + \mathbf{A}_2^{\mathrm{T}}(\mathbf{A}_2\boldsymbol{y} - \boldsymbol{b}_2) + \mathbf{A}_1^{\mathrm{T}}(\mathbf{A}_1\boldsymbol{y} - \bar{\boldsymbol{b}}_1)^-] \\ \frac{\mathrm{d}\boldsymbol{\mu}}{\mathrm{d}t} &= -\kappa \nabla_\mu E(\boldsymbol{v}) = -\kappa[-\bar{\boldsymbol{b}}_1(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} - \bar{\boldsymbol{b}}_1^{\mathrm{T}}\boldsymbol{\mu} - \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{\lambda}) \\ &\quad + \mathbf{A}_1(\mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} + \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda} - \boldsymbol{c}) + \boldsymbol{\mu}^-] \\ \frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}t} &= -\kappa \nabla_\lambda E(\boldsymbol{v}) = -\kappa[-\boldsymbol{b}_2(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} - \bar{\boldsymbol{b}}_1^{\mathrm{T}}\boldsymbol{\mu} - \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{\lambda}) \\ &\quad + \mathbf{A}_2(\mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} + \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda} - \boldsymbol{c})] \end{aligned} \quad (36)$$

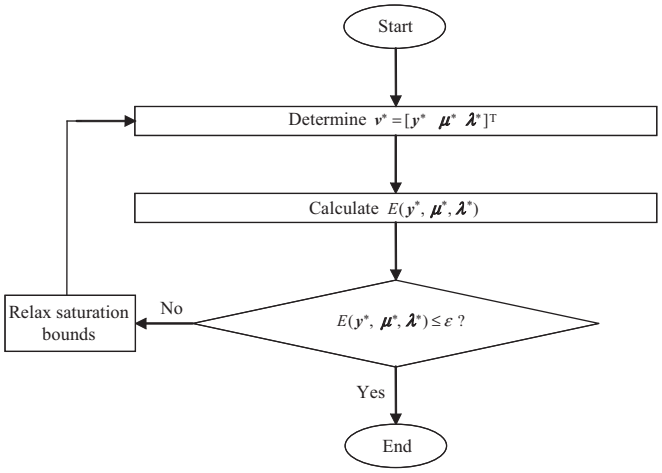


Fig. 6. Flow chart of the solution determination.

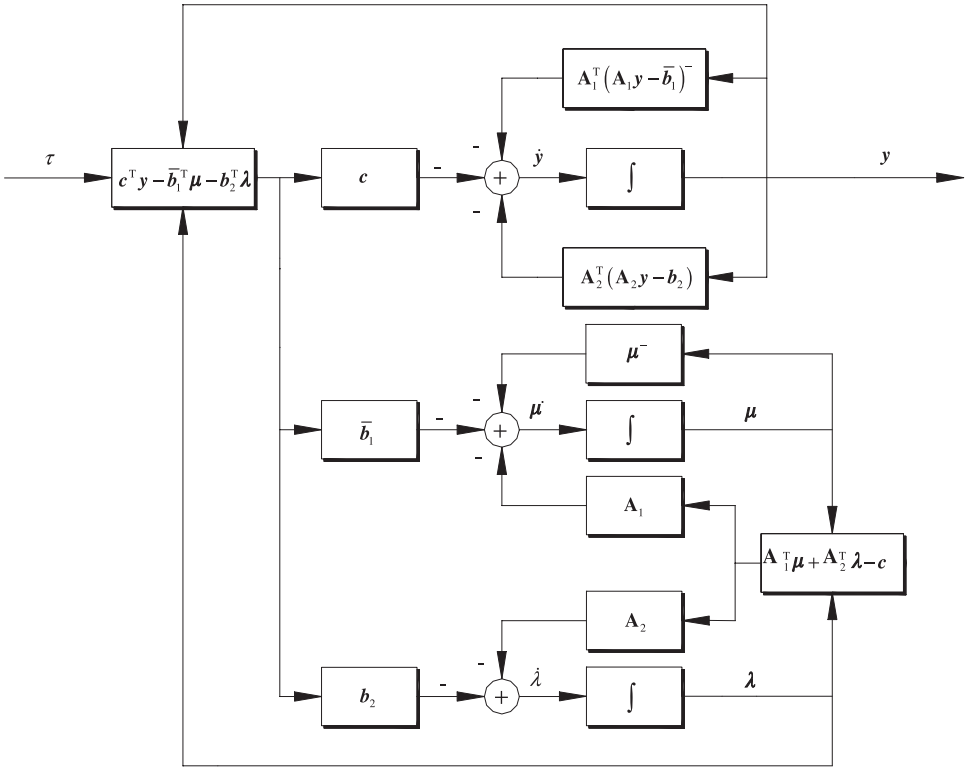


Fig. 5. Block diagram for the architecture of the neural network for the l_∞ norm thrust allocation.

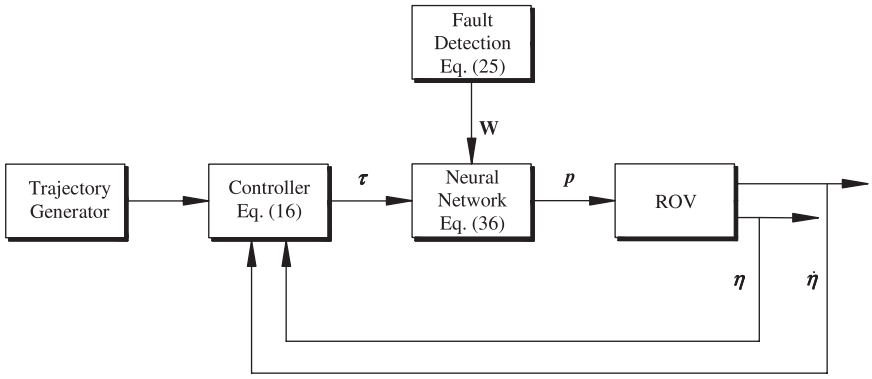


Fig. 7. Block diagram for the controller-neural network scheme.

Table 1
ROPOS hydrodynamic parameters

Coefficients	Surge	Lateral	Heave	Roll	Pitch	Yaw
Linear drag	725 N s/m	1240 N s/m	825 N s/m	3000 N s/m	3000 N s/m	1804 N s/m
Quad. drag	1000 N s^2/m^2	525 N s^2/m^2	400 N s^2/m^2	100 N s^2	100 N s^2	72 N s^2
Added mass	−4380 kg	−9518 kg	−4268 kg	−5000 kg m^2	−5000 kg m^2	−5000 kg m^2

where for any column vector $\boldsymbol{\zeta}$, the symbol $\boldsymbol{\zeta} - |\boldsymbol{\zeta}| = 2(\boldsymbol{\zeta})^-$ with $(\boldsymbol{\zeta})^- = \begin{bmatrix} \zeta_1^- & \zeta_2^- & \cdots & \zeta_n^- \end{bmatrix}^T$ and $\zeta_i^- = \min\{0 \quad \zeta_i\}$. Fig. 5 illustrates the block diagram for the architecture of the recurrent neural network defined in Eq. (36).

Theorem 3. *The neural network given in Eq. (36) is globally stable and will converge to the optimal solutions of* $\boldsymbol{v}^* = \begin{bmatrix} \boldsymbol{y}^{*T} & \boldsymbol{\mu}^{*T} & \boldsymbol{\lambda}^{*T} \end{bmatrix}^T$ *that corresponds to the solution of the primal problem, Eq. (27), and its dual problem, Eq. (32), respectively.*

Proof. See Appendix B.

Note that neural network design challenges such as the generation of training data, the validation of the trained neural network, the determination of type of activation functions, the number of neurons, etc. are eliminated through the hardware implementation of the proposed recurrent neural network, and are not addressed in the current work. For the hardware implementation, neural network dynamic equations are derived by the current research. A detailed discussion regarding the creation of an integrated circuit from neural network dynamic equations is given by Wang (1993).

4.4. Nonexistence of solution

In practice, there are two cases where there is no solution. First case is when there are too many faulty thrusters and the remaining thruster forces can no longer span the task space. Second case is when a solution does not exist within the saturation limits. With regards to the second case, the energy function given in Eq. (33) can be used to determine whether or not the neural network converged to the true solution; if the energy function is not within an allowable tolerance, i.e., $E(\boldsymbol{y}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*) > \varepsilon$, where ε is the allowable tolerance and is set to nearly zero, then it can be concluded that the inequality constraints of the primal problem could not be satisfied. In this case, a heuristic strategy is applied to generate the best approximate solution to the thrust allocation: the upper and lower saturation limits are extended which translates into relaxing the solution range on $\boldsymbol{p}$. The relaxed problem is solved using the same neural network. Finally, the thrusts p_i within the relaxed solution that do exceed saturation limits are truncated to their actual upper or lower saturation limits, and the resulting thrust manifold $\boldsymbol{p}$ is applied to the ROV. This process is outlined in Fig. 6. A similar truncation approach was applied by Bordignon (1996).

Table 2
ROPOS inertia parameters

Mass (kg)	I_x (N m s^2)	I_y (N m s^2)	I_z (N m s^2)	Ixy (N m s^2)	Iyz (N m s^2)	Ixz (N m s^2)
2268	1937	2883	2457	0	0	0

Table 3
Controller parameters

$\boldsymbol{\Lambda}$ = diag(5 5 0 0 0 5)
$\mathbf{K}$ = diag(500 500 0 0 0 500)
$\mathbf{G}$ = diag(500 500 0 0 0 1500)

Fig. 7 depicts the block diagram for the controller-neural network scheme. The output command vector $\boldsymbol{\tau}$ of the adaptive-sliding-mode control in combination with the output of the fault detection algorithm $\mathbf{W}$ are fed into the neural network. The neural network then generates an output signal $\boldsymbol{y}$ containing the solution vector $\boldsymbol{p}$ of the thrust allocation problem. Note that it is assumed that the thruster failure can be detected when it occurs. A detailed discussion on fault-detection methodologies is given by Omerdic and Roberts (2004).

5. Simulation and discussion

5.1. ROV system overview

Simulation studies were performed in order to demonstrate the effectiveness of the proposed control and thruster allocation scheme. The ROV system considered in the simulations is the Canadian Scientific Submersible Facility ROPOS vehicle. The

ROPOS vehicle is rectangular in shape with length, width, and height being 1.75, 2.6, and 1.45 m, respectively. The system has 3 pairs of electro-hydraulic thrusters: one pair of aft thrusters drives the surge and yaw motion; a second pair activates the heave direction through openings in the foam pack; and a third pair, directs lateral motions. The thruster saturation limit for each thruster is ± 900 N. The thruster layout on x–y axis of the vehicle is illustrated in Fig. 4. Dynamic parameters of the ROPOS vehicle are taken from Steinke (2006) and tabulated in Tables 1 and 2.

In the current simulation study, surge sway and yaw position and velocity state feedback is available through the onboard ROPOS navigation system. The ocean current of 0.5 knots along the positive X and Y axis of the inertial frame is considered in the simulation study. The system is assumed to be neutrally buoyant. In addition, the thruster dynamic response is assumed to be much faster than that of the ROV itself, and thus was neglected.

5.2. First simulation study

In this study, ROPOS obstacle-avoidance manoeuvres were simulated in the vehicle's local level plane. ROPOS was driven through a series of waypoints: $A\,(0, 0; 0)$, $B\,(2, 2; \pi/4)$, $C\,(4, 4; \pi/2)$, $D\ (2,\ 6;\ 3\pi/4)$, $E\ (0,\ 8;\ \pi)$ m; rad where the first two digits represent the absolute position values, and the last digit represents the yaw orientation value. Each segment of the path was traversed in 50 s. A corresponding continuous set of trajectory path values was generated using a fifth-order polynomial function.

The new adaptive method was implemented and used to drive the ROPOS vehicle through the manoeuvre. To reflect uncertainties that are present in the underwater vehicle paradigm, 40% modelling inaccuracies were incorporated into the controller's dynamic model. In addition, a constant disturbance moment of

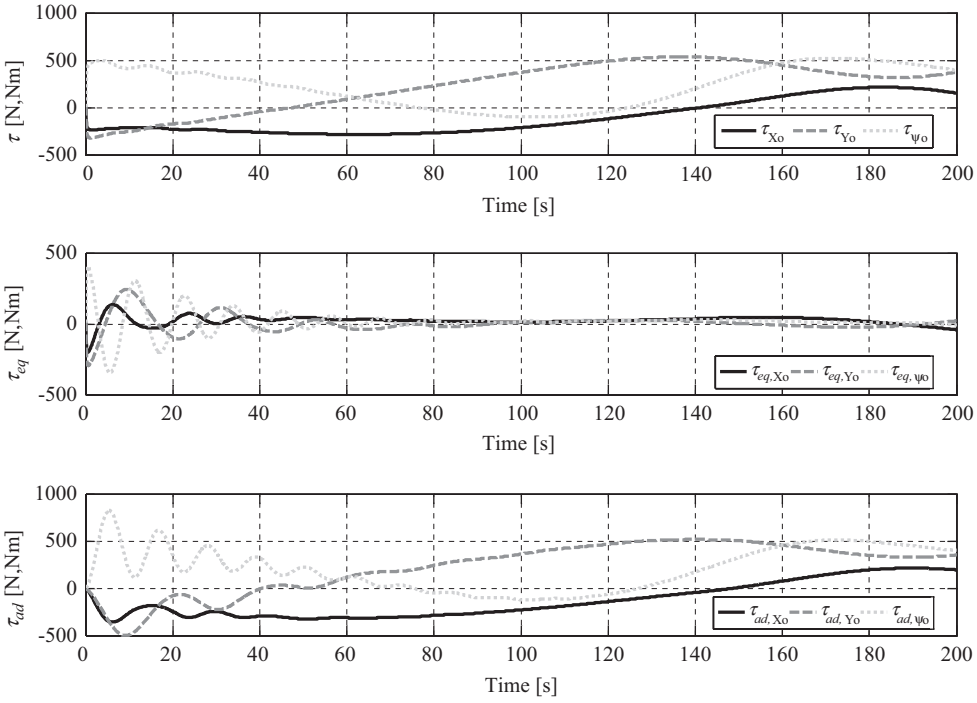


Fig. 8. Chattering-free adaptive sliding-mode controller output.

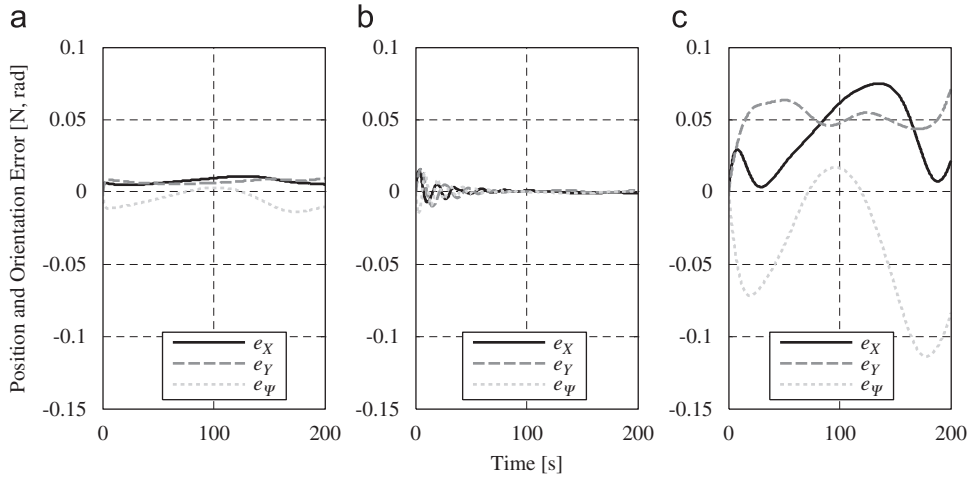


Fig. 9. Position and orientation error with respect to the inertial frame: (a) sliding-mode with boundary layer; (b) adaptive law is ON; (c) adaptive law is OFF.

150 N m is applied to the yaw axis of the vehicle to demonstrate the disturbance rejection capability of the controller. The knowledge of constant disturbance moment and the ocean current are not included into the controller's dynamic model to introduce further error. Controller parameters are given in Table 3. The controller output is given in Fig. 8.

The control law $\boldsymbol{\tau}$ and its equivalent and adaptive components, $\boldsymbol{\tau}_{eq}$ and $\boldsymbol{\tau}_{ad}$, are shown in Fig. 8. As can be seen, the proposed adaptive sliding-mode controller's on-line adaptation law yields a smooth control output without high-frequency controller demand, and therefore is chattering-free. In addition, the proposed controller provides excellent performance in the trajectory—following task is shown in Fig. 9b. This high performance is achieved despite the presence of high uncertainties that are illustrated in Fig. 10a.

As the vehicle moves, the controller performance significantly increases as shown in Fig. 9b. This can be attributed to the adaptation term: it learns as the vehicle progresses over its path, and as a consequence, begins to make better estimation of the lumped uncertainty vector as demonstrated in Fig. 10b and c. This high accuracy in the adaptive estimation justifies the choice of adaptive function given in Eq. (15). The estimation of the uncertainty is incorporated into the control law according to Eq. (16) leading to the improvement in the dynamic model inside the controller. The controller performance without the adaptation term is relatively poor as demonstrated in Fig. 9c. This is due to the fact that when the adaptation term is turned off, only the equivalent control law acts on the system. In this case, the performance relies heavily on the quality of the dynamic model inside the controller, which is poor due to the high uncertainty. The validity of Assumption 2 can be given by the plot of Eq. (43) which is given in Fig. 11. The initial positive value is due to the initial conditions of the simulation and is very quickly eliminated by the controller as the adaptive term begins to close the gap between the exact and estimated unknown dynamics.

A comparison study was performed to reveal the performance difference between two chattering-free sliding-mode approaches: one uses a conventional boundary layer approach (Slotine and Li, 1991) and the other is the proposed adaptive approach. The boundary layer width was set to 0.15 to ensure chattering-free response. The adaptive approach outperforms the boundary layer approach as shown in Fig. 9a, and the superior performance can be credited to the adaptive capability of the new method.

Fig. 12a demonstrates the l_∞ thrust output obtained from the designed neural network. In the simulations, the state dynamic equations of Eq. (36) were solved using the MATLAB *ode15s* command with $\kappa = 1 \times 10^6$. To have a comparison study, the thruster allocation was also completed using the l_2 norm minimization, i.e., the pseudo-inverse, and the results are demonstrated in Fig. 12b. Fig. 13 depicts the history of the largest single commanded thrust produced by the l_2 norm and l_∞ norm methods during the ROPOS manoeuvre. The result shows that the largest thrust value obtained from the l_2 norm minimization and l_∞ norm minimization are 342.6 and 268.4 N, respectively. Therefore, l_∞ norm minimization reduces the largest thrust by 21.6% in comparison to the one obtained from the l_2 norm minimization. The average improvement over the whole manoeuvre is found to be 11.9%. The results are given in Table 4. The results illustrate that when the lowest possible thrust magnitudes are desired to ensure future ROV manoeuvrability for subsequent control iterations, the l_∞ norm thrust allocation is an attractive alternative to the conventional pseudo-inverse solution. Note that since both l_∞ and l_2 generate the same control input based on $\boldsymbol{\tau} = \boldsymbol{Ep}$, there is no difference in the controller performance.

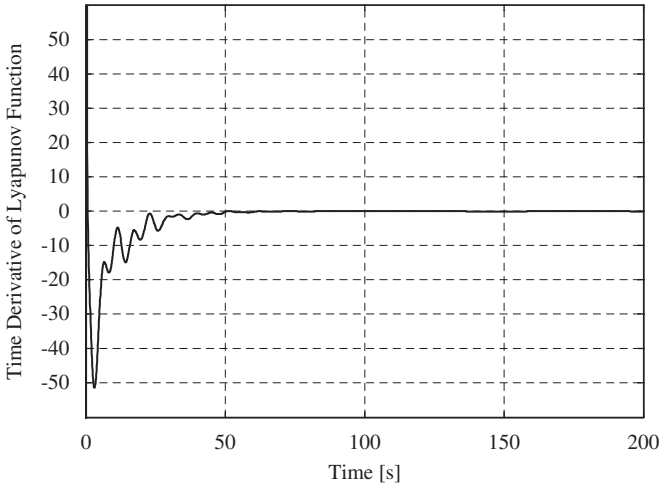


Fig. 11. Time derivative of Lyapunov function.

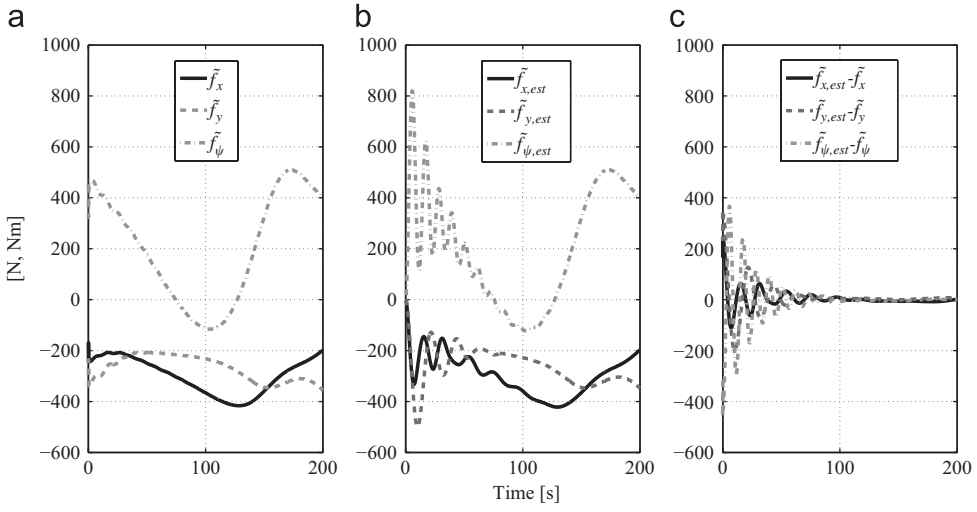


Fig. 10. Lumped uncertainty estimations: (a) real lumped uncertainty; (b) estimation of lumped uncertainty; (c) difference between real and lumped estimation.

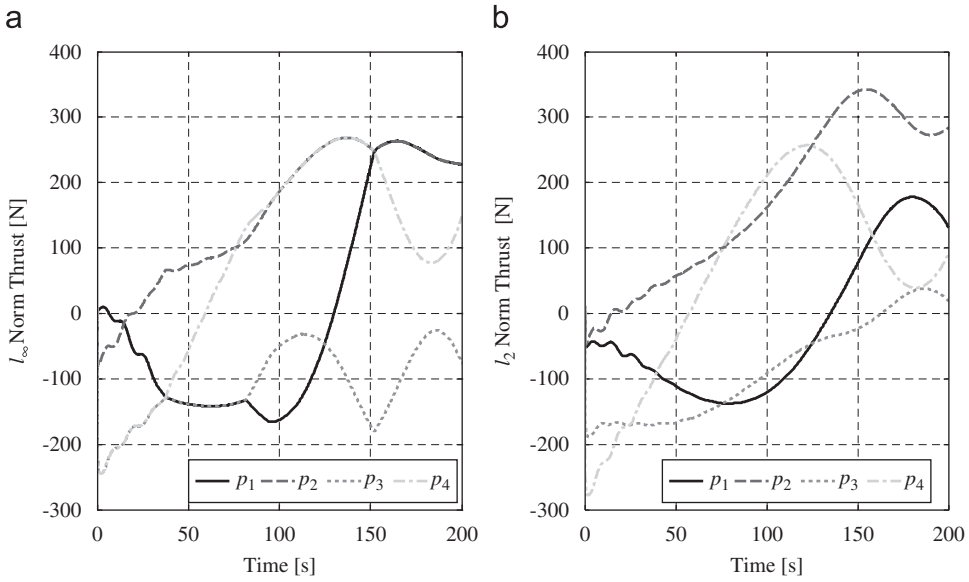


Fig. 12. Demanded thrusts: (a) l_∞ approach; (b) l_2 approach.

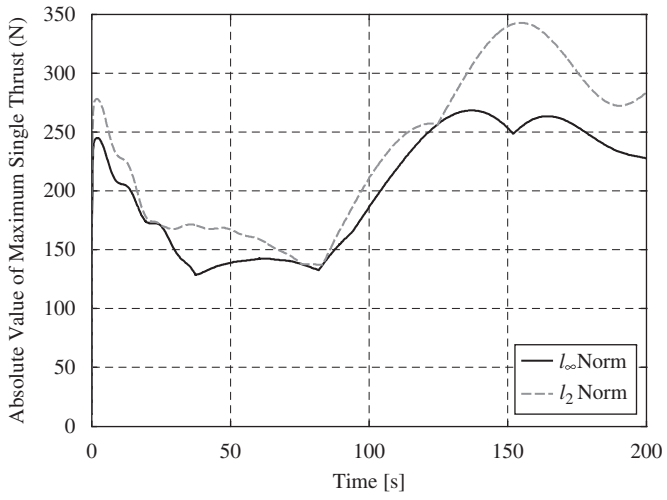


Fig. 13. Maximum single output of the l_∞ norm solution $\boldsymbol{p}_{l_\infty}$, and l_2 norm solution $\boldsymbol{p}_{l_2}$.

Table 4
Comparison of the norms

Max l_2 norm thrust (N)	Max l_∞ norm thrust (N)	Reduction (%)	Average reduction (%)
342.6	268.4	21.6	11.9

5.3. Second simulation study

In the second simulation study, the first case study was extended to include a thruster failure to demonstrate fault-tolerant property of the thrust allocation scheme.

It is assumed that the second and third thrusters partially fail between 40 and 150 s due to a temporary fouling of the propeller. The fouling produces a drop of 78% and 20% in the driving capacity of the second and third thrusters, respectively. This incident is accounted for in the thruster allocation scheme by setting $w_2 = 2/9$ and $w_3 = 4/5$. A complete failure of the second thruster occurs at 150 s, and for $t > 150$ s, the horizontal ROPOS manoeuvre is accomplished with its three functioning thrusters. Note that since the number of the remaining thrusters is the same as the number of the task space dimensions, the thruster allocation problem of $\boldsymbol{\tau} = \boldsymbol{Ep}$ becomes determined. In this case, the neural network converges to the unique solution of $\boldsymbol{\tau} = \boldsymbol{Ep}$ assuming a solution exists within the specified saturation range.

As Fig. 14a reveals, the faulty second thruster was driven to its upper saturation limit of 200 N at the time between 104 and 150 s. After the complete failure that occurs at 150 s, the faulty first thruster is not commanded and the remaining three thrusters are used by the fault-tolerant scheme to meet the controller demand. The third faulty thruster functions within its limits throughout the manoeuvre. Note that the instantaneous changes in the demanded thrust are caused by the sudden thruster failures. As Fig. 14b reveals, the controller performance is not affected by the faulty thrusters, and that the vehicle keeps the normal working process in the presence of faults and saturation limits. This means that the thrust allocation scheme is capable of finding a feasible solution in the smaller solution space due to the thrust failures. This result demonstrates the ability of the thruster allocation scheme to account for the thruster failure and saturation limits.

6. Conclusion

The contribution of the current work is twofold. First, a chattering-free sliding-mode controller has been proposed for ROV systems. The new controller uses an adaptive term in place of the conventional, discontinuous switching term. This adaptive term continuously compensates for the unknown system dynamics caused by poorly approximated nonlinear hydrodynamics or sudden environmental loads. As opposed to conventional sliding-mode controllers, the proposed controller does not need a *priori* knowledge of the upper bounds on the dynamics parameters of ROVs. In addition, the adaptive term does not require the parameterization of a regressor matrix and unknown parameter vector. Second, the l_∞ norm-based thrust allocation of ROVs has been proposed. It has been shown that a thrust allocation based on minimizing the l_∞ norm of the thrust manifold can be formulated as a linear programming problem that allows direct incorporation of faulty thrusters and thruster saturation limits. A recurrent neural network has been proposed

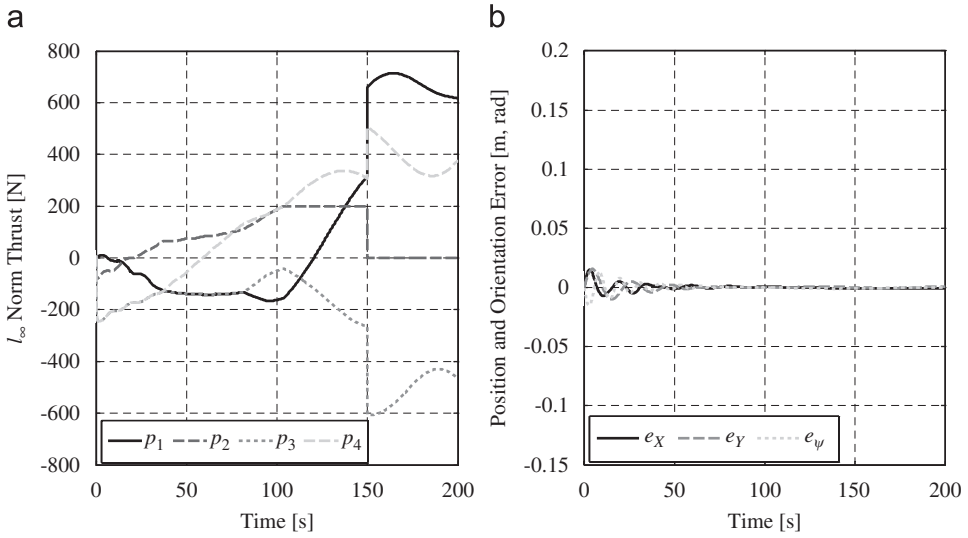


Fig. 14. (a) Demanded thrust from the l_∞ norm approach; (b) position and orientation error.

to solve the resulting linear programming problem. To this end, a general neural network design technique has been extended to include bounds on the thrust variables. The proposed neural network has the ability to achieve the real-time solution rates necessary in the ROV application. The resulting neural network has been applied to minimize the thrust allocation instantaneously with thruster saturation points and thruster failure considered. Future work will involve experimentally validating the proposed controller-thrust allocation scheme on an existing ROV system. This process will involve hardware realization of the recurrent neural network.

Acknowledgment

The authors would like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for providing financial support for this work. The authors also wish to thank Dr. F. Firmani for contributing through several fruitful discussions.

Appendix A. Proof of Theorem 1

Define a Lyapunov function as

$$V = \tfrac{1}{2}(\boldsymbol{s}^{\mathrm{T}}\mathbf{M}_\eta\boldsymbol{s} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w}) \tag{37}$$

where $\boldsymbol{w} = \tilde{\boldsymbol{f}}_{\mathrm{est}} - \tilde{\boldsymbol{f}}$ is the difference vector between the estimated lumped uncertainty vector and the exact lumped uncertainty vector. Differentiating V with respect to time yields

$$\dot{V} = \tfrac{1}{2}(\boldsymbol{s}^{\mathrm{T}}\dot{\mathbf{M}}_\eta\boldsymbol{s} + \dot{\boldsymbol{s}}^{\mathrm{T}}\mathbf{M}_\eta\boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}\mathbf{M}_\eta\dot{\boldsymbol{s}}) + \dot{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} \tag{38}$$

In light of the fact that $\dot{\boldsymbol{s}}^{\mathrm{T}}\mathbf{M}_\eta\boldsymbol{s} = \boldsymbol{s}^{\mathrm{T}}\mathbf{M}_\eta\dot{\boldsymbol{s}}$, and using (10), $\mathbf{M}_\eta\ddot{\boldsymbol{\eta}} = \mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau}_{\mathrm{eq}} - (\mathbf{C}_\eta\dot{\boldsymbol{\eta}} + \mathbf{D}_\eta\dot{\boldsymbol{\eta}} + \boldsymbol{g}_\eta)$ from (2), and $\dot{\boldsymbol{\eta}} = \boldsymbol{s} + \dot{\boldsymbol{\eta}}_r$ from Eq. (8)

$$\dot{V} = \tfrac{1}{2}\boldsymbol{s}^{\mathrm{T}}(\dot{\mathbf{M}}_\eta - 2\mathbf{C}_\eta)\boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}[\mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau} - (\mathbf{M}_\eta\ddot{\boldsymbol{\eta}}_r + \mathbf{C}_\eta\dot{\boldsymbol{\eta}}_r + \mathbf{D}_\eta\dot{\boldsymbol{\eta}} + \boldsymbol{g}_\eta)] + \dot{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} \tag{39}$$

Since $\dot{\mathbf{M}}_\eta - 2\mathbf{C}_\eta$ is a skew symmetric matrix, then $\boldsymbol{s}^{\mathrm{T}}(\dot{\mathbf{M}}_\eta - 2\mathbf{C}_\eta)\boldsymbol{s} = \mathbf{0}$, and Eq. (39) can be simplified to

$$\dot{V} = \boldsymbol{s}^{\mathrm{T}}(\mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau} - \boldsymbol{f}_{\underline{r}}) + \dot{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} \tag{40}$$

where $\boldsymbol{f}_{\underline{r}} = \mathbf{M}_\eta\ddot{\boldsymbol{\eta}}_r + \mathbf{C}_\eta\dot{\boldsymbol{\eta}}_r + \mathbf{D}_\eta\dot{\boldsymbol{\eta}} + \boldsymbol{g}_\eta$. Substituting Eq. (16) into Eq. (40), and using $\dot{\boldsymbol{\eta}}_r = \dot{\boldsymbol{\eta}} - \boldsymbol{s}$ and $\boldsymbol{w} = \tilde{\boldsymbol{f}}_{\mathrm{est}} - \tilde{\boldsymbol{f}}$ yields

$$\dot{V} = \boldsymbol{s}^{\mathrm{T}}(\boldsymbol{w} - \mathbf{K}\boldsymbol{s}) + \dot{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} \tag{41}$$

Substituting the adaptation law of Eq. (15) into Eq. (41) and using $\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma} = \mathbf{I}$, $\boldsymbol{\Gamma}^{\mathrm{T}} = \boldsymbol{\Gamma}$ gives

$$\dot{V} = -\boldsymbol{s}^{\mathrm{T}}\mathbf{K}\boldsymbol{s} - \dot{\tilde{\boldsymbol{f}}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} \tag{42}$$

In view of the positive definiteness of the gain matrix $\mathbf{K}$ and *Assumptions 1 and 2*, it can be stated that

$$\dot{V} \leqslant -\boldsymbol{s}^{\mathrm{T}}\mathbf{K}\boldsymbol{s} - \dot{\tilde{\boldsymbol{f}}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} \leqslant 0 \tag{43}$$

The inequality given in Eq. (43) implies that the system trajectories will converge to the sliding manifold $\boldsymbol{s} = \mathbf{0}$ from any nonzero initial error. However, Eq. (43) alone does not imply that $\dot{V} \to 0$ as $t \to \infty$, i.e., the system trajectories may not converge to the desired values in finite time. This problem can be solved using Barbalat's lemma (Popov, 1973; Slotine and Li, 1991), i.e., since V is lower bounded ($V \geqslant 0$), $\dot{V}$ is negative semi-definite ($\dot{V} \leqslant 0$), and $\ddot{V}$ is bounded (i.e., since $\boldsymbol{w}$ and $\dot{\tilde{\boldsymbol{f}}}$ are bounded, then s is bounded), then $\dot{V} \to 0$ as $t \to \infty$. In turn, $\boldsymbol{e} \to \mathbf{0}$ and $\dot{\boldsymbol{e}} \to \mathbf{0}$ as $t \to \infty$. □

Appendix B. Proof of Theorem 3

The stability proof of the neural network given in Eq. (35) involves the following two lemmas:

Lemma 1. *If $f : C \subset \mathbb{R}^n \to \mathbb{R}$ is differentiable over a convex set $C_c \subset C$, then f is convex over C_c if and only if*

$$(\boldsymbol{x}_1 - \boldsymbol{x})^{\mathrm{T}}\nabla f(\boldsymbol{x}) \leqslant f(\boldsymbol{x}_1) - f(\boldsymbol{x}) \quad \forall \boldsymbol{x}_1,\ \boldsymbol{x} \in C_c \tag{44}$$

where $\nabla f(x)$ is the gradient vector of $f(x)$.

Proof of Lemma 1. See Ortega and Rheinboldt (1970).

Lemma 2. *$\boldsymbol{y}^*$ and $(\boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ are the optimal solutions to the primal problem, Eq. (27), and its dual problem, Eq. (32), respectively, if and only if $E(\boldsymbol{v}^*) = 0$ and*

$$(\boldsymbol{v}^* - \boldsymbol{v})^{\mathrm{T}}\nabla E(\boldsymbol{v}) \leqslant -E(\boldsymbol{v}) \tag{45}$$

where $\boldsymbol{v}^* = [\boldsymbol{y}^{*\mathrm{T}} \quad \boldsymbol{\mu}^{*\mathrm{T}} \quad \boldsymbol{\lambda}^{*\mathrm{T}}]^{\mathrm{T}}$ and $\boldsymbol{v} = \begin{bmatrix}\boldsymbol{y}^{\mathrm{T}} & \boldsymbol{\mu}^{\mathrm{T}} & \boldsymbol{\lambda}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$.

Proof of Lemma 2. According to the definition of the energy function, it is evident that $E(\boldsymbol{v}^*) = 0$. By applying Lemma 1 to the energy function with $E(\boldsymbol{v}^*) = 0$, Lemma 2 can be verified. □

Without loss of generality, let $\kappa = 1$ and consider the following positive definite Lyapunov function candidate for the neural

network of Eq. (35) (Tang and Wang, 2001):

$$V(\boldsymbol{v}) = \tfrac{1}{2}(\boldsymbol{v}^* - \boldsymbol{v})^{\mathrm{T}}(\boldsymbol{v}^* - \boldsymbol{v}) \tag{46}$$

Taking the time derivative of Eq. (46) and using Eq. (35) yields

$$\frac{\mathrm{d}V(\boldsymbol{v})}{\mathrm{d}t} = \frac{\mathrm{d}V(\boldsymbol{v})}{\mathrm{d}\boldsymbol{v}}\left(\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t}\right) = (\boldsymbol{v}^* - \boldsymbol{v})^{\mathrm{T}}\left(\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t}\right) = (\boldsymbol{v}^* - \boldsymbol{v})^{\mathrm{T}}\nabla E(\boldsymbol{v}) \tag{47}$$

By applying Lemma 2 and using the fact that the energy function is positive definite $E(\boldsymbol{v}) \geqslant 0$,

$$(\boldsymbol{v}^* - \boldsymbol{v})^{\mathrm{T}}\nabla E(\boldsymbol{v}) \leqslant -E(\boldsymbol{v}) \leqslant 0 \tag{48}$$

Therefore, the neural network of (35) is Lyapunov stable. According to LaSalle's invariance rule (Slotine and Li, 1991), all trajectories $\boldsymbol{v}(t)$ converge to the largest invariant set in the set $\Omega = \{\boldsymbol{v} \in \mathbb{R}^{5n+m} | \dot{V} = 0\}$.

In view of Eq. (47), $\dot{V} = 0$ implies $\dot{\boldsymbol{v}} = 0$, and this further implies that the set Ω also contains the equilibrium points of (35) i.e., $\Omega = \{\boldsymbol{v} \in \mathbb{R}^{5n+m} | \dot{V} = 0\} = \{\boldsymbol{v} \in \mathbb{R}^{5n+m} | \dot{\boldsymbol{v}} = 0\}$. Thus, Eq. (35) converges to its equilibrium points. According to Lemma 2, $E(\boldsymbol{v}^*) = 0$ if and only if $\nabla E(\boldsymbol{v}^*) = 0$ implying that $\boldsymbol{v}^*$ makes $\dot{\boldsymbol{v}} = 0$. Therefore, the equilibrium points of Eq. (35) correspond to $\boldsymbol{v}^*$.

Since the energy function Eq. (33) is differentiable and convex for all $\boldsymbol{v}$, the local minimum of the energy function corresponds to the global minimum. The neural network given in Eq. (35) is globally stable and will converge to the optimal solutions of the primal problem, Eq. (27), and its dual problem, Eq. (32). This completes the proof. □

References

Antonelli, G., 2003. Underwater Robots: Motion and Force Control of Vehicle–Manipulator Systems. Springer, Berlin.

Antonelli, G., Chiaverini, S., Sarkar, N., West, M., 2001. Adaptive control of and autonomous underwater vehicle: experimental results on ODIN. IEEE Transactions on Control Systems and Technology 9 (5), 756–765.

Antonelli, G., Caccavale, F., Chiaverini, S., Fusco, G., 2003. A novel adaptive control law for underwater vehicles. IEEE Transactions on Control Systems and Technology 11 (2), 109–120.

Antoniou, A., Lu, W.S., 2007. Practical Optimization: Algorithms and Engineering Applications. Springer, Berlin.

Arati, S.D., Walker, I.D., 1997. Minimum effort inverse kinematics for redundant manipulators. IEEE Transactions on Robotics and Automation 13 (5), 767–775.

Bordignon, K.A., 1996. Constrained control allocation for systems with redundant control effectors. Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Conte, G., Serrani, A., 1998. Robust control of a remotely operated underwater vehicle. Automatica 34 (2), 193–198.

Debitetto, P., 1994. Fuzzy logic for depth control of unmanned undersea vehicles. In: Proceedings of the AUV Symposium, Cambridge, MA, USA, pp. 233–241.

Durham, W., 1993. Constraint control allocation. Journal of Guidance, Control and Dynamics 16 (4), 717–725.

Elmali, H., Olgac, N., 1992. Theory and implementation of sliding mode control with perturbation estimation. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 3, Nice, France, pp. 2114–2119.

Firmani, F., Zibil, A., Nokleby, S.B., Podhorodeski, R.P., 2007. Wrench capabilities of planar parallel manipulators—part I: wrench polytopes and performance indices, Robotica, in preparation.

Fossen, T., 1994. Guidance and Control of Ocean Vehicles. Wiley, New York.

Gill, P., Murray, W., Wright, M.H., 1991. Numerical Linear Algebra and Optimization 1. Addison-Wesley, Reading, MA.

Healey, A., Lienard, D., 1993. Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. IEEE Journal of Oceanic Engineering 18 (3), 327–339.

Ishii, K., Ura, T., 2000. An adaptive neural-net controller system for an underwater vehicle. Control Engineering Practice 8, 177–184.

Kato, N., 1995. Applications of fuzzy algorithm to guidance of and control of underwater vehicle. In: Yuh, J. (Ed.), Underwater Robotic Vehicles: Design and Control. TSI Press, Albuquerque.

Kodogiannis, V., 2003. Direct adaptive control of underwater vehicles using neural networks. Journal of Vibration and Control 9, 605–619.

Lin, F.J., Wai, R.J., 2002. Robust control using neural network uncertainty observer for linear induction motor servo drive. IEEE Transactions on Power Electronics 17 (2), 241–254.

Omerdic, E., Roberts, G., 2004. Thruster fault diagnosis and accommodation for open-frame underwater vehicles. Control Engineering Practice 12, 1575–1598.

Ortega, J.M., Rheinboldt, W.G., 1970. Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York.

Pepijn, W.J., Colin van de Ven, F., Daniel, T., 2005. Neural network control of underwater vehicles. Engineering Applications of Artificial Intelligence 18, 533–547.

Popov, V., 1973. Hyperstability of Control Systems. Springer, Berlin.

Sarkar, N., Podder, T.K., Antonelli, G., 2002. Fault-accommodating thruster force allocation of an AUV considering thruster redundancy and saturation. IEEE Transactions on Robotics and Automation 18 (2), 223–231.

Slotine, J.J., Coetsee, J., 1986. Adaptive sliding controller synthesis for non-linear systems. International Journal of Control 43, 1631–1651.

Slotine, J.J., Li, W., 1991. Applied Nonlinear Control. Prentice-Hall, Englewood Cliffs, NJ.

Slotine, J.J., Shastry, S., 1983. Tracking control of nonlinear system using sliding surfaces. International Journal of Control 38 (2), 465–492.

Sordalen, O., 1997. Optimal thrust allocation method for marine vessels. Control Engineering Practice 5 (9), 93–106.

Soylu, S., Buckham, B., Podhorodeski, R., 2007. Robust control of underwater vehicles with fault-tolerant infinity-norm thruster force allocation. In: Proceedings of the OCEANS 2007 MTS/IEEE Conference and Exhibition, Vancouver, BC, Canada.

Steinke, D., 2006. Design and simulation of a Kalman filter for ROV navigation. Master's Thesis, Victoria, BC, Canada.

Tang, W., Wang, J., 2001. A recurrent neural network for minimum infinity-norm kinematic control of redundant manipulators with an improved problem formulation and reduced architecture complexity. IEEE Transactions on Systems, Man, and Cybernetics 31 (1), 98–105.

Van de Ven, P., Flanagan, C., Toal, D., 2005. Neural network control of underwater vehicles. Engineering Applications of Artificial Intelligence 18, 533–547.

Wang, J., 1993. Analysis and design of a recurrent neural network for linear programming. IEEE Transactions on Circuits and Systems I 40 (9), 613–618.

Wolfe, P., 1961. A duality theorem for nonlinear programming. Quarterly of Applied Mathematics 19, 239–244.

Wright, M.H., 1992. Interior methods for constrained optimization. Acta Numerica 1, 341–407.

Xia, Y., 1996. A new neural network for solving linear programming problems and its applications. IEEE Transactions on Neural Networks 7, 525–529.

Xia, Y., Wang, J., 1998. A general methodology for designing globally convergent optimization neural networks. IEEE Transactions on Neural Networks 9 (6), 1331–1343.

Yoerger, D., Slotine, J.J., 1985. Robust trajectory control of underwater vehicles. IEEE Journal of Oceanic Engineering OE-10, 462–470.

Zeinali, M., Notash, L., 2007. Fuzzy logic-based adaptive robust control for parallel manipulators. In: Proceedings of the 12th World Congress in Mechanism and Machine Science, Besançon, France.

Appendix D. Automatic Navigation and Control of the Falcon™ ROV

(Manuscript in Preparation)

AUTOMATIC NAVIGATION AND CONTROL OF THE FALCON™ ROV

Serdar Soylu*, Alison Proctor, Ron P. Podhorodeski, and Bradley J. Buckham
Department of Mechanical Engineering,
University of Victoria, P.O. Box 3055, Victoria, B.C., Canada, V8W 3P6
{ serdar, proctora, bbuckham, podhoro }@uvic.ca
*Corresponding author: Fax: (250) 721-6035

Abstract

The current work addresses the design, implementation and testing of a guidance, navigation and control system for inspection-class remotely operated underwater vehicles (**ROV**), specifically the Saab-Seaeye Falcon™ ROV. For the guidance problem, a Lyapunov based guidance algorithm is implemented that can function satisfactorily in the presence of significant uncertainties in the system kinematics. A new sensor suite, including off-the-shelf sensors of a Systron-Donner™ IMU, Explorer™ DVL, Sparton™ Compass, and SouthStar™ SBL, are developed and integrated into the existing Falcon™ telemetry system. To accurately estimate the Falcon™ motion, an Extended Kalman Filter (**EKF**) is designed that blends the on-board sensor measurements with low-noise and high update rate using a process model. For the process model of the EKF, 4-degrees of freedom (DOF), decoupled, simplified hydrodynamic model of the Falcon™ ROV are obtained through the sea-trials using the on-board sensors. To comprehensively tune the EKF parameters, including the covariance matrix diagonals for the process model as well as the SBL, further wet-tests are performed for reliable navigation. The verification of the navigation algorithm is performed by comparing the system's outputs with those of a high precision optical motion capture system. An average estimation error of 0.15m for

the surge motion and 0.16m for the sway motion is found for the manoeuvre considered. For automatic control, a novel control law is implemented that is composed of a model-based equivalent control signal and two adaptive signals; one is a PID contribution with adaptive gains, and the second is derived from an estimate on the upper bound on the difference between the known and the unknown system dynamics. The experimental validation of the proposed guidance, navigation and control scheme was performed for the way-point and trajectory following tasks via sea-trials. The average RMS error values of 0.08m, 0.04m, 0.04m, and 2.52degree values were recorded along X, Y, Z axis and around Z axis of the North-East-Down reference frame, respectively, during a rectangular trajectory following task. In conclusion, centimetre-level precision is achieved with the automatic control scheme, thereby significantly extending the capabilities of the Falcon™ system for high precision.

1 INTRODUCTION

Remotely Operated Underwater Vehicles (**ROVs**) are widespread in scientific and industrial investigations in the marine environment including bottom surveys, seafloor imaging, oceanographic and geological data acquisition and robotic interactions with natural features or man-made infrastructure. In typical ROV operations, the motion of the ROV is directed by a human pilot stationed on a surface vessel through a tether that provides power and telemetry. The human presence makes complex multi-objective underwater missions possible: human reasoning can manage sudden changes in a mission plan caused by the unpredictable nature of the ocean environment. However, the limitations in sensory feedback to the ROV pilot make certain operations, such as high precision trajectory following, impossible without some form of machine intelligence.

Especially for low-cost inspection ROVs, there is an acute need for automatic correction of erroneous motions that are caused by currents and tether disturbances that are unbeknown to the human pilot.

Traditional linear controllers cannot provide high precision motion control ability, as their performance degrades significantly in the face of highly non-linear ROV hydro dynamics and tether disturbances. More advanced control techniques are needed for reliable, high precision, effective ROV operations in such conditions and several nonlinear control techniques have been applied to ROV control. These include: the H_∞ approach in [1]; adaptive control techniques in [2], [3], [4], and [5]; sliding-mode control in [6], [7], [8], and [9]; fuzzy logic control in [10]; neural network control in [11] and [12]; combined approaches such as H_∞ and sliding-mode control in [13], adaptive, fuzzy, and sliding-mode control in [14] and sliding-mode control and neural network in [15]; and other types of non linear controllers such as in [16] and [17].

In the current, work, a novel controller is implemented on an inspection-class ROV, specifically a Saab-Seaeye Falcon™ ROV, for high precision trajectory and way-point following. The control law is three layered and includes: equivalent control, adaptive control, and PID control signals. The equivalent control signal is a model based component that reflects the knowledge of the system dynamics available to the controller designer. The robust adaptive control signal constantly estimates the upper bound on a lumped uncertainty vector, which is defined as the difference between the known and the unknown dynamics. This term provides stability to the controller with its ability to track the upper bound on the lumped uncertainty matrix. Finally, the PID control layer, with adaptive PID feedback gains, is used to further enhance the controller performance as

well as to tune the controller behaviour in particular when the lumped uncertainty is small.

In the implementation phase of this controller, a two layered hierarchical architecture proposed in [18] is employed. In this architecture, the outer guidance loop algorithm receives target way-point locations and generates motion directives based on the ROV kinematics that are fed into the inner control loop. The controller then commands the on-board thrusters to realize the motion directives based on the state feedback that comes from the EKF. For the guidance algorithm, a Lyapunov based guidance algorithm originally presented in [18] is implemented. As noted in [16], this guidance method has the ability to eliminate operational errors without any off-line or on-line trajectory replanning and hence is suitable for ROV control in the presence of high uncertainties. With regards to the EKF, the methodology presented in [19] was implemented to blend the measurements of short base line (**SBL**) acoustic positioning system, a DVL, a low-cost IMU, a compass, and a depth sensor to provide the controller with low noise, high update rate and accurate state feedback. The EFK state estimation is also complemented with the dynamic model of the ROV extracted from a number of sea-trials to further enhance the state estimation as well as to provide robustness to the EKF in the case of sensor drop-outs.

The EKF is fed a series of motion signals produced by a custom sensor pod attached to the FalconTM ROV. The pod includes a Systron-Donner™ IMU, Explorer™ DVL, Sparton™ Compass, and SouthStar™ SBL and was integrated into the existing Falcon™ telemetry system. The SBL system has its own dedicated cable that runs through the umbilical cable, allowing faster transponder pinging rates. In order to evaluate the

accuracy of the EKF state estimation and the control strategy, a high precision optical motion capture system is utilized. The optical system provides the true states to which the EKF estimation is compared allowing a direct evaluation of system performance rather than an inferred one based on error bounds produced in the EKF calculations. In addition, system performance is tested through a series of sea-trials based on trajectory following and way point tasks. The results indicate the effectiveness of the proposed guidance, navigation and control scheme in obtaining high precision ROV motion.

The remainder of the current work proceeds as follows: Section 2 provides the mathematical background used in the current work. The topics covered in Section 2 are the kinematics and dynamics of the Falcon™ ROV, navigation, guidance and finally control. Section 3 introduces the wet-test facility built for the current research and provides the brief definition of the Falcon™ experimental platform. This section also gives a brief description of the underwater sensor suite available to the current study as well as the navigation-pod particularly designed for the current study. A small discussion about the integration of the navigation-pod into the existing Falcon™ telemetry is also provided in this section. Section 4 outlines the steps taken to obtain the hydrodynamics of the Falcon™ ROV based on the field data and reports the results. In addition, the verification of the navigation algorithm via sea-trials is also provided in this section. Finally, Section 5 demonstrates the controller performance in the way-point and trajectory following tasks through a number of wet tests.

2 THEORY

2.1 Kinematics of Falcon™ ROV

To express the motion of the ROV, two reference frames are used. The first is an inertial reference frame that is coincident with the North-East-Down (**NED**) frame, and a body-fixed frame affixed to the vehicle center of mass as shown in Figure 1. The ROV's state vector with respect to (**wrt**) its body-fixed frame is defined as $\boldsymbol{q} = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T$ and the vehicle state wrt the inertial frame is given by $\boldsymbol{\eta} = \begin{bmatrix} X & Y & Z & \phi & \theta & \psi \end{bmatrix}^T$. The spatial transformation matrix between the inertial frame and ROV's body-fixed frame is given by **J**, which includes the angular velocity transformation matrix $\mathbf{J}_2$ and the linear velocity transformation matrix $\mathbf{J}_1$. The term **J** can be obtained by the Euler sequence of rotations: the first rotation ψ is about the *Z* axis of the inertial frame (yaw), the second rotation θ is about the new *Y* axis (pitch), and finally, the last rotation ϕ is about the new *X* axis (roll);

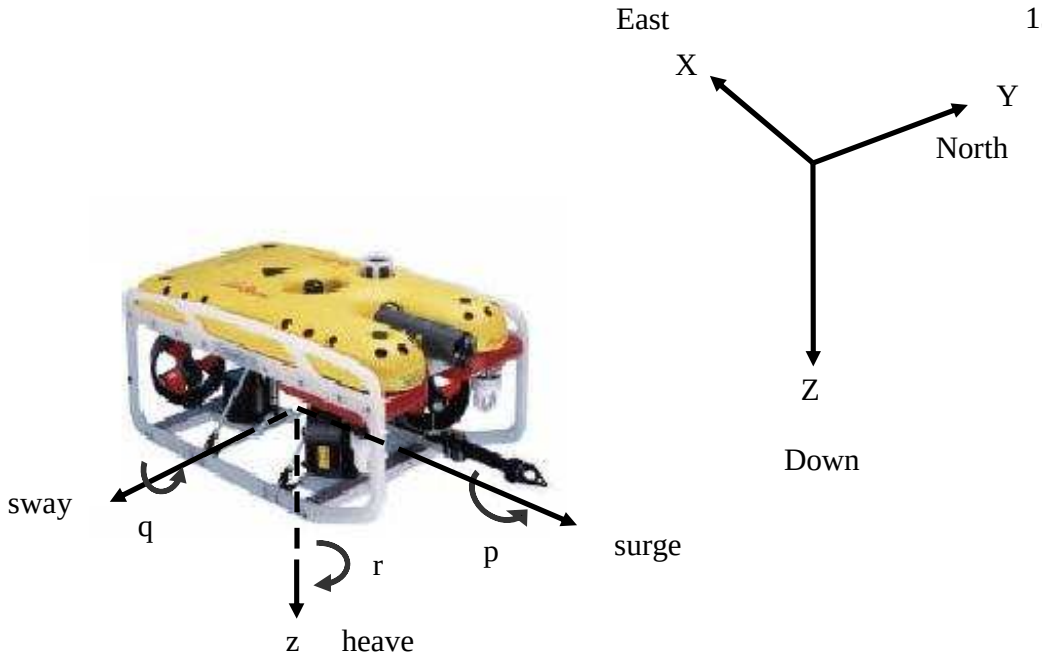


Figure 1: Coordinate frames. Image courtesy of http://seaeye.com.

$$\dot{\boldsymbol{\eta}} = \mathbf{J}\boldsymbol{q}, \quad \mathbf{J} = \begin{bmatrix} \mathbf{J}_2 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{J}_1 \end{bmatrix},$$

$$\mathbf{J}_2 = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi + \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}, \tag{1}$$

$$\mathbf{J}_1 = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix}.$$

In the current study, the roll and pitch motion of the Falcon™ cannot be actively controlled since the thruster layout of does not create pitching and rolling moments. However, the motion around these axes is self-regulated due to large buoyant restoring moments. Assuming that the pitch and roll motions are small, the local and inertial state vectors can be simplified to $\boldsymbol{q} = \begin{bmatrix} u & v & w & r \end{bmatrix}^T$ and $\boldsymbol{\eta} = \begin{bmatrix} X & Y & Z & \psi \end{bmatrix}^T$, respectively. Applying $\phi = \theta = 0$ in $\mathbf{J}$ of Eq. (1) and eliminating the rows and columns associated with

the roll and pitch motion leaves a simplified transformation between the two representations of the vehicle state:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}\boldsymbol{q}$$

$$\mathbf{J} = \begin{bmatrix} \cos\psi & \cos\psi - \sin\psi & 0 & 0 \\ \sin\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Any further mention to $\boldsymbol{q}$ and $\mathbf{J}$ is specific to the reduced entities in Eq. (2).

2.2 Dynamic Modelling of Falcon™ ROV

Applying the simplified kinematic model of the ROV given in Eq. (2), the ROV dynamics model given in [20] can be reduced to:

$$(\mathbf{M} + \mathbf{M_A})\dot{\boldsymbol{q}} + \mathbf{C}\boldsymbol{q} + \mathbf{D}\boldsymbol{q} + \mathbf{g} = \boldsymbol{\tau} \tag{3}$$

The term $\mathbf{M} \in \mathrm{R}^{4\times4}$ and $\mathbf{M_A} \in \mathrm{R}^{4\times4}$ are the inertia and added mass matrix, respectively, $\mathbf{C} \in \mathrm{R}^{4\times4}$ is the matrix of Coriolis terms, $\mathbf{D} \in \mathrm{R}^{4\times4}$ is the drag matrix, $\mathbf{g} \in \mathrm{R}^4$ is the vector of gravity and buoyancy forces and moments, and finally $\boldsymbol{\tau} \in \mathrm{R}^4$ is the control forces and moments acting on the ROV centre of mass.

Mass and Added Mass: For the reduced dynamics model considered, the mass matrix is defined as:

$$\mathbf{M} = diag(m \quad m \quad m \quad I_r) \tag{4}$$

where m is the true vehicle mass and I_r is the inertia around the Z axis (yaw) of the vehicle. The added mass effect, which characterizes the additional resistance of the surrounding water to the accelerating body, is defined as:

$$\mathbf{M_A} = diag(x_{\dot{u}} \quad y_{\dot{v}} \quad z_{\dot{w}} \quad I_{\dot{r}}) \tag{5}$$

where $x_{\dot{u}}$, $y_{\dot{v}}$ and $z_{\dot{w}}$ defines the added mass effect along the *x*, *y* and *z* axis, respectively, and $I_{\dot{r}}$ defines an added inertia in the yaw degree of freedom caused by rotation of the vehicle around the yaw axis. Given the ROV's structural symmetry and the low operating speeds, the added mass matrix is assumed to be diagonal.

Coriolis: The Coriolis matrix is defined for the 4-DOF reduced dynamics as:

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & -mv \\ 0 & 0 & 0 & mu \\ 0 & 0 & 0 & 0 \\ mv & -mu & 0 & 0 \end{bmatrix} \tag{6}$$

With this definition, it can be shown that $\mathbf{C} = -\mathbf{C}^T$; a property that allows simplifications in the control law derivations.

Drag Forces: A blend of linear and quadratic drag terms are applied according to [20]. The drag matrix **D** is defined as:

$$\mathbf{D} = diag\left(k_u + k_{u|u|}|u| \quad k_v + k_{v|v|}|v| \quad k_w + k_{w|w|}|w| \quad k_r + k_{r|r|}|r|\right) \tag{7}$$

where k_i and $k_{|i|}, i = u, v, w, r$ are the linear and quadratic drag coefficients associated with the surge, sway, heave, and yaw motion of the vehicle, respectively. Note that off-diagonal terms of the drag matrices are neglected as the hydrodynamic coupling is assumed to be insignificant at the low speeds of the inspection class ROV.

Buoyancy Forces: The Falcon™ is trimmed to be neutrally buoyant, and its roll and pitch motions are neglected. As such, the vector of gravity and buoyancy forces g is assumed to be zero in our implementation.

Representation in terms of the Inertial Reference Frame: Equation (3) can be represented in the inertial reference frame as [20]:

$$\boldsymbol{f} = \mathbf{M}_{\eta}(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \mathbf{C}_{\eta}(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \mathbf{D}_{\eta}(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \boldsymbol{g}_{\eta}(\boldsymbol{\eta}) = \mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau} \tag{8}$$

where $\mathbf{M}_{\eta}(\boldsymbol{\eta}) = \mathbf{J}^{-\mathrm{T}}\mathbf{M}\mathbf{J}^{-1}$, $\mathbf{C}_{\eta}(\boldsymbol{q},\boldsymbol{\eta}) = \mathbf{J}^{-\mathrm{T}}\left[\mathbf{C} - \mathbf{M}\mathbf{J}^{-1}\dot{\mathbf{J}}\right]\mathbf{J}^{-1}$, $\mathbf{D}_{\eta}(\boldsymbol{q},\boldsymbol{\eta}) = \mathbf{J}^{-\mathrm{T}}\mathbf{D}\mathbf{J}^{-1}$ and $\boldsymbol{g}(\boldsymbol{\eta}) = \mathbf{J}^{-\mathrm{T}}\boldsymbol{g}$. The dynamics defined by Eq. (8) have the following structural properties [20]:

Property 1: The inertia matrix $\mathbf{M}_{\eta}$ is symmetric and positive definite, i.e., $\mathbf{M}_{\eta}^{\mathrm{T}} = \mathbf{M}_{\eta}$;

Property 2: Matrix $\dot{\mathbf{M}}_{\eta} - 2\mathbf{C}_{\eta}$ is skew symmetric, i.e., for any vector $\boldsymbol{\zeta}$, $\boldsymbol{\zeta}^{\mathrm{T}}(\dot{\mathbf{M}}_{\eta} - 2\mathbf{C}_{\eta})\boldsymbol{\zeta} = 0$.

Lumped Uncertainty: Given the existence of inaccuracies in the system dynamics mode, Eq. (8) can be written as the sum of estimated dynamics $\hat{\boldsymbol{f}}$ and the unknown dynamics $\tilde{\boldsymbol{f}}$,

$$\boldsymbol{f} = \hat{\boldsymbol{f}} + \tilde{\boldsymbol{f}} \tag{9}$$

where the estimated dynamics vector is defined as:

$$\hat{\boldsymbol{f}} = \hat{\mathbf{M}}_{\eta}(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \hat{\boldsymbol{h}}(\boldsymbol{q},\boldsymbol{\eta}) \tag{10}$$

with $\hat{\boldsymbol{h}}_\eta(\boldsymbol{q},\boldsymbol{\eta}) = \hat{\mathbf{C}}_\eta(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \hat{\mathbf{D}}_\eta(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \hat{\boldsymbol{g}}_\eta(\boldsymbol{\eta})$ and the unknown dynamics vector defined as:

$$\tilde{\boldsymbol{f}} = \tilde{\mathbf{M}}_\eta(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \tilde{\boldsymbol{h}}_\eta(\boldsymbol{q},\boldsymbol{\eta}) \tag{11}$$

with $\tilde{\boldsymbol{h}}_\eta(\dot{\boldsymbol{q}},\boldsymbol{\eta}) = \tilde{\mathbf{C}}_\eta(\dot{\boldsymbol{q}},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \tilde{\mathbf{D}}_\eta(\dot{\boldsymbol{q}},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \tilde{\boldsymbol{g}}_\eta(\boldsymbol{\eta})$ where

$$\begin{aligned} \tilde{\mathbf{M}}_\eta &= \mathbf{M}_\eta - \hat{\mathbf{M}}_\eta,\ \tilde{\mathbf{C}}_\eta = \mathbf{C}_\eta - \hat{\mathbf{C}}_\eta, \\ \tilde{\mathbf{D}}_\eta &= \mathbf{D}_\eta - \hat{\mathbf{D}}_\eta,\quad \tilde{\mathbf{g}}_\eta = \mathbf{g}_\eta - \hat{\mathbf{g}}_\eta \end{aligned} \tag{12}$$

The unknown dynamics vector is also called the lumped uncertainty vector and is proposed for use in adaptive control in [21].

2.3 Navigation: Extended Kalman Filter

The measurements coming from the on-board sensors for the position tracking of the ROV contain both the true states of the ROV and noise, and a Kalman Filter (KF), as discussed in [22], can be used to isolate the true state in the corrupted signal. The KF provides a computationally efficient and recursive method to estimate the state of a process model through the minimization of the estimation of the error variance. During the state estimation, the KF projects the behaviour of the process model and disregards the components of the measurement signal that disagrees with the predicted behaviour past an allowable margin. To define that margin, the KF uses statistical knowledge pertinent to the available sensors as well as the dynamics model of the process model.

However, the standard KF uses a linear representation of the process model and measurement functions that map the sensor data to the KF's observed states, and hence is not directly applicable for estimating nonlinear ROV motion. To overcome this issue, a

nonlinear version of the KF, the Extended Kalman Filter (**EKF**) [23] was implemented. The EKF linearizes the non-linear process models and the measurement functions about the current state estimate at each iteration; making it much more suitable for observing non-linear ROV motions. However, unlike a KF estimate, the EKF estimate is not guaranteed to provide a minimum error variance estimate since the nonlinear process dynamics and measurement functions violate the zero-mean Gaussian noise requirement. Nevertheless, near optimality can still be obtained through the EKF, provided that proper estimations of the process uncertainty are made.

The following EKF state vector is used in the current work:

$$\boldsymbol{x} = \begin{bmatrix} u & v & w & p & q & r & X & Y & Z & \phi & \theta & \psi \end{bmatrix} \tag{13}$$

where $\begin{bmatrix} u & v & w \end{bmatrix}$ and $\begin{bmatrix} p & q & r \end{bmatrix}$ are the body-fixed translational and rotational velocities, respectively, whereas $\begin{bmatrix} X & Y & Z \end{bmatrix}$ and $\begin{bmatrix} \phi & \theta & \psi \end{bmatrix}$ are the position and Euler angles of the vehicle for the orientation in terms of the NED frame as depicted in Figure 1. Note that although the controller does not use the rotational states around the local *y* and z axes, they were recorded for the purpose of validating the "close to zero" assumptions made for the ROV rotations about those axes.

The complete EKF algorithm is demonstrated in Figure 2 based on the works of [19] and [24]. In the initialization state, the initial rough estimate of $\hat{\boldsymbol{x}}_0$ and the corresponding initial state error covariance matrix $\hat{\mathbf{P}}_0$ is used whose diagonal elements corresponds to high estimated error variances. The overhead hat $(\wedge)$ denotes an estimated value.

In the *Prediction* phase, the initial estimate of $\hat{\boldsymbol{x}}_k^-$ and $\hat{\mathbf{P}}_k^-$ at time step k are calculated based on the following equations:

$$\hat{\boldsymbol{x}}_k^- = \hat{\boldsymbol{x}}_{k-1}^+ + \int_{t_{k-1}}^{t_k} f\left(\hat{\boldsymbol{x}}_{k-1}^+,\ t\right) d\tau \tag{14}$$

$$\hat{\mathbf{P}}_k^- = \hat{\mathbf{P}}_{k-1}^+ + \int_{t_{k-1}}^{t_k} (\mathbf{FP} + \mathbf{PF}^T + \dot{\mathbf{Q}}) d\tau \tag{15}$$

Where the superscript $(\cdot)^T$ denotes transpose, and the superscripts $(\cdot)^-$ and $(\cdot)^+$ denote *a piori* and *a postpriori*, respectively. While *a piori* state values are obtained through advancing the dynamic model of the EKF states forward in time through a chosen integration scheme, *postpriori* state values are obtained through incorporating new measurements into the *priori* estimates to obtain improved *posteriori* estimates. In Eq. (14), $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, t)$ is the time rate of change of the Kalman states. In Eq.(15), $\mathbf{F} = \left.\dfrac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=\hat{x}_k^-}$, is a Jacobian matrix that defines the contributions of each state variable to the entries of $\dot{\boldsymbol{x}}$, and finally, $\dot{\mathbf{Q}}$ is the time rate of change of Kalman Filter process model error covariance matrix $\mathbf{Q}$.

In the *Measurement* phase, the error covariance matrix ${}^m\mathbf{R}_k = diag({}^m\sigma_{k,1},\ {}^m\sigma_{k,2},\ \ldots\ {}^m\sigma_{k,n})$ is formed at step *k*, where ${}^m\sigma_k$ is the error variance of the m^{th} Kalman state and *n* is the number of measurement made during the time step $d\tau$. Similarly, the measurement matrix ${}^m\mathbf{H}_k = \left[{}^m\mathbf{h}_{k,1};\ {}^m\mathbf{h}_{k,2}\ \ldots;\ {}^m\mathbf{h}_{k,n}\right]$ are formed by concatenating ${}^m\mathbf{h}_{k,1}$ row vectors corresponding to m^{th} Kalman state. The term

${}^{m}\mathbf{h}_{k,1}$ is defined as ${}^{m}\boldsymbol{h}_k = \left.\frac{\partial {}^{m}h(\boldsymbol{x})^T}{\partial \boldsymbol{x}}\right|_{x=\hat{x}_k^-}$ where ${}^{m}h(\boldsymbol{x})$ is the scalar measurement function that maps each Kalman state to measured states, ${}^{m}z$. Finally, the measurement innovation vector ${}^{m}\boldsymbol{e}_k = \left[{}^{m}e_{k,1},\ {}^{m}e_{k,2},\ \ldots,\ {}^{m}e_{k,n}\right]^T$ is formed for each Kalman state using measurements received during the time step $d\tau$ according to ${}^{m}e_k = {}^{m}z_k - {}^{m}h\left({}^{m}\hat{\boldsymbol{x}}^-\right)$.

In the *Collect* phase, ${}^{m}\mathbf{H}_k$ and ${}^{m}\mathbf{R}_k$ are assembled to form the final $\mathbf{H}_k$ and $\mathbf{R}_k$. This step is followed by the *Compute Gain* phase in which Kalman gain matrix are computed according to $\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left[\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k\right]^{-1}$. Finally, in the *Blend* phase, *postpriori* state values $\hat{\boldsymbol{x}}_k^+$ are calculated by correcting the model based initial estimate with the innovation vector according to $\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \mathbf{K}_k \mathbf{e}_k$. Additionally, the error covariance matrix is also computed in this phase based on $\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k]\mathbf{P}_k^-$. The refined estimate $\hat{\boldsymbol{x}}_k^+$ is then used as the initial estimate for the *Prediction* state at step *k+1* and same procedure is followed recursively. For more details, the reader is referred to [19].

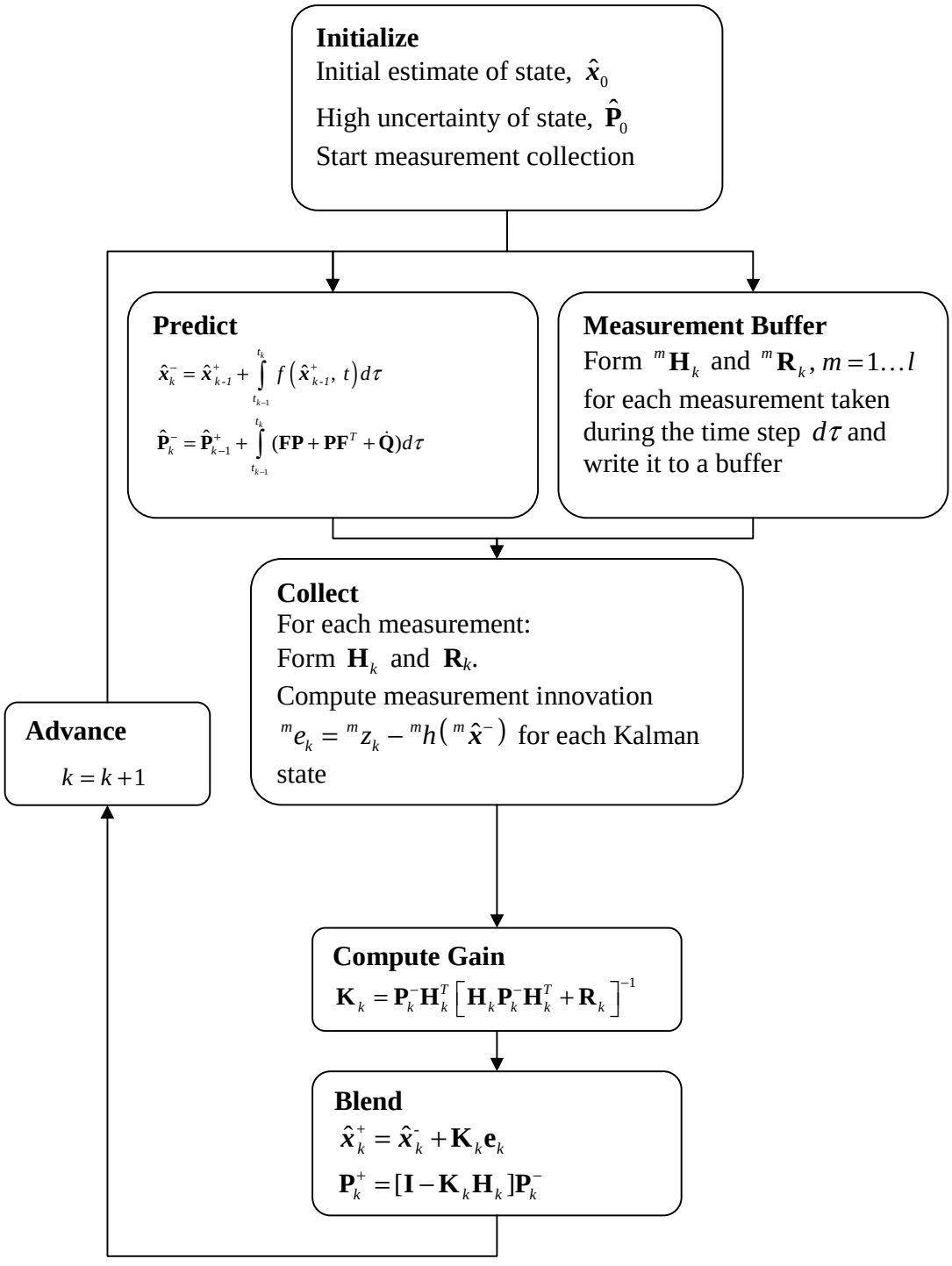


Figure 2: Flowchart of the EKF estimation algorithm.

2.4 Guidance

For the synthesis of the guidance equations, the Lyapunov stability theory presented in [18] was implemented in the current work. The same approach was also implemented to an ROV in [16]. To utilize this approach, the following Lyapunov function candidate was considered:

$$V = \frac{1}{2}e^2 \tag{16}$$

where $e = \zeta - \zeta_d$ is the error (defined as the "task function" in [18]) with ζ and ζ_d being the position or orientation error of interest. Defining the reference velocities as

$$\dot{\eta}_d = \dot{e} = -\lambda_p e \tag{17}$$

with $\lambda_p > 0$ yields a negative-definite $\dot{V}$, and hence, $e \to 0$ as $t \to \infty$, according to the Lyapunov theory.

As noted in [16], high λ gains in the proportional term can cause instability and limit–cycling problems. In order to overcome this potential problem, an additional accumulative contribution can be added to the reference velocities as follows:

$$\dot{\eta}_d = \dot{e} = -\lambda_p e - \lambda_I \int_0^t e d\tau \tag{18}$$

with $\lambda_p = \varepsilon + \gamma$, $\lambda_I = \varepsilon\gamma$, $\varepsilon > 0$ and $\gamma > 0$. In additon, to minimize the wind-up effect induced by the integral term, a hysteresis mechanism can be included. With this mechanism, the integral action is activated only when $|e|$ is smaller than predefined threshold values, $|e| \le \sigma$. This addition also ensures that the integral action does not lead

to potential instability. Otherwise, this would have been in question since the integral action could make the time derivative of the Lyapunov function positive. As well, the output of proportional and integral term in Eq. (18) can be bounded to avoid excessive velocities such that $\left|\lambda_p e\right| \le \xi_p$ and $\left|\lambda_I \int_0^t e d\tau\right| \le \xi_I$.

The types of automated motion tasks considered in this work are: *Depth and Yaw* and *Horizontal Motion*. In conventional *Depth and Yaw*, only the vehicle's heading and depth are guided. To this end, the guidance algorithm generates reference velocities for a given desired heading ψ_d, and depth, Z_d values defined with respect to the NED frame. The reference velocities $\dot{Z}$ and $\dot{\psi}$ can then be found by substituting $e_z = Z - Z_d$ and $e_\psi = \psi - \psi_d$ in Eq. (18), respectively. For *Horizontal Motion*, the guidance algorithm generates reference velocities for a given desired heading X_d, and depth, Y_d values defined with respect to the NED frame. The reference velocities $\dot{X}$ and $\dot{Y}$ can then be found by substituting $e_X = X - X_d$ and $e_Y = Y - Y_d$ in Eq. (18), respectively. The desired position and acceleration values can be found by $\eta_d = \int \dot{\eta}_d, \ddot{\eta}_d = d\dot{\eta}_d / dt$, respectively.

2.5 Control

For the control problem of the Falcon™ vehicle, the controller proposed in [25] was adopted due to its adaptability and robustness. Below is a brief summary of this controller.

For a second order system such as an underwater vehicle system, reasonable desired dynamics would be a stable first order system. This first order dynamics can be defined as:

$$\boldsymbol{s} = \dot{\boldsymbol{e}} + \boldsymbol{\Lambda e} \tag{19}$$

with the error vector being $\boldsymbol{e} = \boldsymbol{q} - \boldsymbol{q}_d$ and $\boldsymbol{\Lambda} \in \mathbf{R}^{n \times n}$ being a constant, symmetric, positive definite and diagonal gain matrix. In the sliding-mode theory, (19) characterizes the desired dynamics and represents a time-varying line in the state space that passes through the desired state variables [26]. The control signal to be designed works to keep the value of $\boldsymbol{s}$ at zero. When $\boldsymbol{s} = \mathbf{0}$, the system is adhering to the prescribed dynamic behaviour.

The following control law is proposed for the control problem posed by the Falcon™ ROV:

$$\boldsymbol{\tau} = \mathbf{J}^{\mathrm{T}} \left(\hat{\boldsymbol{f}}_r(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_r) - \mathbf{K}\boldsymbol{r} - \hat{\rho} sat(\boldsymbol{s}) - \hat{\mathbf{C}}_q \boldsymbol{s} \right) \tag{20}$$

where $\mathbf{K} = \begin{bmatrix} \mathbf{K_D} & \mathbf{K_P} & \mathbf{K_I} \end{bmatrix} \in \mathbf{R}^{n \times 3n}$ is a gain matrix with $\mathbf{K}_{\mathrm{D}} \in \mathbf{R}^{\mathrm{n \times n}}$, $\mathbf{K}_{\mathrm{P}} \in \mathbf{R}^{\mathrm{n \times n}}$, $\mathbf{K}_{\mathrm{I}} \in \mathbf{R}^{\mathrm{n \times n}}$ whose elements are the derivative, proportional and integral constants (PID gains), respectively. The term $\boldsymbol{r} = [\dot{\boldsymbol{e}}^{\mathrm{T}} \quad \boldsymbol{e}^{\mathrm{T}} \quad \int \boldsymbol{e}^{\mathrm{T}}]^T \in \mathbf{R}^{3n \times 1}$ is the error term and $\hat{\rho} \in \mathbf{R}^{n \times 1}$ is the estimated upper bound on the lumped uncertainty vector defined in (11) unknown system dynamics. The $sat(\cdot)$ function is defined as:

$$\begin{aligned} sat(s_i/\Phi_i) &= s_i/\Phi_i \qquad \text{if } |s_i/\Phi_i| \le 1 \\ sat(s_i/\Phi_i) &= sign(s_i/\Phi_i) \quad otherwise \end{aligned} \tag{21}$$

where *sign*(.) function is defined as $sign(s_i/\Phi_i) = -1$ for $s_i/\Phi_i \le 0$ and $sign(s_i/\Phi_i) = 1$ for $s_i/\Phi_i > 0$, $i = 1 \ldots n$ [26].

Theorem 1: Consider the nonlinear dynamical system described by (8) . If the control law is expressed as (20) with the following adaptation laws,

$$\begin{aligned} \dot{\mathbf{K}}^{\mathrm{T}} &= \boldsymbol{\beta}\left(\Delta\left(\boldsymbol{r}\boldsymbol{s}^{\mathrm{T}}\right) + \kappa_1\left(\mathbf{K}_0 - \mathbf{K}\right)^{\mathrm{T}}\right), \\ \dot{\hat{\rho}} &= \upsilon\left(\|\boldsymbol{s}\|_2 + \kappa_2\left(\rho_0 - \hat{\rho}\right)\right) \end{aligned} \tag{22}$$

Then the stability of the closed-loop control system is guaranteed in Lyapunov sense with the condition of $\left\|\tilde{\boldsymbol{f}}\right\|_2 \le \rho$.

In Eq. (22), the term $\boldsymbol{\beta} \in \mathbf{R}^{3n\times 3n}$ is a positive definite diagonal matrix, $\kappa_1 > 0$ and $\kappa_2 > 0$ user-defined parameters that determine the degree of trade-off between the tracking performance and the robustness of the control signal to the parameter drift problem, $\mathbf{K}_0$ and ρ_0 are the initial PID gain and upper bound constants, respectively, $\upsilon > 0$ is the upper bound estimation gain, and finally Δ is the operator that allows only the diagonal elements of $\mathbf{K}$ to adapt.

Proof: The Lyapunov stability proof is given in [25] with $V = (1/2)\left(\boldsymbol{s}^{\mathrm{T}}\mathbf{M}_q\mathbf{s} + tr(\mathbf{K}\boldsymbol{\beta}^{-1}\mathbf{K}^{\mathrm{T}}) + \upsilon^{-1}\tilde{\rho}^2\right)$ being a Lyapunov candidate function. In the Lyapunov function, $tr(\cdot)$ is the trace of a matrix and $\tilde{\rho} = \rho - \hat{\rho} \in \mathbf{R}^{1\times 1}$ is the difference between the actual and estimated upper bound on the lumped uncertainty vector.

Note that although the proof in [25] is for ROVM systems, it is still applicable to the ROV problem considered here since both systems have similar matrix elements with the same matrix features that are exploited in the Lyapunov stability proof.

3 EXPERIMENTAL SET-UP

3.1 Underwater Research Facility

Over the course of the control project, a new wet test facility was successfully constructed at Van Isle Marina in Sidney BC as shown in Figure 3 and Figure 4. The construction of the lab included a new working deck, walkable moving platform for easy access to the ROV while in the water, a small office/pilot's room, a launching platform for easy deployment and recovery of the Falcon™ ROV, a set of depth adjustable support mounts for the SBL reference stations mounted on the four corners of the boathouse and two camera supports structures fixed to the side walls of the boathouse. In addition, a plastic wrapping was used to cut down on light since the daylight interferes with the light sensitive motion capture systems used in the current work. The same plastic wrapping was used to enclose the entryway of the boathouse across the working deck for reducing the light interface.

Figure 3: Wet-Test facility entrance.

Figure 4: Wet-Test facility exit.

3.2 Experimental ROV Platform

The experimental ROV used in this experimental work uses an inspection class, Saab-Seaeye FALCON™ ROV provided by Suboceanic Sciences Ltd. The system main components are the vehicle, tether, and surface station as shown in Figure 5. The vehicle is designed to dive up to 300m deep and has 75kg mass. The Falcon™ ROV has 5 electric thrusters; the first four thrusters are for the vehicle's horizontal plane motion, the

last one is for the vertical motion in the water column. In terms of the on-board navigational components, the system is equipped with compass for heading measurement, rate gyro for pitch and roll angles, and finally a pressure sensor for depth measurement. In addition, the system has passive sonar that is used to detect nearby objects in the vehicle's horizontal plane.

The Falcon™ ROV employed in this work has a 350m long fiber-optic tether that is used to transmit power and bi-directional telemetry between the surface station and the vehicle. To communicate with each on-board device, the Falcon™ system implements RS485 multi-drop network in which each node contains a micro controller that receives/sends commands/replies from/to the master node over the network.

Finally, the surface station contains power, surface control unit (**SCU**) and monitoring devices for the pilot to operate the vehicle safely from a surface vessel. The power unit transmits 2KW power to down the vehicle. The SCU contains a PC-104 stack by which the master node is programmed. The master node receives commands from the HCU and transmits the commands to the targeted nodes one at a time over the RS485 network and receives replies from the corresponding nodes. The HCU allows the pilot to control the on board thrusters, lights and the tilt motor of the on-board camera through the surface control. The sonar display indicates the nearby objects and finally the video screen provides visual information to the pilot as well as the video overlay indicating the vehicle's orientations.

Figure 5: SeaEye Falcon™ ROV system components. Image courtesy of [19].

3.3 Navigation Pod Design

The original Falcon™ ROV has only depth sensor and a compass, and hence, is not suitable for tasks that require accurate position knowledge such as advanced position control applications. Therefore, an instrument suite that provides an accurate position tracking is acute for the Falcon™ ROV control task intended in the current work. To obtain such a position tracking system, a new sensor suite, including off-the-shelf sensors of a Systron-Donner™ IMU, Explorer™ DVL, Sparton™ Compass, and SouthStar™ SBL was designed. The DVL and IMU are used to obtain vehicle translational and rotational velocities in the vehicle's body fixed frames, respectively. The compass is used to measure the heading of the vehicle in the **NED** reference frame, and finally the SBL system is used to obtain the position of the ROV with respect to a user-defined reference frame.

A series of subsea sensor pods for each sensor along with a navigation skid were designed in collaboration with Dynamic System Analysis Ltd. as shown in Figure 6. With the modular design, DVL, IMU, Compass and SBL transducer have their own housing; allowing the location optimization of each sensor on the new navigation skid in order to minimize magnetic interferences mostly due to the onboard thrusters.

Figure 6: Pressure housings.

The navigation skid was designed to host the sensor housing components as shown in Figure 7 (left). This design provides mounting surface not only for the sensor housings but also for addition tools that might be needed for the Falcon™ system in the future. As well, this design can be used for *ROV* and *ROVM* configurations with its original mounting adapters designed for this purpose.

Radial O-rings were used in each sensor housing unit, as they are easy to assemble and stay in their groove during the dismantling process. As well, each unit has its own vent-plug for easy disassembly especially after a deep water mission and its own dedicated wetable Subconn™ underwater cable connected to the main housing. Each cable carries power and telemetry to the corresponding sensor unit.

A main telemetry and power circuitry as shown in Figure 7 (right) was designed and installed in the main housing that also holds the DVL and its electronic box. The circuitry contains a series of DC to DC converters that are used to convert Falcon™'s 48Volt into the appropriate voltage rating of each sensor. In addition, this circuitry unit contains a voltage regulator for the DVL and fuses, diodes and LED's for effective circuit protection and troubleshooting. On the telemetry side, this circuitry routes the telemetry traffic coming from the main Junction Box of the Falcon™ ROV in the form of RS232 signals to its intended sensor unit.

3.3.1 SouthStar™ SBL

The SouthStar™ SBL system tracks the position of an ROV fitted with a target transponder via acoustic interrogation. To this end, the target transponder emits an acoustic pulse that is received by four fixed reference stations. The SBL software is then calculated the distances (range) values between the moving target transducer and the fixed reference stations based on the travel time of the acoustic signal using the speed of sound in the water.

The SouthStar™ SBL system has its own tether from its surface station to the target transponder to provide higher update rates. This is not desired since the extra tether line

creates additional disturbance on the vehicle attitude, which makes it more difficult to realize the control task at hand. In order to eliminate this extra tether line and its disturbance effect, the SBL signals were rerouted through the fiber-optic cable, instead. In the new transmission scheme, SBL's RS485 telemetry signal issued by the SBL's dedicated surface station was routed through the fiber-optic cable and sent through the main circuitry to the target transducer. The TTL signal -the other telemetry signal issued by the surface station for the synchronization among the SBL units, was first converted into RS232 signal on the surface side and then transmitted through the same fiber-optic cable. This RS232 signals is converted back into a TTL signal with a microchip installed on the main circuitry before being sent to the target transducer mounted on the Falcon™ ROV.

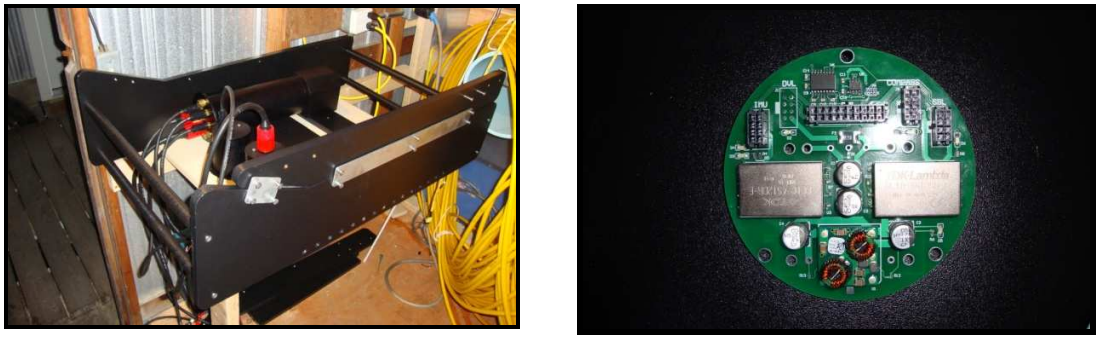


Figure 7: Navigation skid (Left). Telemetry PCB (Right).

3.3.2 Sensor Telemetry for Control

On the vehicle, the Falcon's standard depth, compass and thruster are connected to the Falcon's telemetry through their dedicated RS485 lines that are connected to the fiber-optic multiplexer residing in the vehicle's Junction Box. The external DVL, IMU, and Compass are enclosed in their respective waterproof pressure housings and use RS232

communication protocol for their telemetry. As for the SBL target transducer, it uses RS485 communication protocol for its telemetry and TTL signal for its sync signal. As mentioned before, this sync signal is converted into the RS232 signal in the main housing in order to be interfaced with the existing fiber-optic multiplexer. Like the standard Falcon's components, the RS232 and RS4845 telemetry signals are connected to the same fiber-optic multiplexer that converts the corresponding RS485-RS232 telemetry signals into fiber-optic telemetry signal. This signal is then sent up through the fiber-optic umbilical that is connected to the surface station.

At the surface, the fiber-optic signals are demultiplexed back into RS232 signals for the DVL, IMU, Compass and SBL sync signal and RS485 signals for the Falcon and SBL telemetry. The SBL's RS232 sync signal is converted back into a TTL signal with an appropriate converter and is then sent to the SBL's surface station along with its RS485 telemetry signal. The SBL surface station sends its output to a separate PC that processes the information with the SouthStar™ software. This program streams its processed data to the serial port of the same PC that is connected to the PXI 6251.

The PXI-6251 runs the LabVIEW code in real time that was developed for sensor drivers, guidance algorithm and the controller. In addition, an entirely new telemetry scheme was developed in LabVIEW since the original telemetry communication was at a rate of roughly 4Hz - extremely slow for an automatic control system. The new scheme completely bypasses the master node and is able to communicate with the thrusters at roughly 20 Hz with its own original programming structure. Although higher rates are possible with the proposed master node architecture, 20 Hz update rate was found to be sufficient for the control codes given the slow dynamics of the ROV.

As demonstrated in Figure 8, signals at each RS232 channel received by the PXI 6251 were parsed, processed and times-stamped by their dedicated threads within the Kalman software and stored in the measurement buffer until it is called by the Kalman estimation algorithm that reads the measurement buffer and comes up with the state estimation based on the algorithm summarized in Section 2.3. The controller uses this estimation along with the desired vehicle attitude generated by the Guidance algorithm of Section 2.4 to generate the control signal based on Eq. (20) and Eq. (22) of Section 2.5. The generated control signal is then mapped into the desired thrust in terms of a set of percentage values that are sent to the respective thrusters by the custom master node through the fiber-optic umbilical.

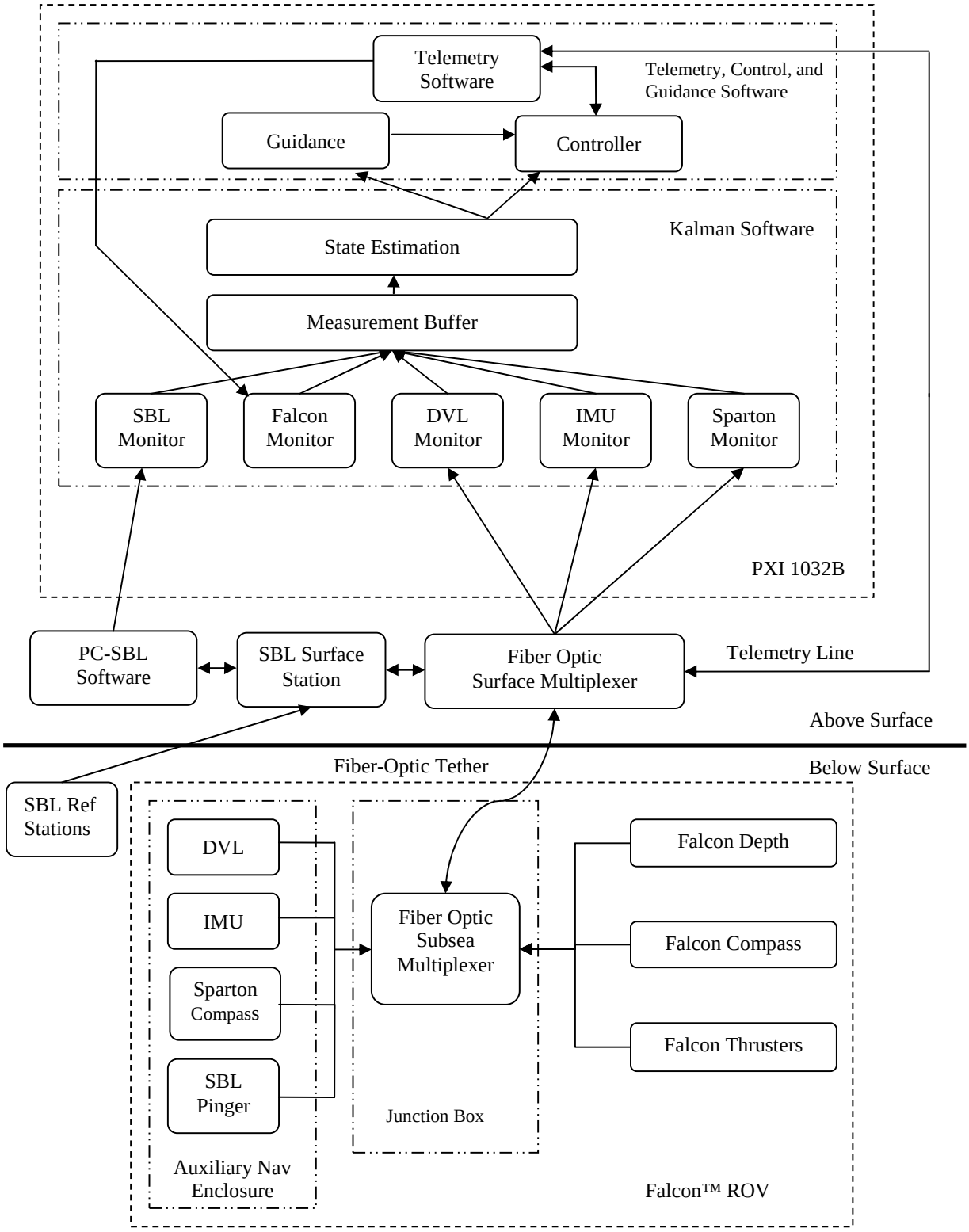


Figure 8: Signal data flow for the Falcon™ controller implementation.

4 SYSTEM IDENTIFICATION

4.1 Thruster Modelling

A set of bollard pull tests were performed in the wet laboratory to reveal the relationship between the ROV pilot's thrust commands and the resulting forces developed at the thrusters. To this end, the platform shown in Figure 9 was constructed and bolted to the working deck. On the wet side of the platform, a pulley was attached to the bottom of the metal pole. The pulley was used to redirect a cord, attached to an axial-load cell vertically mounted above the waterline, so that it ran horizontally to the ROV in the water. The setup ensured that the cord was aligned with the line of the commanded thrust during any test.

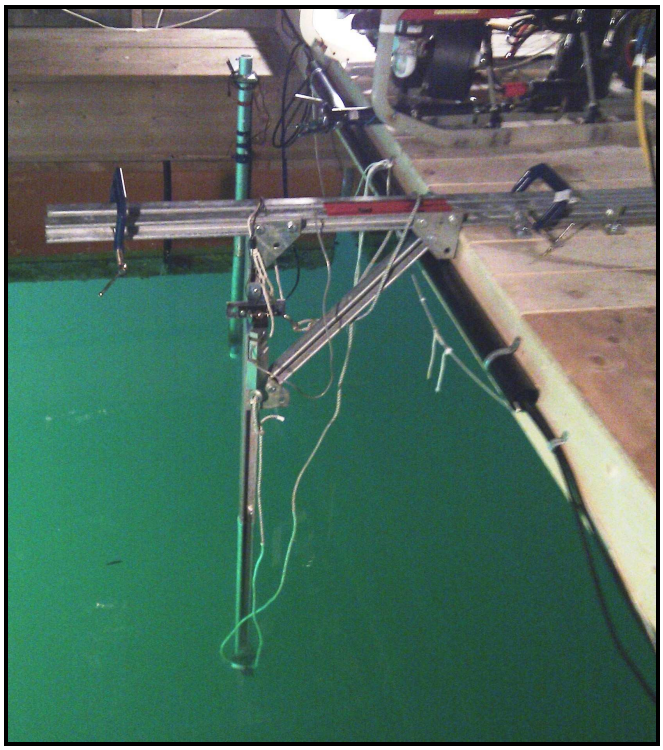


Figure 9: Thruster test platform.

In the thruster characterization stage, it was assumed that each thruster shows similar thrust performance, and hence the bollard pull test was performed for only one thruster. During any one test, the vehicle's horizontal plane was kept at the same depth as that of the pulley in order to minimize any potential rope misalignment that can potentially cause distortion on the measured force.

For the bollard pull test, a LabVIEW™ code was generated that automatically increments desired thrust by 5%, sustains that command for 30 seconds and records the desired thrusts along with the obtained load-cell output. Figure 10 and Figure 11 are *Thrust Percent Command* versus *Measured Thrust* graphs for the forward and reverse thrust manoeuvres, respectively.

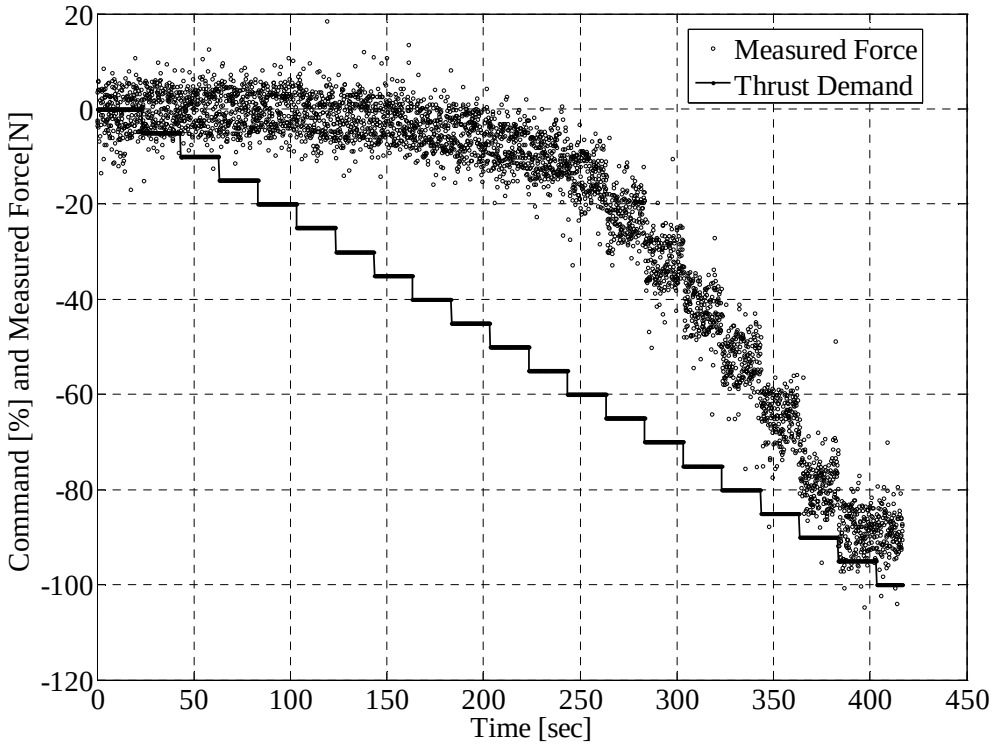


Figure 10: Forward thrust response.

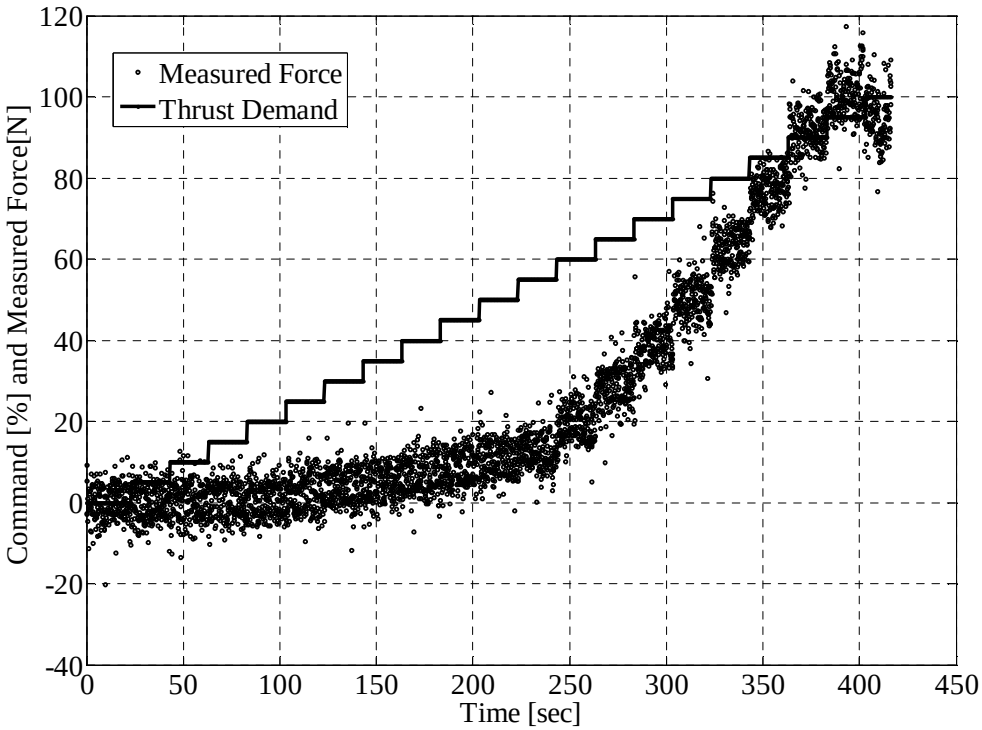


Figure 11: Reverse thrust response.

Using the steady-state average thrust values, a piecewise quadratic curve was fitted with a least-squares approach to relate the pilot's command; yielding the following set of equations:

$$\begin{aligned}
&0\% \le F_{PilotFwd} \le 54\% \quad \rightarrow F_{Thruster} = 0.0050445F_{Pilot}^2 - 0.035543F_{Pilot} \\
&54\% \le F_{PilotFwd} \le 90\% \quad \rightarrow F_{Thruster} = 0.032745F_{PilotFwd}^2 - 2.4862F_{PilotFwd} + 51.499 \\
&90\% \le F_{PilotFwd} \le 100\% \rightarrow F_{Thruster} = 90 \\
&F_{PilotFwd} \le 0\% \qquad \rightarrow F_{Thruster} = -0.94F_u(-F_{Pilot}).
\end{aligned} \tag{23}$$

The experimental thrust curve versus the fitted curve corresponding to Eq. (23) is shown in Figure 12. In the construction of Figure 12, the average measured force values were used for each thrust percentage.

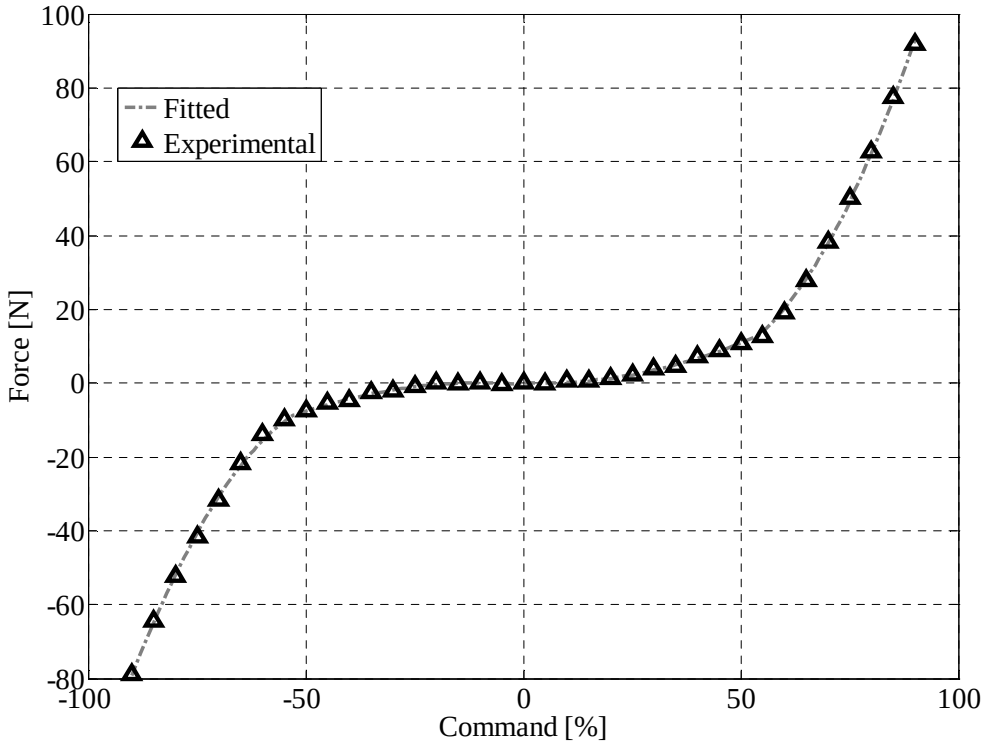


Figure 12: Steady state average surge thrust response to percentage thrust inputs.

Durıng the thruster modelling test, the rpm values were also recorded to reveal the relation between obtained thrust and the rpm. As expected from the lumped thruster dynamic model [27] , the obtained results indiate a near linear relationship between the steady state average thrust τ and the absolute squared propeller speed $\Omega|\Omega|$ as shown in Figure 13.

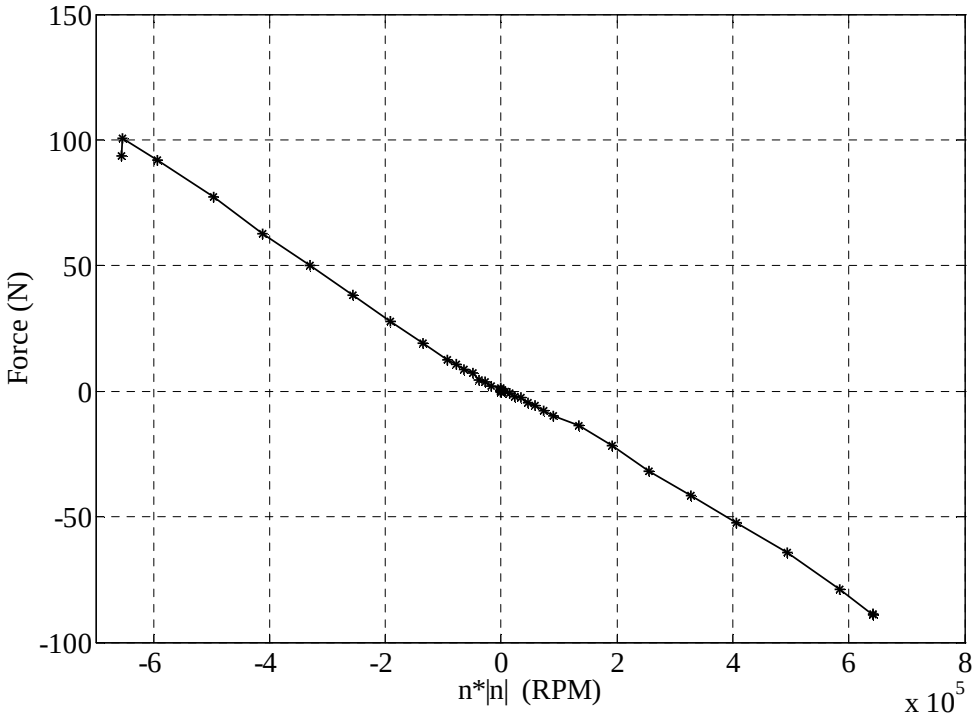


Figure 13: Steady state thrust as a function of steady-state absolute squared propeller speed.

The corresponding linear function can be defined as:

$$\tau = C_t \Omega |\Omega| \tag{24}$$

where $C_t = -0.00013719\ \mathrm{Ns}^2$ is a proportionality constant [27]. The obtained thruster dynamic model can be used to compensate for the thruster dynamics for a controller design as done in [27].

4.2 Dynamic Modelling of the Falcon™ ROV

Parameter identification tests were performed in the wet laboratory facility in Sidney. In these tests, similar to the work of [28], simplified 1-D manoeuvres were completed in order to identify the dynamic parameters in Eq. (4), Eq. (5) and Eq. (7) in Section 2.2.

For the modelling process, the DVL's body-fixed velocity output along the surge, sway and heave directions along with the IMU's yaw rate were recorded at 5 Hz for the DVL and 450 Hz for the IMU, respectively. During the tests, a set of 1-D step inputs were applied to the Falcon ROV with 10% increment at each sea-trial starting from 20% percent up to 80% along the surge, sway and heave and around the yaw axis of the vehicle, respectively. At each trial, the data were recorded until the system reached steady-state velocities. The conversion from commanded thrust percentage to achieved force values was carried out based on Eq. (23) obtained in the thruster identification stage summarized in Section 4.1. The roll and pitch motion of the Falcon were not considered since the motion around those axes cannot be driven due to the Falcon™'s existing thruster lay-out. In fact, these motions are self-regulated by the restoring moments due to the moment arm of the buoyancy force with respect to the center of mass of the vehicle, and hence can be assumed to be zero.

The drag coefficients along the three linear motion directions and around the yaw axis were found by fitting the drag equations in Eq. (7) to the Falcon™ data collected in the boathouse using the Matlab™ Parameter Estimation Toolbox. For the identification, the entire data set collected for each one degree of freedom test was fitted based on the nonlinear least-square method. Table 1 summarizes the dynamic parameters obtained from the field data.

Table 1: Falcon™ hydrodynamic parameters obtained from sea-trials.

Direction	Mass+Added Mass	Value	Coefficient	Value
Surge	$m + x_{\dot{u}}$	127.61 kg	k_u	0 Ns/m
			$k_{u\|u\|}$	150.91 Ns^2/m^2
Sway	$m + y_{\dot{v}}$	108.8 kg	k_v	0 Ns/m
			$k_{v\|v\|}$	598.6 Ns^2/m^2
Heave	$m + z_{\dot{w}}$	471.42 kg	k_w	56.941 Ns/m
			$k_{w\|w\|}$	294.75 Ns^2/m^2
Yaw	$I + I_{\dot{r}}$	25.441 $kg\ m^2$	k_r	7.2467 Nsm
			$k_{r\|r\|}$	69.839 Ns^2

In order to validate these parameters, simulation and experimental results were compared against each other. To this end, 70% thrust step inputs were applied, in each degree of freedom, in simulations of the Falcon™ ROV using the derived hydrodynamic parameters. The simulation results alongside with the DVL outputs for translational velocities and IMU output for the heading rate for the same thrust input are given in Figure 14 through Figure 17. The results indicate that the dynamic parameters sufficiently capture the ROV dynamics. The simulation's dynamic model constitutes the process model in the EKF as well as the model based part of the proposed controller.

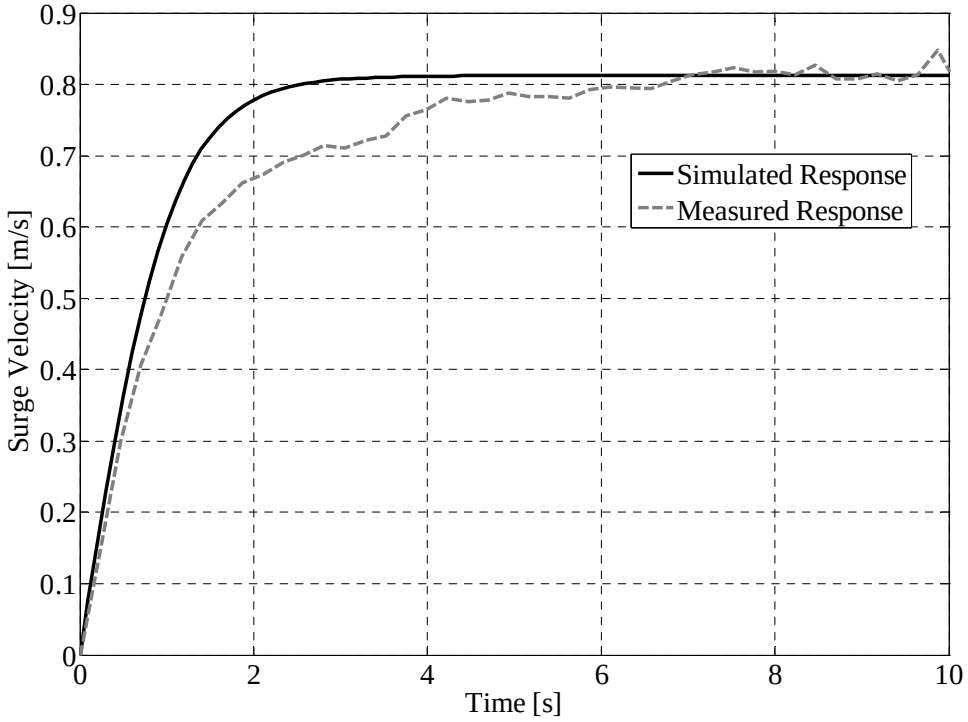


Figure 14: Surge model verification with DVL data.

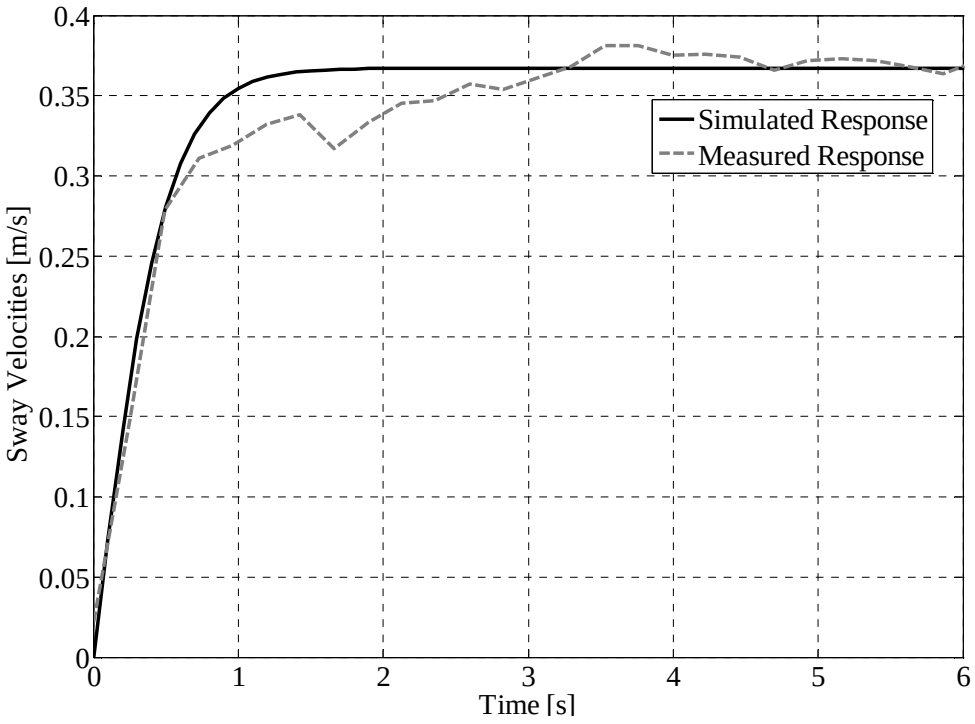


Figure 15: Sway model verification with DVL data.

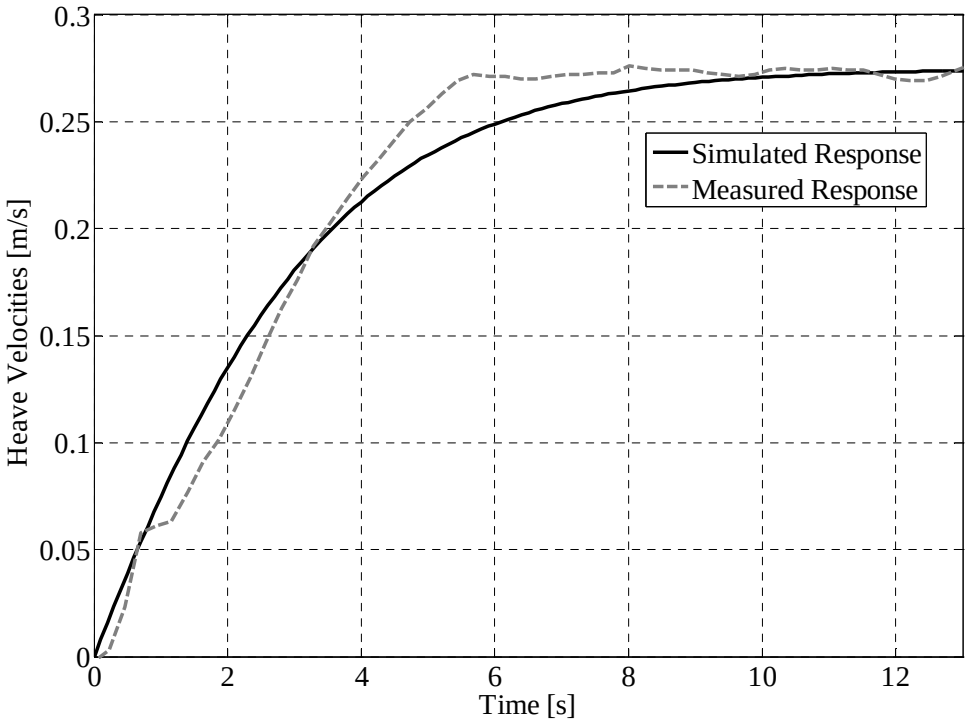


Figure 16: Heave model verification with DVL data.

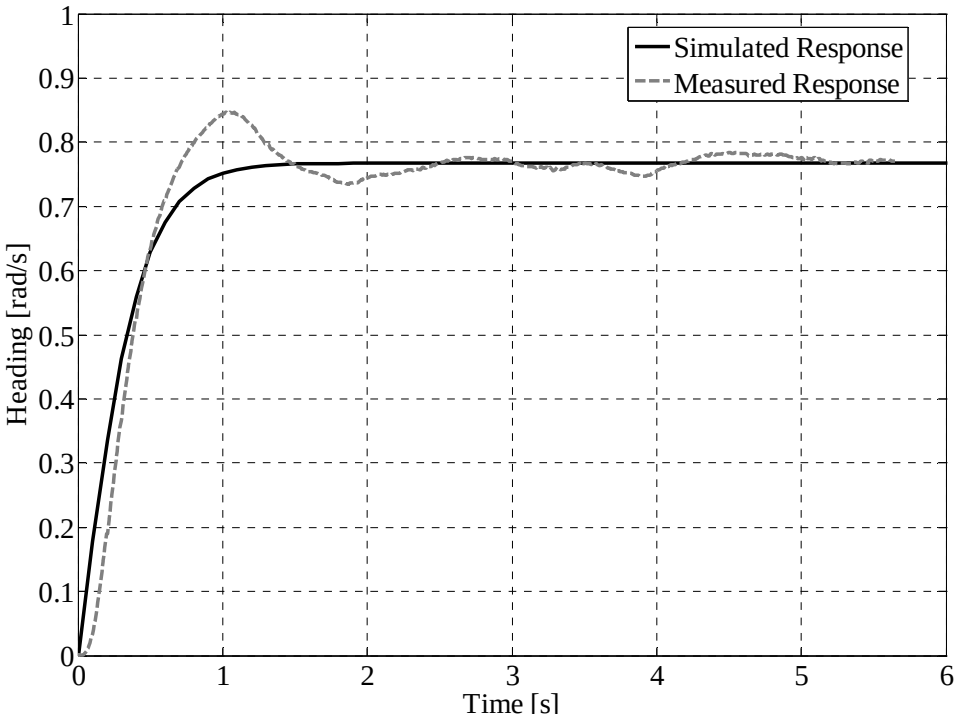


Figure 17: Yaw model verification with IMU data.

4.3 Kalman Parameters

For the EKF design, it is required to quantify the confidence in the measurements, in the form of the measurement covariance matrix R, for the calculation of the Kalman gain matrix. In the current work, the following sensor parameters were used:

Table 2: Falcon™ ROV measurement errors.

Measurement	Quoted Accuracy $(m/s)^2$	Static Variance	Dynamic Variance	Bias
Depth		$0.012\ m^2$	$0.036\ m^2$	
Surge u	$1.44x10^{-4}(m/s)^2$	$4 \times 10^{-5}\ (m/s)^2$	$4 \times 10^{-5}\ (m/s)^2$	1.5×10^{-4} (m/s)
Sway v	$1.44x10^{-4}(m/s)^2$	$8 \times 10^{-5}(m/s)^2$	$4 \times 10^{-5}\ (m/s)^2$	0.7×10^{-4} (m/s)
Heave w	$1.44x10^{-4}(m/s)^2$	$1 \times 10^{-5}\ (m/s)^2$	$1 \times 10^{-5}(m/s)^2$	0.5×10^{-4} (m/s)
Sparton Heading	$8 \times 10^{-5}\ rad^2$ Static	$8.4 \times 10^{-5}\ rad^2$	$3.3 \times 10^{-2}\ rad^2$	-
Sparton Pitch and Roll	$1 \times 10^{-5}\ rad^2$ Static	$6.1 \times 10^{-6}\ rad^2$	-	-
X Axis Gyro Error Variance	$0.003(°/s)^2$	$0.005\ (°/s)^2$	$0.005\ (°/s)^2$	-0.026 (°/s)
Y Axis Gyro Error Variance	$0.003(°/s)^2$	$0.010\ (°/s)^2$	$0.010\ (°/s)^2$	0.025 (°/s)
Z Axis Gyro Error Variance	$0.003(°/s)^2$	$0.008\ (°/s)^2$	0.008 (°/s)	0.080 (°/s)

The sensors outlined in Table 2 are the same as the ones used in the work of [19], and thus, the associated error measurement parameters were taken from the same work. However, since the SouthStar™ SBL acoustic position tracking system used in the current work is novel, new measurement error tests were performed for this device.

4.3.1 Kalman Parameter Identification for the SouthStar™ SBL

In order to tune the SBL parameters within the EKF, the exact value of the ranges being measured had to be obtained. Without an external truth signal, this is extremely difficult to complete due to the time varying static and dynamic biases, unidentified sensor misalignments, high noise levels and low resolution associated with the sensors employed in the sensor pod on the vehicle.

In the current work, the high precision VZ3000 Visualeyez™ vision-based position tracking system by Phoenix Technologies Inc. was used to generate true position reference values that are used in the calculations of the covariance matrices for the SBL system. The vision system comes with a set of cameras and optical markers (LED light sources). The camera units contain high precision light sensors, and as long as the markers are within the camera view of the light sensors, these built-in light sensors can track and estimate the position of the optical markers with very high precision – millimetre-level precision. In the experimental set-up, the camera units are mounted high up on the boathouse walls such that they illuminate a wide swath of the water surface. The markers were fixed to a lightweight mast that was attached to the ROV and stiffened with a set of guy wires to minimize the mast vibrations during tests. The mast keeps the markers above the water level during ROV sea-trials and hence allows operation of the vision system during ROV maneuvering. The camera locations in the boathouse are depicted in Figure 18.

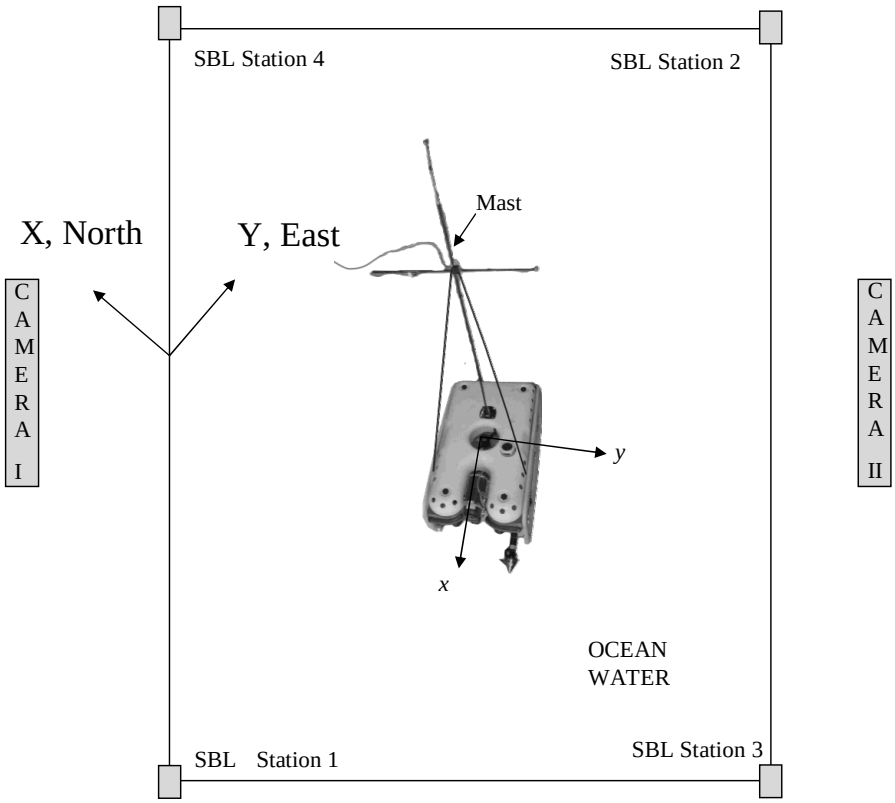


Figure 18: SBL and camera layout.

A stationary test was performed to obtain the static bias values associated with the range output of the SBL system. To this end, the four SBL stations were fixed to the deck of the boathouse as depicted in Figure 18 and their locations were found with respect to the NED reference frame using the motion capture system. The coordinates found are tabulated in Table 3. The target transducer was then put in one location for several minutes and its corresponding ranges were measured by both the SBL and motion capture systems. The differences in the range measurements were considered to be the static bias. The results are tabulated in Table 4.

Table 3: SBL reference station locations.

Reference Transducer #	X Location (m)	Y Location (m)	Depth (m)
1	-2.48	-1.9	0.74
2	-1.29	4.15	0.80
3	-4.72	0.06	0.80
4	0.9	2.1	0.74

Table 4: SBL range measurement bias.

Target Transducer			Station 1			Station 2		
X	Y	Z	True	Meas	Diff	True	Meas	Diff
2.56	0.260	1.86	3.52	3.71	-0.19	2.76	2.66	0.1
2.56	1.030	1.91	4.27	4.49	-0.22	2.2	2.1	0.1
2.56	-0.23	1.91	3.09	3.17	-0.08	3.12	3.08	0.04
2.56	1.05	1.87	4.29	4.35	-0.06	2.21	2.21	0
Mean Bias					**-0.14**			**0.06**

	Station 3			Station 4		
	True	Meas	Diff	True	Meas	Diff
	3.71	3.98	-0.27	2.28	2.19	0.07
	4.38	4.64	-0.26	1.73	1.71	0.02
	3.27	3.02	-0.19	2.73	2.73	0
	4.42	4.58	-0.16	1.69	1.69	0
Mean Bias			**-0.22**			**0.02**

In order to find the dynamic biases, the Falcon™ ROV was operated continuously within the confines of the boathouse and its motion was tracked with the SBL system as well as the motion capture system. The results for the range values based on the sea-trials are shown in Figure 19 through Figure 22 and the dynamic range bias and error variance values based on the sea-trials were tabulated in Table 5. The calculated range

error variances were used in the EKF's measurement covariance matrix R related to each reference station range measurement. In Figure 19 through Figure 22, the SBL ranges are shown as grey stars whereas the motion capture system ranges are shown as bold lines. Since the ranges can be calculated from the motion capture system only when the ROV is within the camera view, there are gaps in the motion capture data that occurred when the ROV travelled to the corners of the test range.

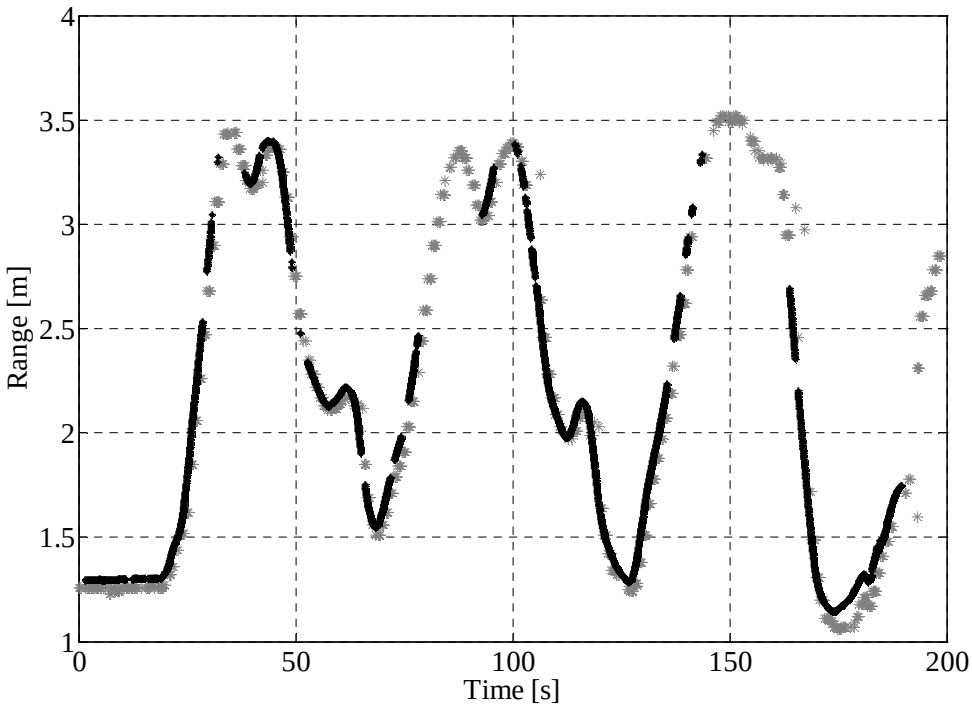


Figure 19: SBL ranges versus motion capture system for Range 1.

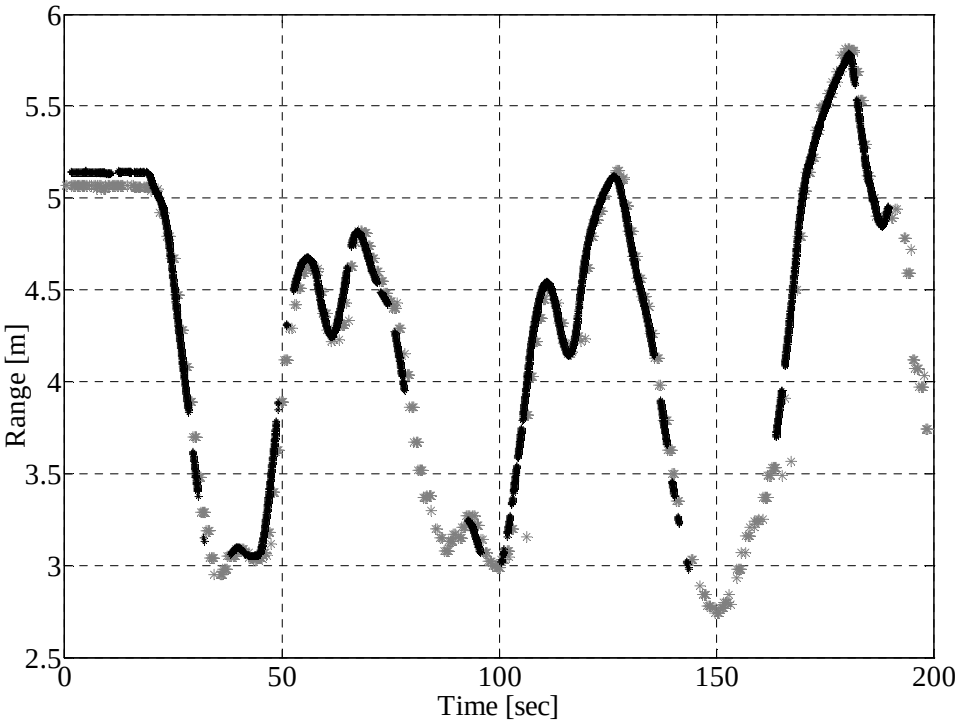


Figure 20: SBL ranges versus motion capture system for Range 2.

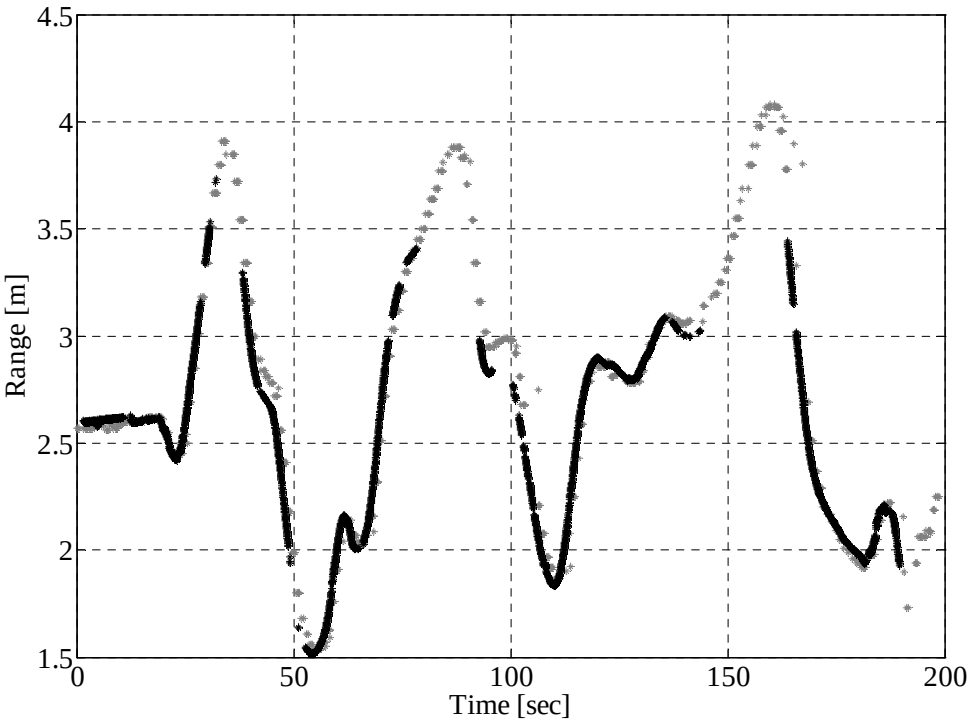


Figure 21: SBL ranges versus motion capture system for Range 3.

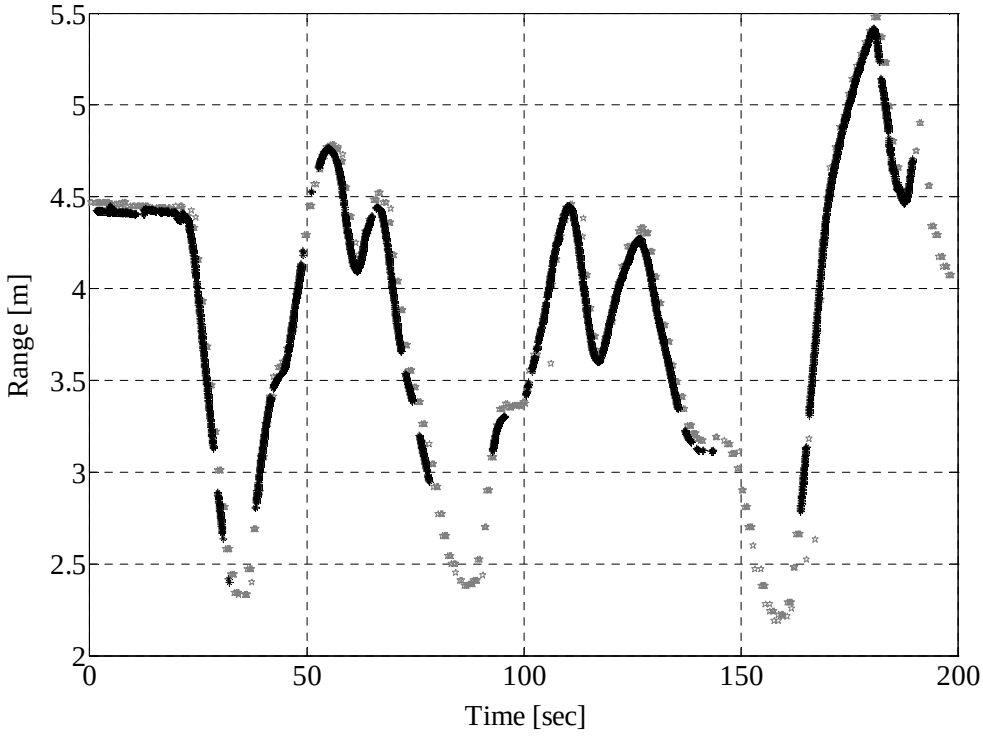


Figure 22: SBL ranges versus motion capture system for Range 4.

Table 5: SBL range error and variances.

Reference Transducer #	Range Bias	Range Error Variance
1	-0.03m	0.03 m^2
2	0.06m	0.02 m^2
3	-0.1m	0.04 m^2
4	0.02m	0.04 m^2

4.3.2 Estimating Dynamic Model Uncertainty

In EKF design, it is necessary to quantify the confidence in the EKF's process model in order to propagate state error covariance matrix via Eq. (15). This confidence is specified through the error covariance matrix **Q**, and the approach presented in [19] was used in the current work to this end. The approach first defines a reference EKF or aided EKF

whose output is considered to be “truth” estimates. The aided EKF generates “truth” estimates by using the complete set of instrument measurements. With the overestimated **Q** entries, the aided EKF generates “truth” states that rely heavily on the measurements. Secondly, an unaided EKF is defined whose output corresponds to the output of the EKF’s process model. The difference between the state estimations of the aided EKF and the unaided EKF is considered to be the error in the EKF process model.

In order to identify the accuracy of the EKF process model, an unaided EKF is created by removing select measurement(s) from the aided EKF. The removal of measurements makes the output of the unaided EKF rely heavily on the process model. This step is performed in four stages. That is, first, the rotation rate model uncertainty is found by excluding the IMU rotational rates of p, q, r from the aided EKF to obtain unaided EKF, and the difference is found. From this difference, the error variances are calculated and the time rate of change of the error variance is entered in the corresponding location of the $\dot{\boldsymbol{Q}}$ matrix. This step is repeated for the Euler angle model uncertainty by excluding Sparton compass measurements of ϕ, θ, ψ; for the velocity model uncertainty by excluding the DVL measurements of u, v, w; and finally for the position model uncertainty by excluding the SBL and depth measurements.

These steps were completed based on a new set of ROV state data obtained through driving the ROV in the boathouse. The results of this process, based on sea-trials, are tabulated in Table 6.

Table 6: Kalman filter process error model.

Axis	Variance
Roll, p	$5.2\text{x}10^{-5}$ $(\text{rad/s})^2$
Pitch, q	$3.0\text{x}10^{-5}$ $(\text{rad/s})^2$
Yaw, r	$1.2\text{x}10^{-5}$ $(\text{rad/s})^2$
Roll	$2.3\text{x}10^{-6}$ rad^2
Pitch,	$6.1\text{x}10^{-8}$ rad^2
Yaw	$1.9\text{x}10^{-5}$ rad^2
Surge, u	$4.1\text{x}10^{-5}$ $(\text{m/s})^2$
Sway, v	$6\text{x}10^{-6}$ $(\text{m/s})^2$
Heave, w	$4.1\text{x}10^{-5}$ $(\text{m/s})^2$
North, X	$6\text{x}10^{-5}$ m^2
West, Y	$7.5\text{x}10^{-5}$ m^2
Depth, Z	$4.8\text{x}10^{-7}$ m^2

4.3.3 Verification of the EKF during Shallow Water Test

The experimental validation of the designed EKF through a near surface test manoeuvre was performed in order to evaluate the accuracy of the EKF's state estimation. To this end, the EKF state estimation was compared against the highly accurate position data obtained from the optical motion capture system. During the tests, the ROV was piloted randomly within the confines of the water test facility while maintaining a depth of 1 meter to keep the optical markers above the water level.

In Figure 23, the EKF estimate of the horizontal positions were compared to that of the motion capture systems, and the corresponding error plots and RMS errors are reported in Figure 24 and Table 7, respectively. The results indicate that the EKF algorithm successfully captures the ROV's horizontal motion, which was derived based on the acoustic range measurements and the estimate of the ROV velocity and the orientation.

These results are believed to be satisfactory for the subsequent controller implementation of the Falcon™ vehicle.

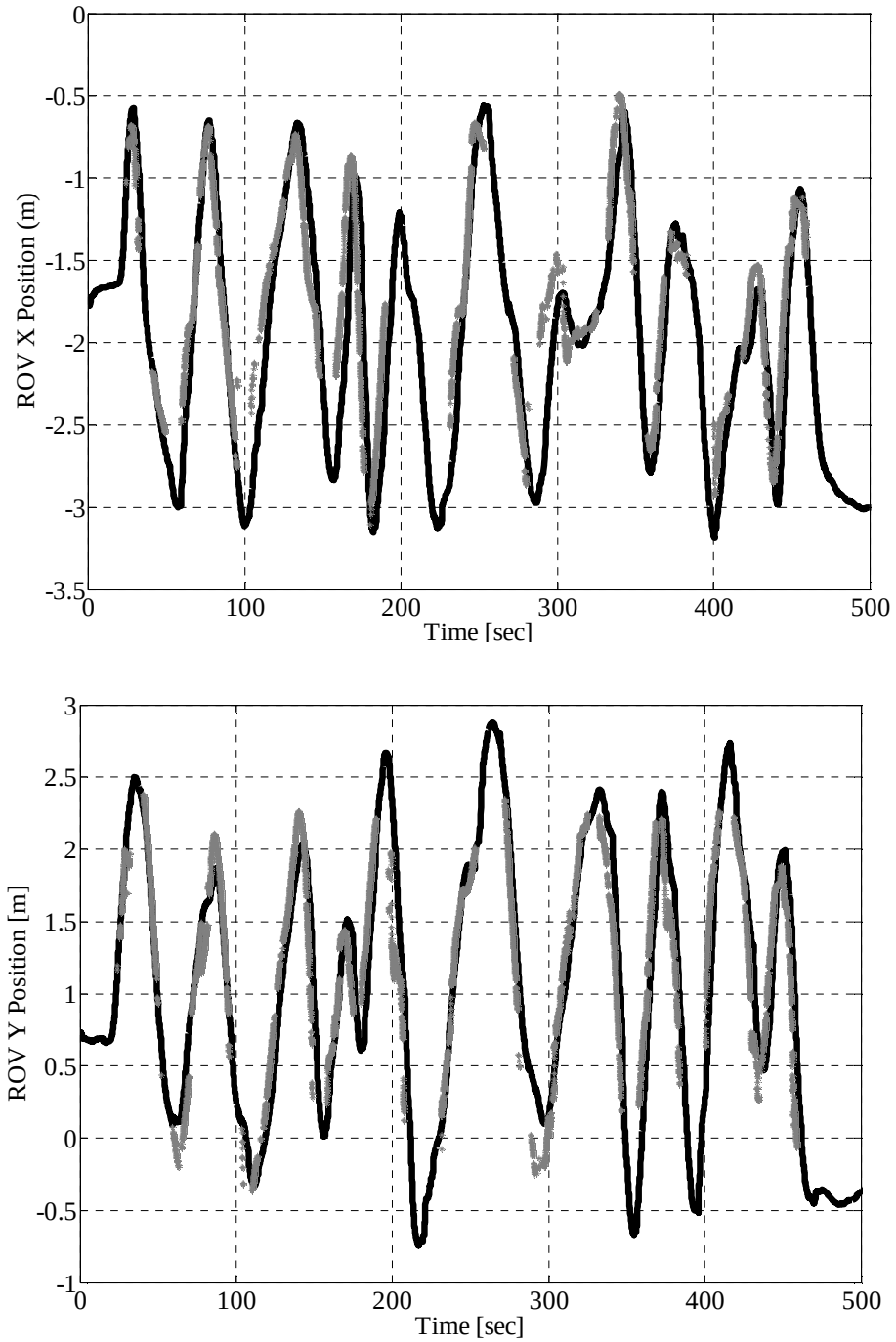


Figure 23: ROV X position (top) and Y position (bottom) during the shallow water test. The optical measurements are grey broken lines.

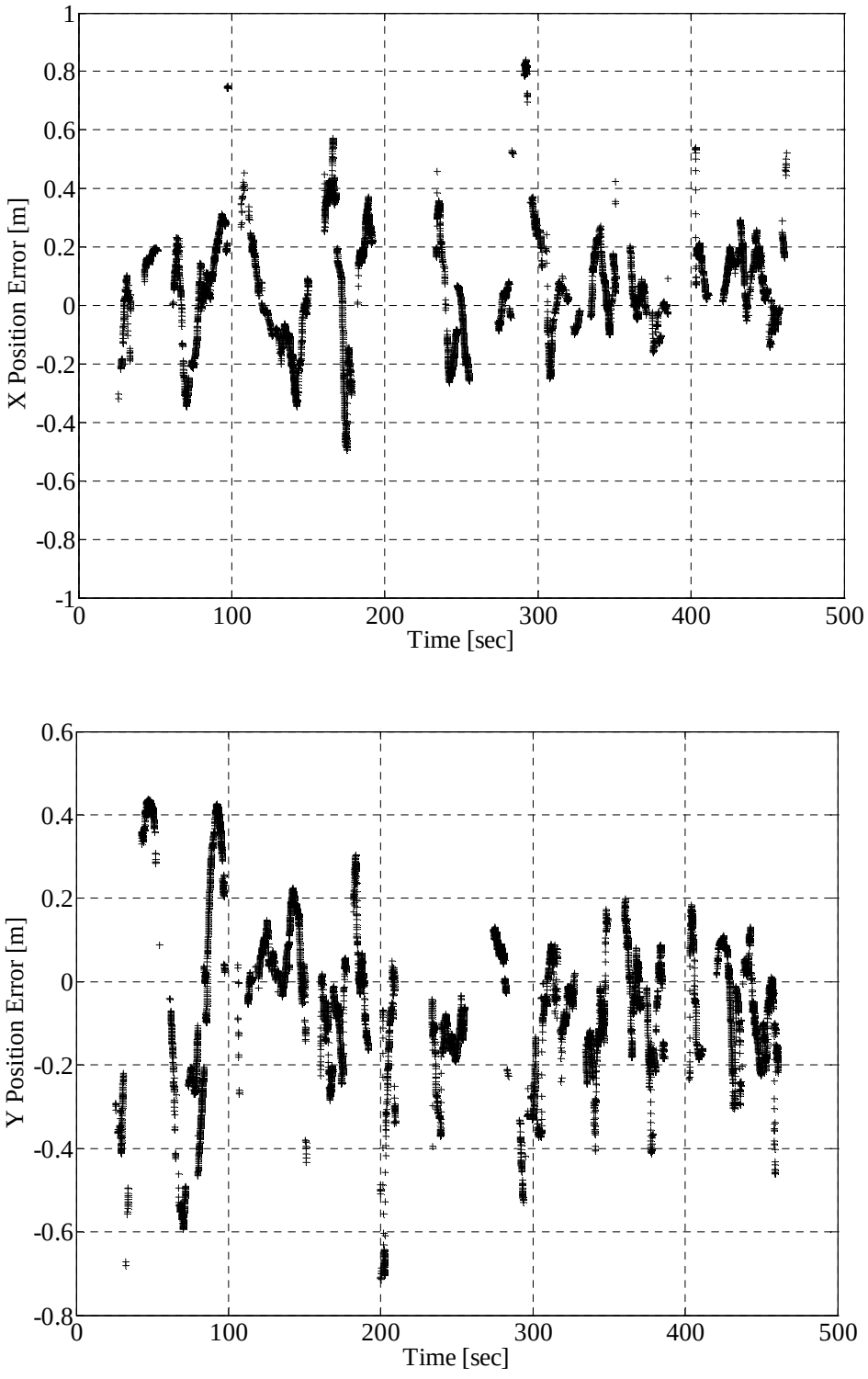


Figure 24: ROV X position error (top) and Y position error (bottom) during the shallow water test. Missing data points correspond to times when there is no optical motion capture data available.

Table 7: RMS error of the horizontal position estimates during the shallow water test.

Coordinate	EKF Position RMS Error (m)
X	0.15
Y	0.16

5 CONTROLLER FIELD TESTS

The controller discussed in Section 2.5 was implemented on the inspection-class Seaeye Falcon™ ROV. As mentioned in Section 3.2, the real-time implementation of the proposed controller was carried out using National Instruments' PXI-6251 real-time target machine, which runs the sensor drivers, the EKF, the guidance and the controller algorithms. The controller's feedback signal was the state estimation performed using the algorithm presented in Section 2.3 The controller uses this estimation along with the desired vehicle attitude produced by the Guidance algorithm of Section 2.4 to generate the control signal based on Eq. (20) and (22) of Section 2.5. The generated control signal is then mapped into the desired thrust using a simple pseudo-inverse solution [20]. The desired thrust values were mapped into a set of percentage values for the thrusters, based on Eq. (23), that were consistent with the existing Falcon^{TM} telemetry. Using this scheme, a number of wet tests were performed in the water facility to determine suitable control gains. While the testing was based on trial and error, it was extensive and the authors have found that the gains, reported in Table 8, provide good performance.

Table 8: Tuned controller parameters.

	Λ	k_1	k_2	υ	$\mathbf{K}_{P,0}$	$\mathbf{K}_{I,0}$	$\mathbf{K}_{D,O}$	$\hat{\rho}_0$
X	2	0.1	0.1	0.1	40	5	150	20
Y	2				50	5	150	
Z	0.8				50	10	60	
ψ	0.6				10	7	50	

5.1 Step Input Field Tests

In order to evaluate the step input performance of the controller, the Falcon™ was asked to move to a certain point from its current position. These tests were performed individually for the X (surge), Y (sway), Z (heave) and ψ (yaw) motions, respectively. To generate the desired velocities in terms of the NED reference frame for a given set of desired positions, the guidance algorithm of Section 2.4 was used with the parameters of Table 9 obtained through the sea-trials. The results are given in Figure 25 through Figure 40. The time history of the controlled state values in response to the step input applied are shown in Figure 25, Figure 29, Figure 33, and Figure 37. As can be seen in these figures, the controller successfully guides the system state to its desired value with a small overshoot, small rise time, and small steady state errors despite the existence of coupled dynamics due to the motions along\around non-controlled axes. As can be seen in Figure 33, the performance of the controller along the Z axis is not as well as the other

axes. This can be partially attributed to that fact that the sensor resolution on the Falcon™ depth sensor is 0.1m.

In regards to the time history of the adaptive parameters, they are demonstrated in Figure 27, Figure 31, Figure 35, and Figure 39. As the figures reveal, the adaptation law of Eq. (22) provides smooth values for the adaptive terms and ensures they converge to values that provide satisfactory controller performance. In addition, it does not lead to parameter drift phenomenon, which might have otherwise occurred due to the lack of a significant input signal - especially after the initial large transient motion has decayed to yield small velocities. Finally, Figure 28, Figure 32, Figure 36, and Figure 40 illustrate the output of the controller in terms of force values at the center of mass of the Falcon™ ROV. Small oscillations effect was observed in the controller output that can be partially attributed to a wide dead-band that was observed for the on-board thrusters observed during the thruster characterisation. High chattering action especially for the heave motion observed in the control force can be attributed to the noisy state feedback. This problem can be remedied through light filtering.

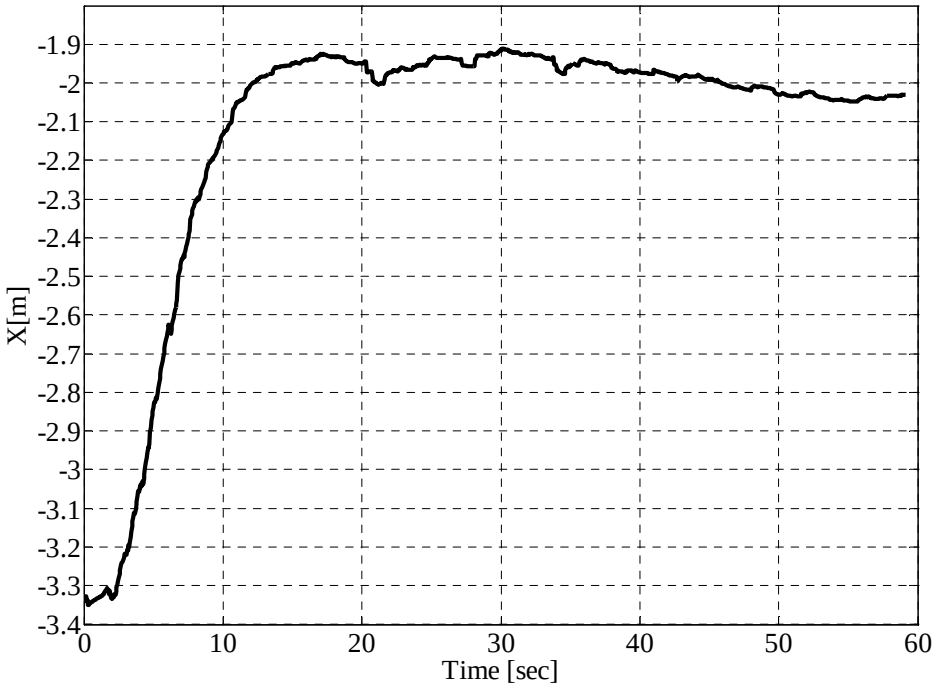


Figure 25: Step input response for the X (surge) motion.

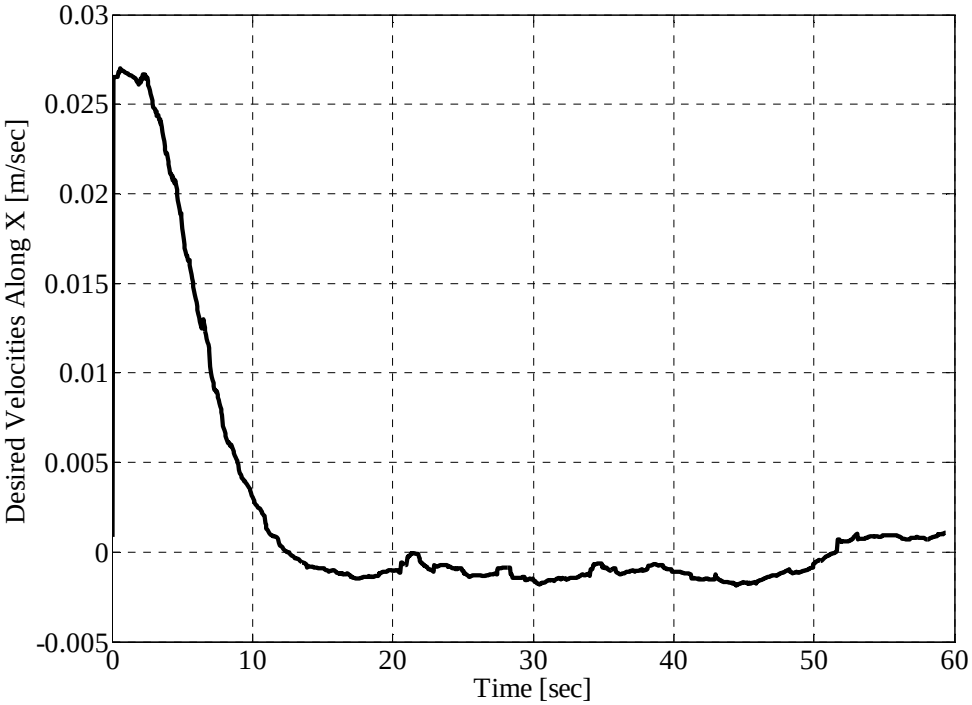


Figure 26: Time history of the desired velocities along the X axis.

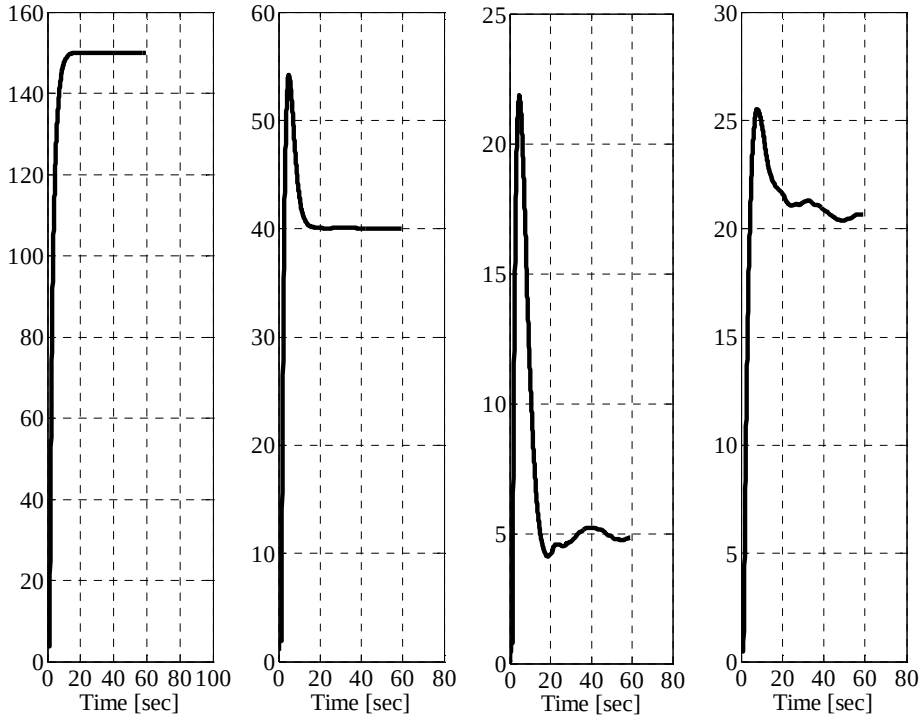


Figure 27: Time history of adaptation parameters for the X (surge) motion.

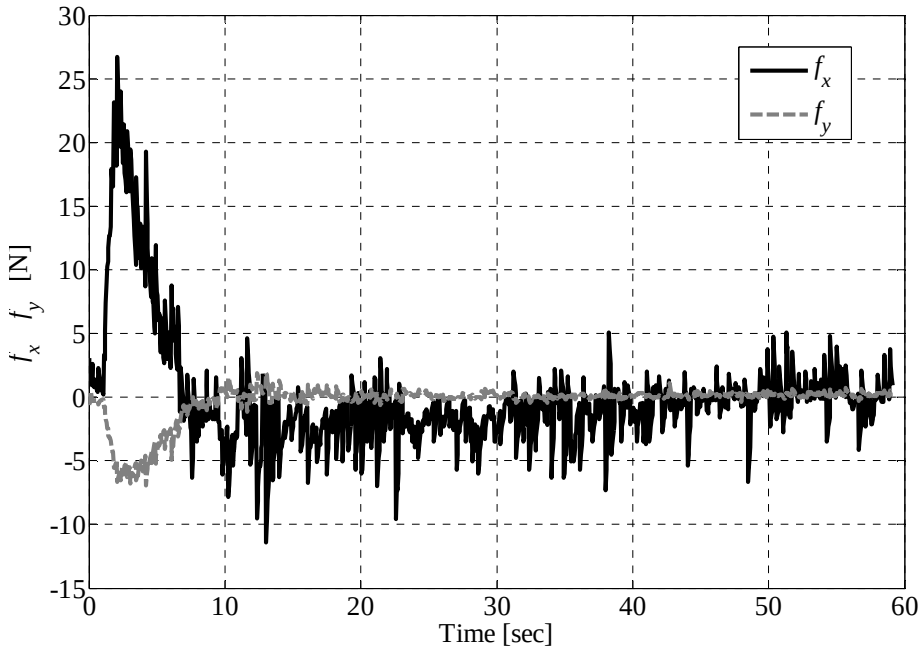


Figure 28: Time history of controller force output for the X (surge) motion.

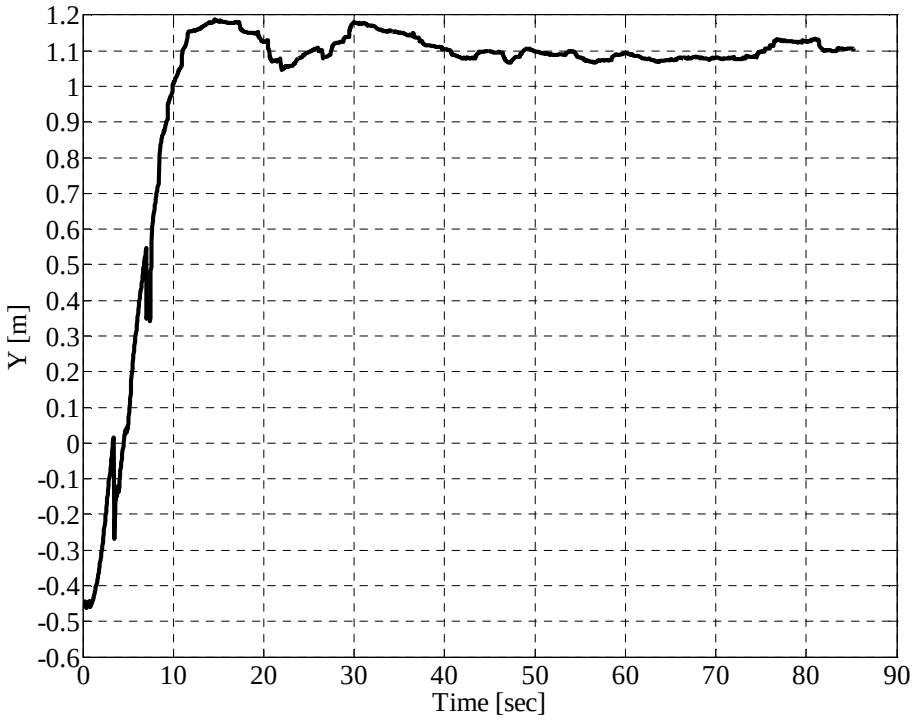


Figure 29: Step input response for the Y (sway) motion.

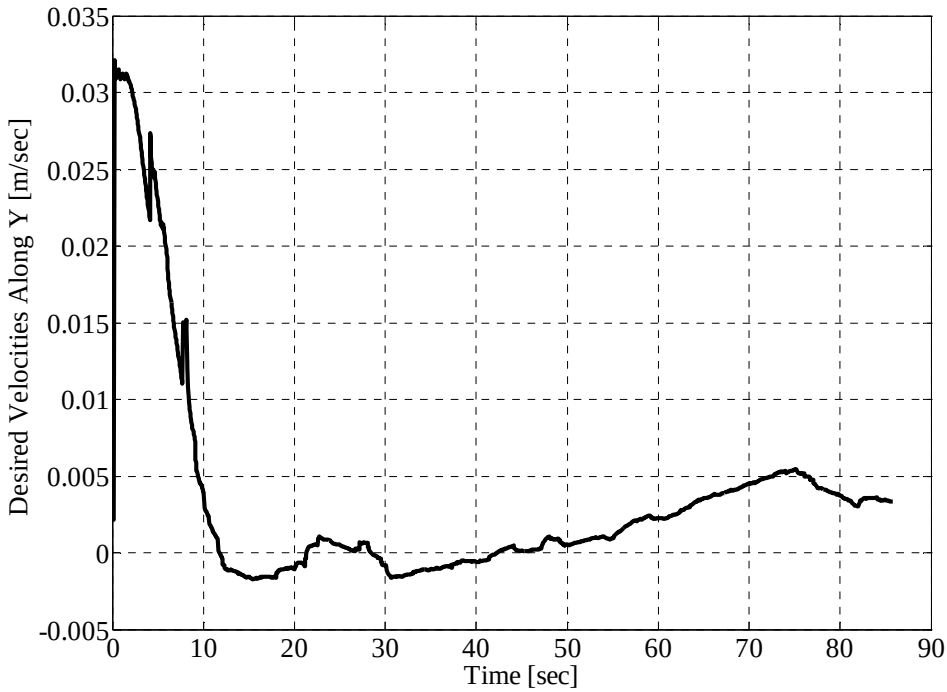


Figure 30: Time history of the desired velocities along the Y axis.

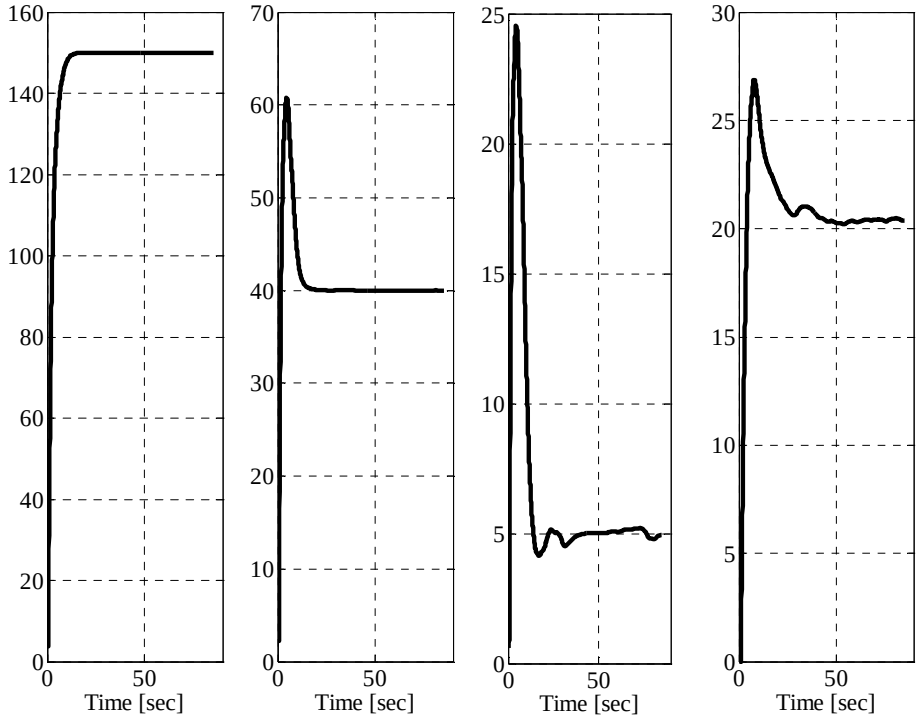


Figure 31: Time history of adaptation parameters for the Y (sway) motion.

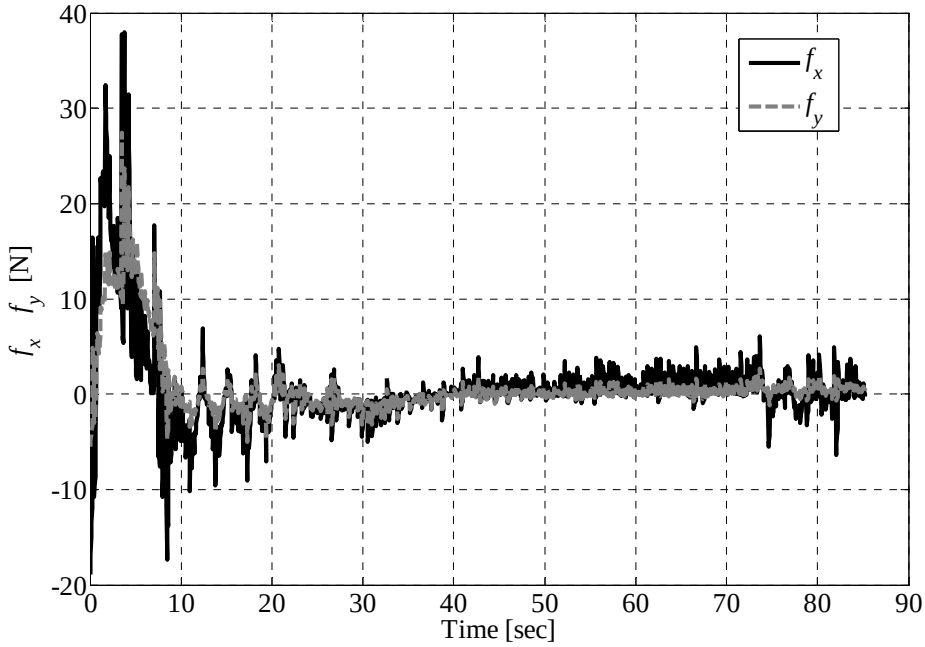


Figure 32: Time history of controller force output for the Y (sway) motion.

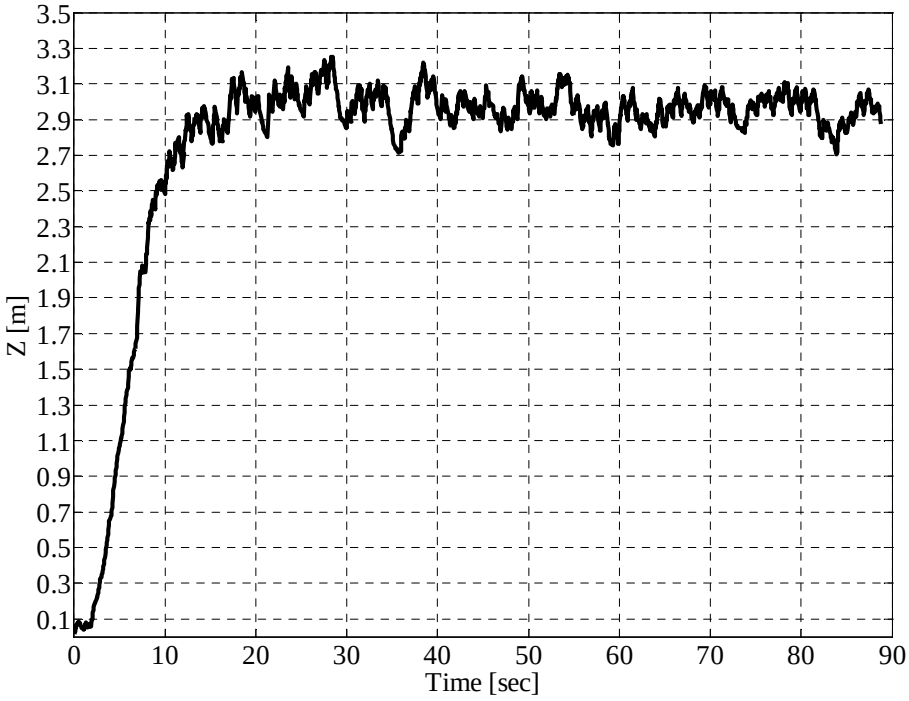


Figure 33: Step input response for the Z (heave) motion.

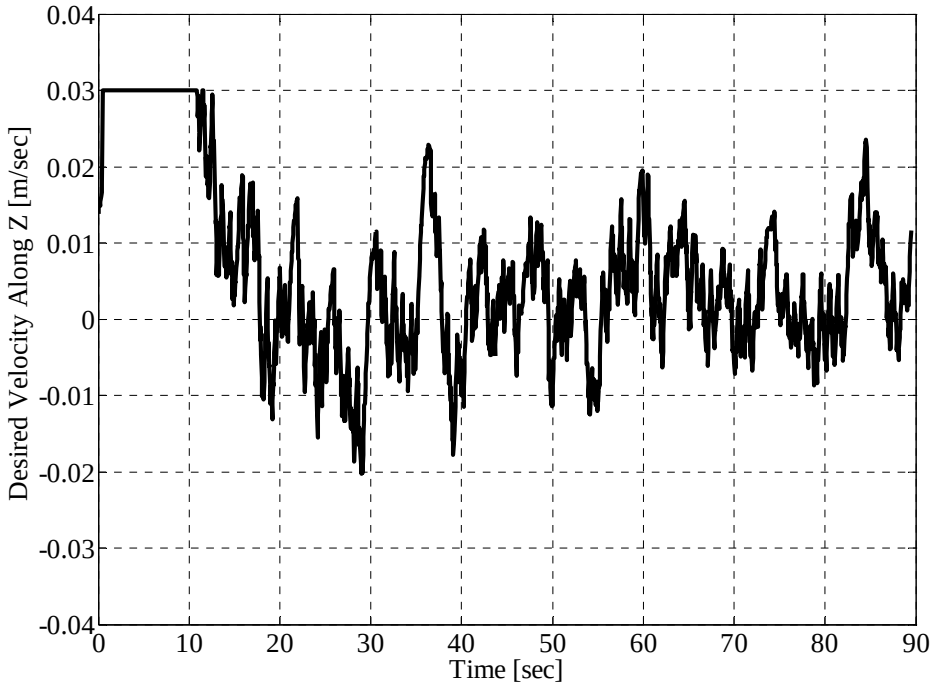


Figure 34: Time history of the desired velocities along the Z axis.

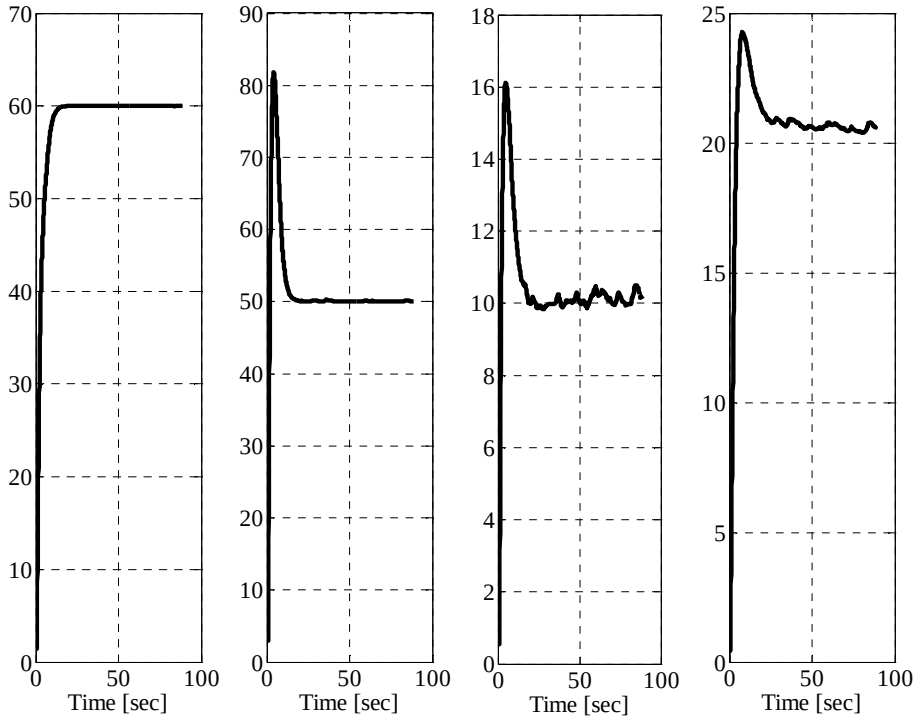


Figure 35: Time history of adaptation parameters for the Z (heave) motion.

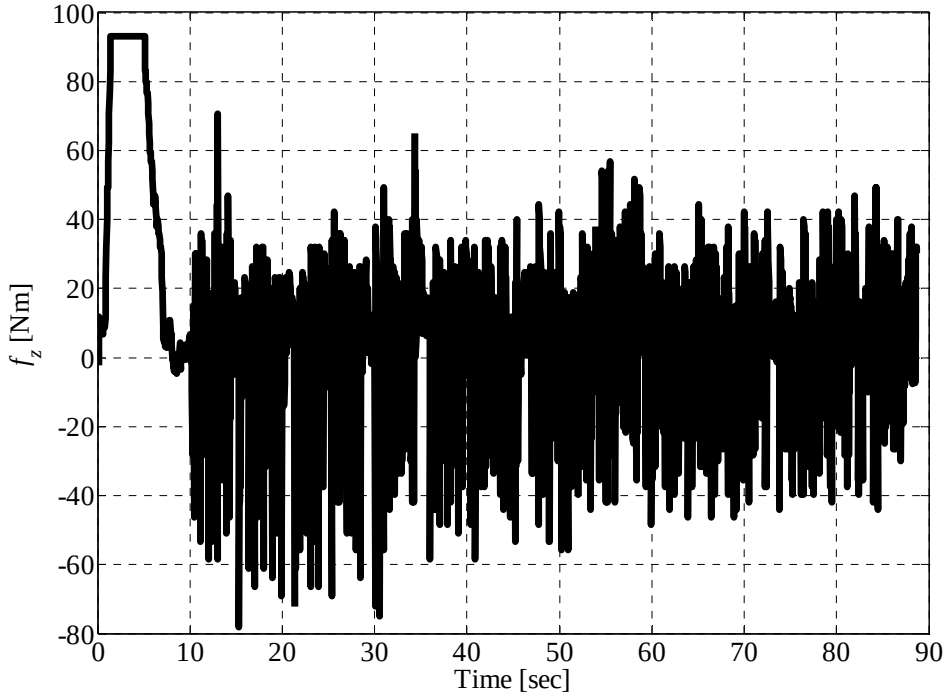


Figure 36: Time history of controller force output for the Z (heave) motion.

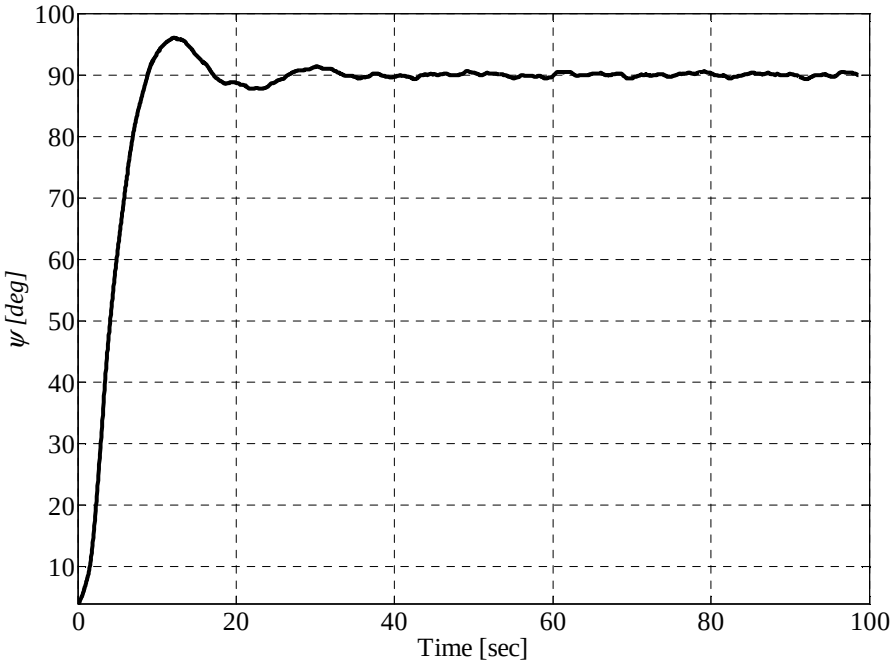


Figure 37: Step input response for the ψ (yaw) motion.

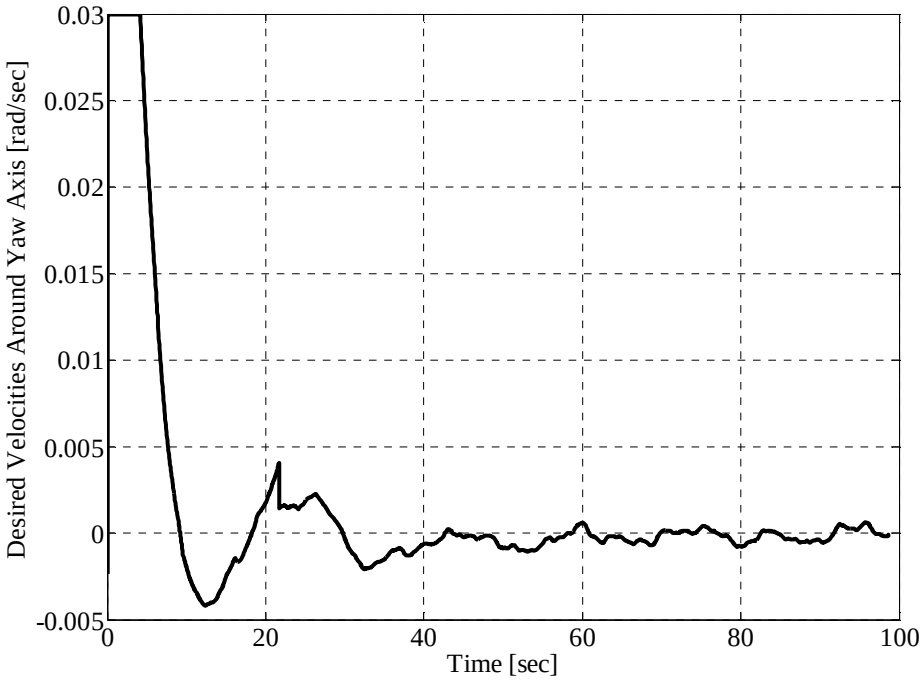


Figure 38: Time history of the desired velocities around the yaw axis

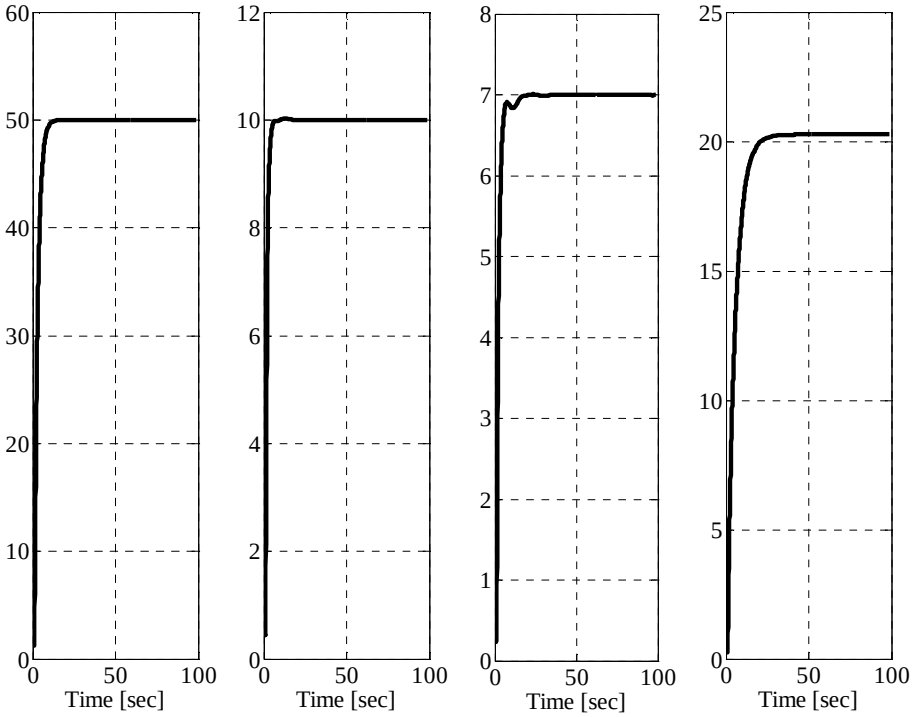


Figure 39: Time history of adaptation parameters for the ψ (yaw) motion.

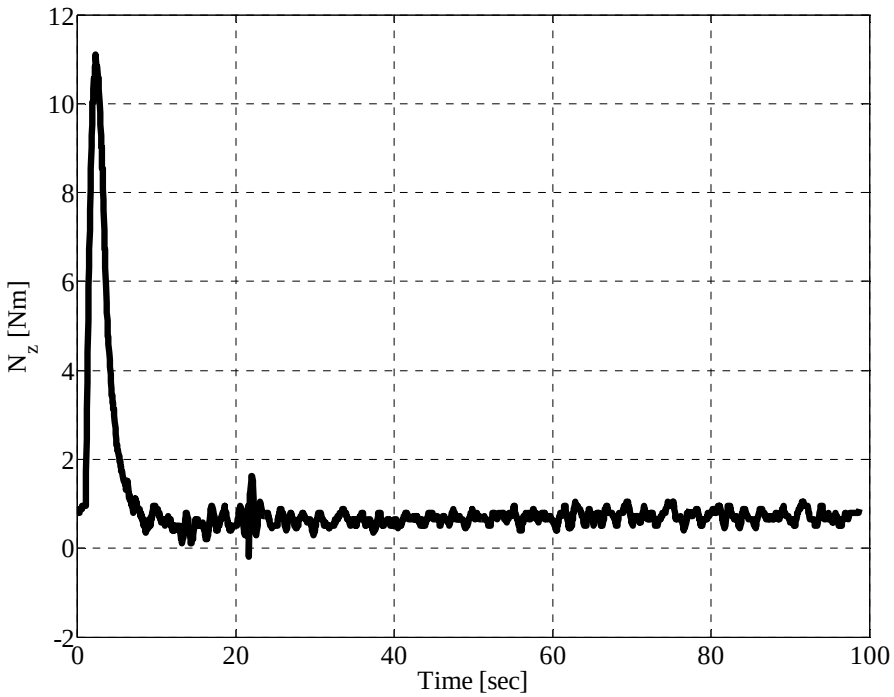


Figure 40: Time history of controller force output for the ψ (yaw) motion.

5.2 Horizontal Motion and Depth and Yaw

In this subsection, the controller performance in following a desired *Horizontal Motion* (X, Y) and also *Depth and Yaw* (Z, ψ) was evaluated. To generate the desired position, velocities and accelerations in terms of the NED reference frame for a given set of way-points, the guidance algorithm of Section 2.4 was used with the parameters of Table 9.

In the first task, the vehicle's horizontal motion was controlled. To this end, the Falcon™ ROV was requested to move from its initial position of (X=-1.9m, Y=-0.5m) to the desired final position of (X=0.3m, Y=-1.3m). T he results of the controller manoeuvre are shown in Figure 41. As can be seen from the figures, the Falcon™ ROV successfully reached the desired points with satisfactory performance.

Table 9: Guidance algorithm parameters.

	λ_p	λ_I	ξ_p	ξ_I	σ
X	0.02	0.01	0.05	0.02	0.035
Y	0.02	0.01	0.05	0.02	0.035
Z	0.08	0.05	0.3	0.2	0.2
ψ	0.04	0.04	0.03	0.02	0.035

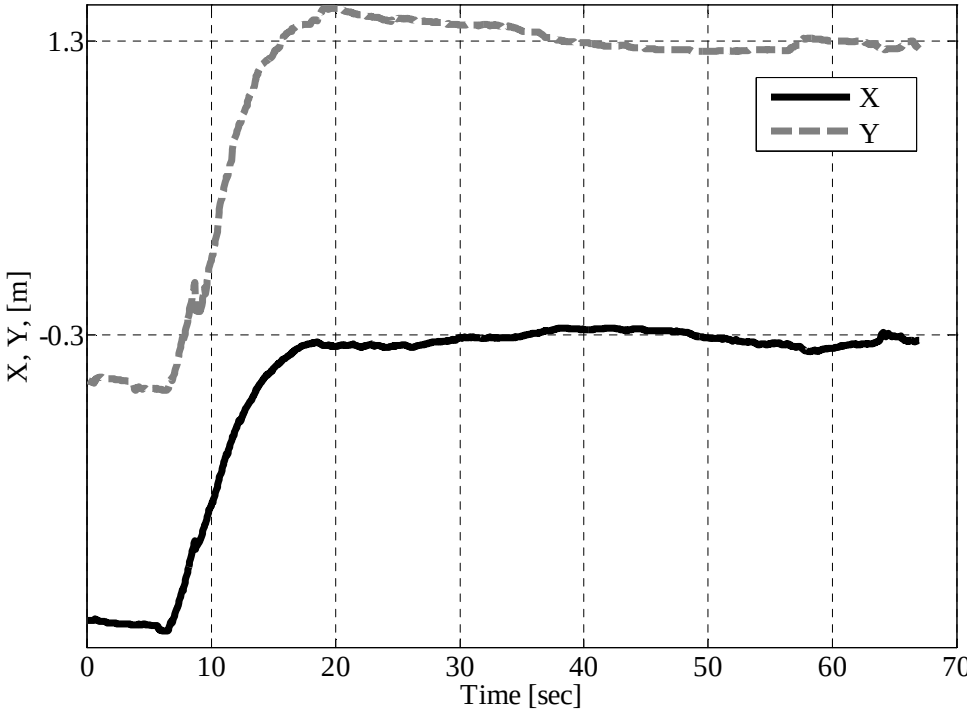


Figure 41: Time history of the Falcon™ ROV's horizontal motion.

In the second task, only the vehicle's heave and depth motions were controlled. To this end, the Falcon™ ROV was requested to move from its initial configuration of ($Z=0.2, \psi = 0\,\text{deg}$) to the desired final position of ($Z=1.5,\ \psi = 90\,\text{deg}$). Similar to the horizontal motion case, the controller successfully guided the respective states to their desired values satisfactorily.

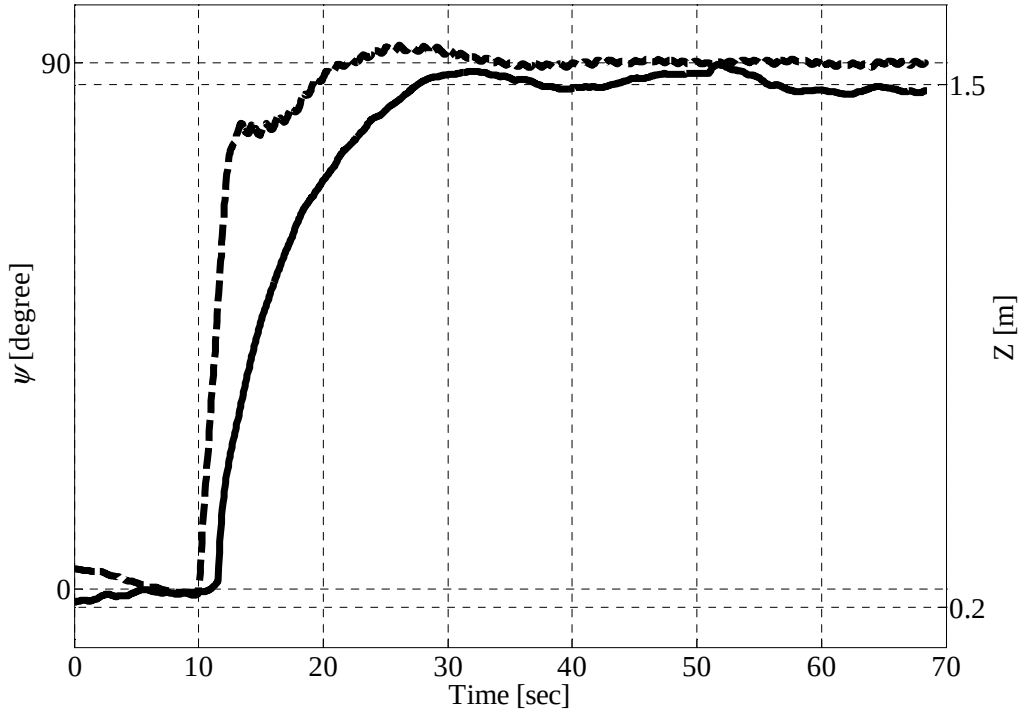


Figure 42: Time history of the Falcon™ ROV's depth and heave motions.

5.3 Way Point Tracking Performance

In this test, the controller's ability to navigate way-points was evaluated. As opposed to the other cases studied, this case involves simultaneous control of four axes; namely, X, Y, Z and ψ. To evaluate the way-point tracking performance of the controller, the Falcon™ ROV was requested to move through the way-points of Table 10 in a sequential manner. In this test, the guidance algorithm of Section 2.4 was used. The guidance algorithm receives the way-point definitions and generates desired state velocities. The resulting values do not pose any constraints on the travel time and the type of the motion and solely guides the vehicle to the desired way-points. The generated velocities along with way-point destinations provide reference values to the controller.

Table 10: Way-Points in terms of the NED reference frame.

Way Point Number	X_d (m)	Y_d (m)	Z_d (m)	ψ_d (deg)
#1	-1.9	-0.6	1.8	50
#2	-0.29	1.31	1.8	50
#3	-0.29	1.31	1.8	140
#4	-1.82	2.6	1.8	140
#5	-1.82	2.6	1.8	50
#6	-3.43	0.68	1.8	50
#7	-3.43	0.68	1.8	140
#8	-1.9	-0.6	1.8	140
#9	-1.9	-0.6	1.8	50

Figure 43 is the plan view of the desired and obtained way-point manoeuvre while Figure 44 and Figure 45 are the corresponding time history of the controlled states. As the figures demonstrate, the controller guides the vehicle through the way-points with centimetre-level precision and provides stable vehicle motion throughout the manoeuvre.

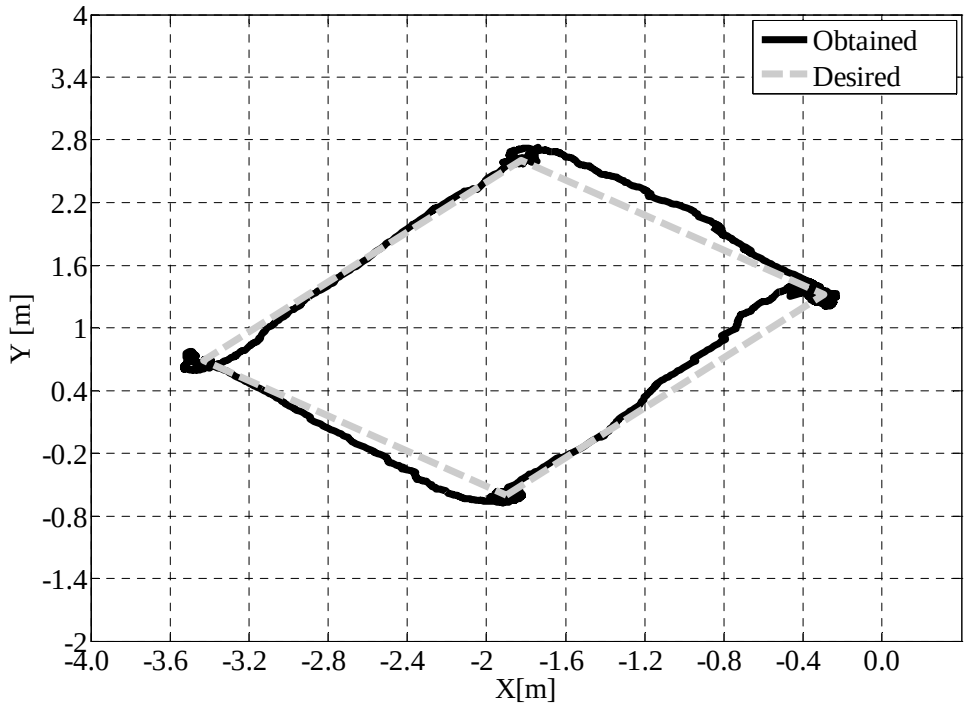


Figure 43: Plan view of the Falcon™ ROV manoeuvre for way-point tracking.

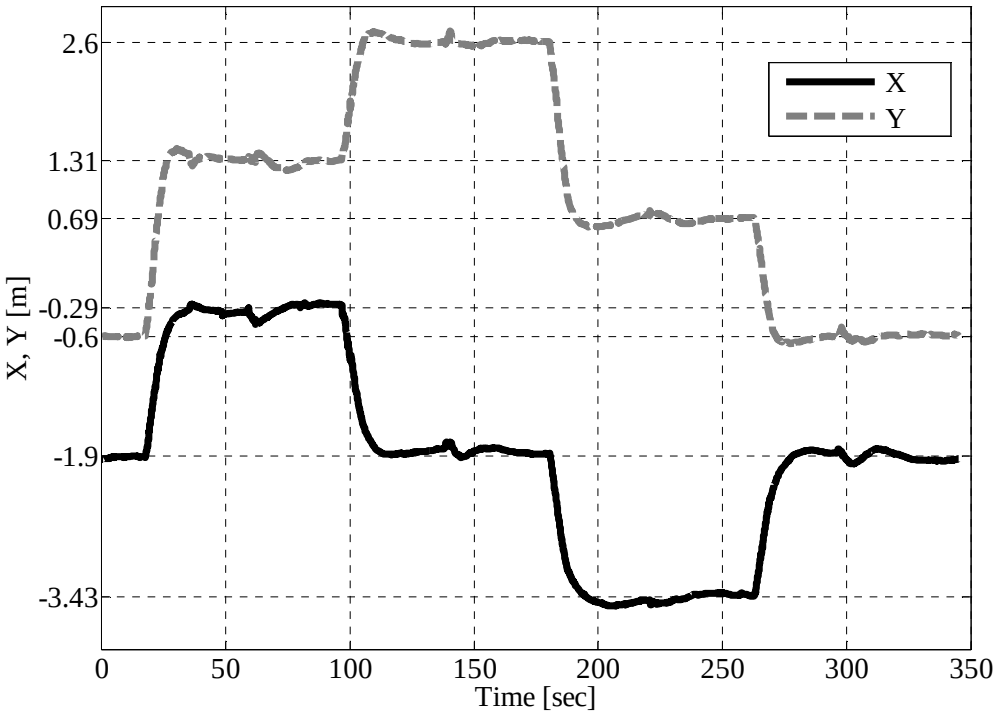


Figure 44: Time history of the Falcon™ ROV's surge and sway motions.

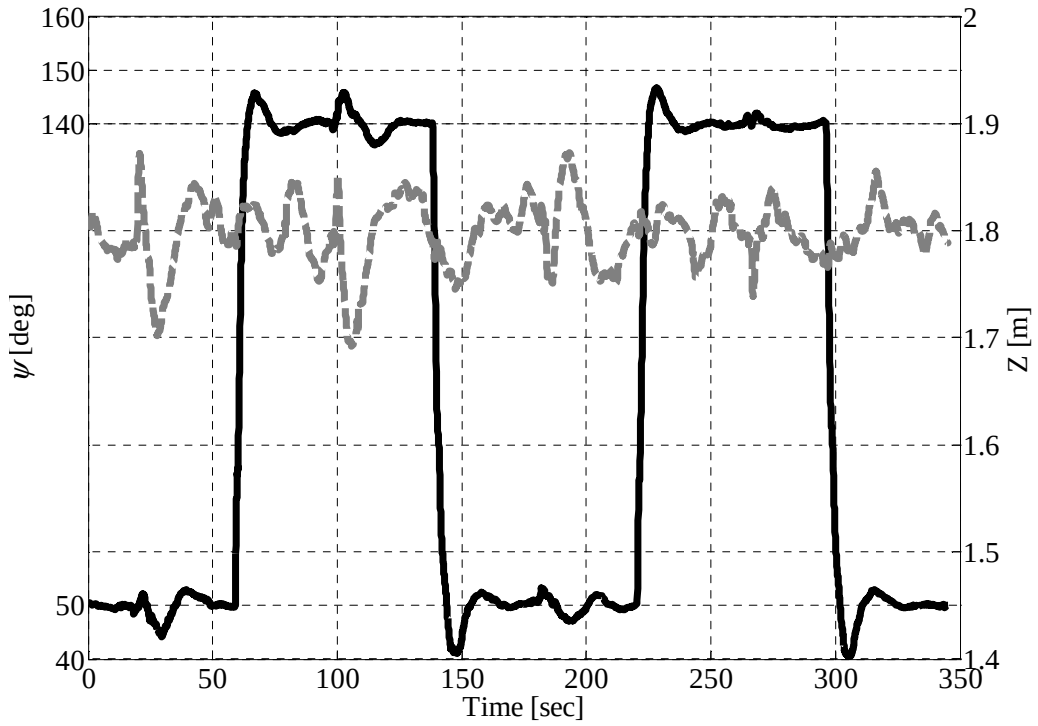


Figure 45: Time history of the Falcon™ ROV's heave and depth motions.

5.4 Automatic Trajectory Following

In this test, the Falcon™ ROV was demanded to follow a time-dependant desired trajectory instead of merely reaching a desired position as with the cases of the previous sections. The way-points of Table 10 were applied again. In this test, instead of the guidance algorithm of Section 2.4, a third-order polynomial function was used to generate reference velocity values as well as position values. Each segment was defined by the two consecutive way-points was needed to be traversed in 20s while the rotational segments were traversed in 10s.

Figure 46 is the plan view of the desired and obtained manoeuvre for the trajectory following task. To better judge the controller's trajectory following performance, the desired and obtained position and velocity values as well as the corresponding position

and velocity errors for each individual controlled axis are demonstrated in Figure 47, Figure 48, Figure 49 and Figure 50.

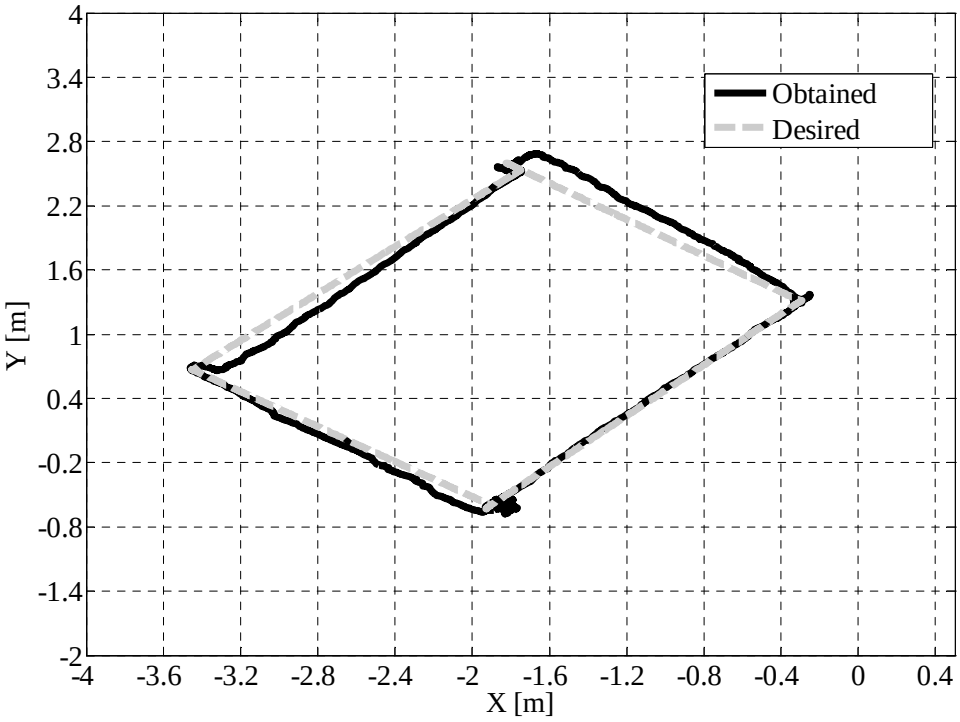


Figure 46: Plan view of the Falcon™ ROV manoeuvre for trajectory following.

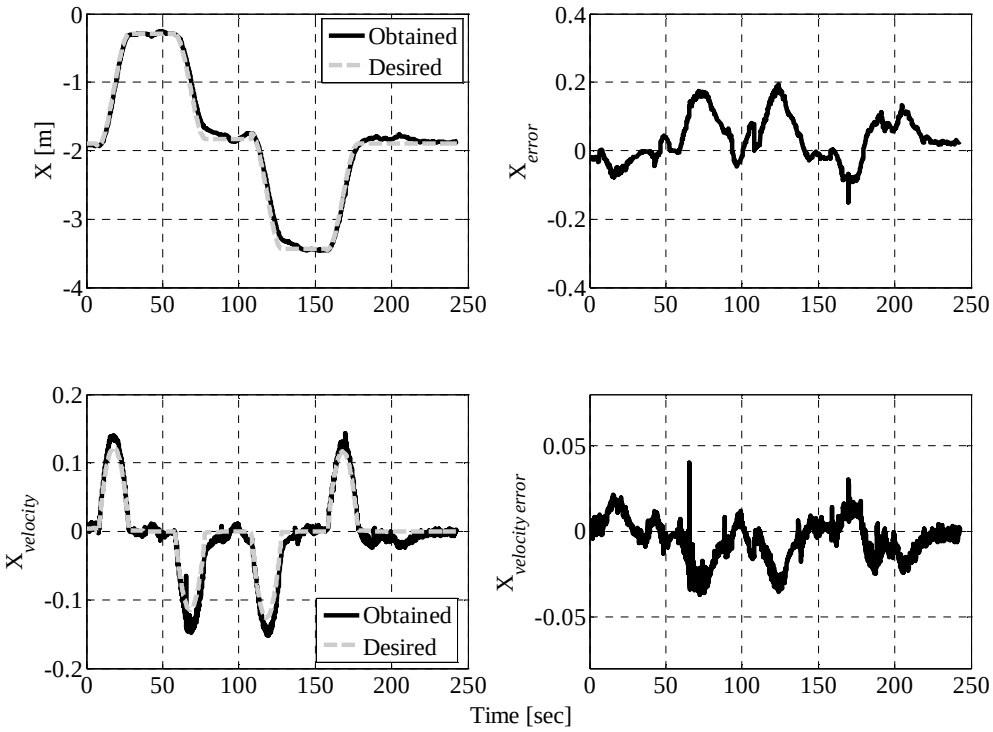


Figure 47: Controller performance for the surge motion.

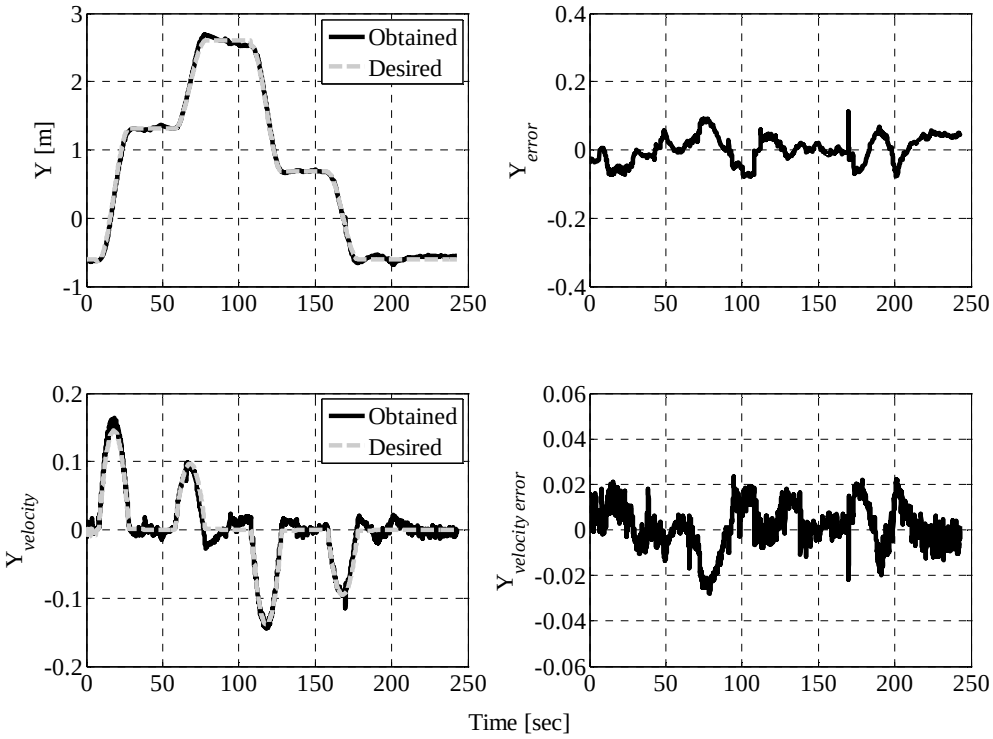


Figure 48: Controller performance for the sway motion.

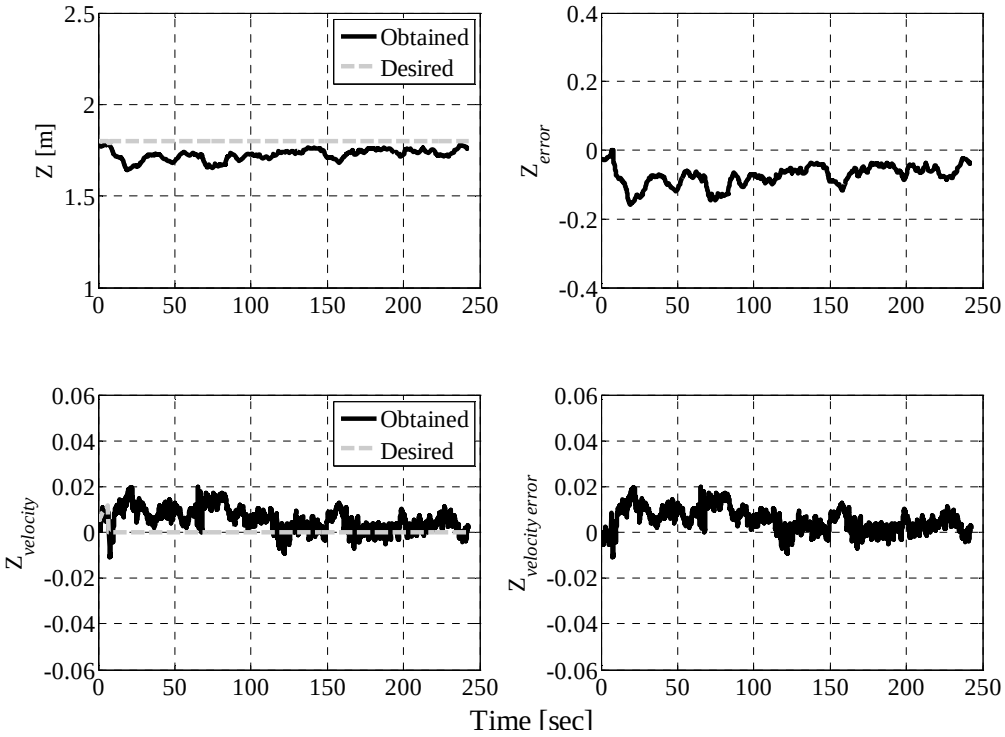


Figure 49: Controller performance for the heave motion.

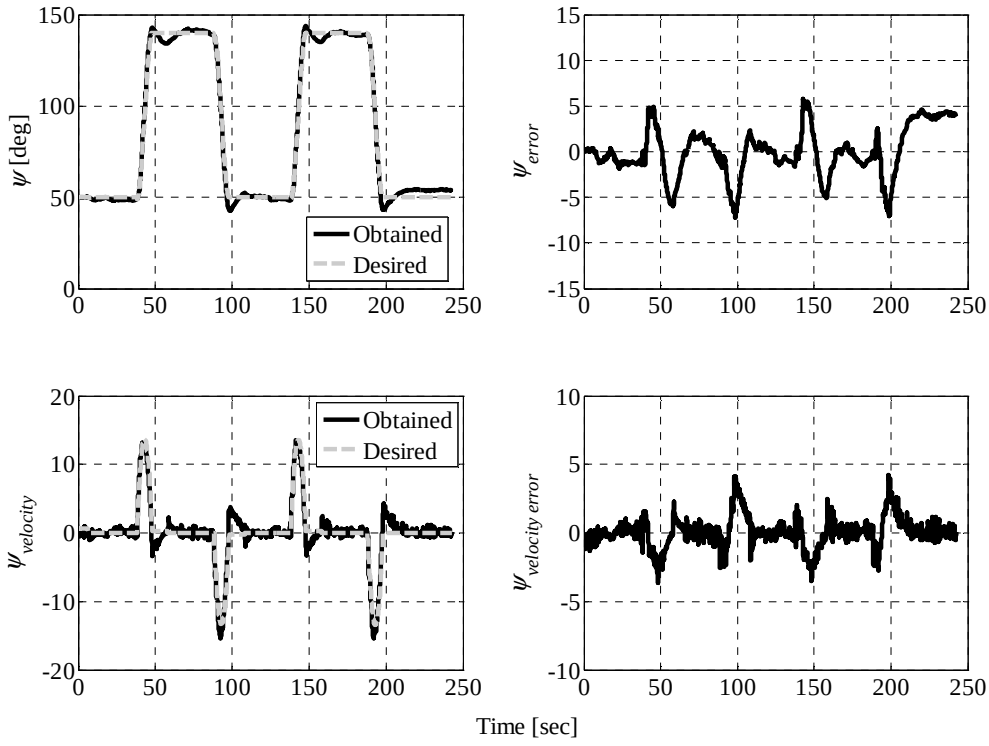


Figure 50: Controller performance for the yaw motion.

To reveal the repeatability of the controller, the same manoeuvre was performed four times and the corresponding error values were tabulated in Table 11. As the table indicates, the error values exhibits very small changes; indicating repeatable sea performance.

Table 11: Trajectory tracking performance evaluation.

	X			Y			Z			ψ		
	MEAN	STD	RMS	MEAN	STD	RMS	MEAN	STD	RMS	MEAN	STD	RMS
Test 1	0.03	0.07	0.08	-0.008	0.04	0.04	-0.06	0.04	0.08	-0.81	2.52	2.64
Test 2	0.04	0.07	0.08	-0.001	0.04	0.04	-0.08	0.04	0.08	-0.38	2.62	2.65
Test 3	0.03	0.07	0.07	0.002	0.04	0.04	-0.07	0.03	0.08	-0.02	2.69	2.69
Test 4	0.01	0.07	0.07	-0.01	0.07	0.04	-0.07	0.03	0.08	-0.7	2.38	2.47

6 CONCLUSION

A complete algorithm that involves guidance, navigation and a control algorithm has been proposed and implemented on the inspection-class Falcon™ ROV. In the context of navigation, low noise and high update rate was obtained using , an EKF has been designed that blends the measurements of an acoustic positioning system, a DVL, a low-cost IMU, a compass, and a depth sensor and uses a dynamic model of the ROV to further increase performance of the navigation algorithm. In addition, pressure hulls have been designed for the sensor suite of the Falcon™ ROV. The accuracy of the EKF in estimating the system states has been demonstrated through wet tests using a high

precision optical motion capture system. In terms of the controller, a novel controller has been proposed that contains two layers of adaptivity; the first layer is for adjusting PID gains and the second layer is for estimating the upper bound on a lumped uncertainty vector. The proposed adaptation laws are robust to the parameter drift problem that would otherwise lead to instability in the presence of poor system input signals. The controller performance in the way-point and trajectory following tasks has been demonstrated through a number of wet tests. The experimental tests have demonstrated the efficacy of the proposed guidance, navigation and controller algorithm.

Acknowledgment

The authors would like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for providing financial support for this work. Also, the authors wish to extend their thanks to Suboceanic Sciences Canada Ltd. in Duncan, BC for their participation and support in the research.

References

[1] G. Conte and A. Serrani, “Robust control of a remotely operated underwater vehicle,” Automatica,” vol. 34, no. 2, pp. 193–198, 1998.

[2] T. I. Fossen and S. I. Sagatun Adaptive control of nonlinear systems, “A case study of underwater robotic systems,” Journal of Robotic Systems, pp. 392-342, 1991

[3] L. Hsu, R. R. Costa, F. Lizarrade, and J. P. V. Soares da Cunha, “Dynamic positioning of remotely operated underwater vehicles,” IEEE Robotics and Automation Magazine, vol. 7, no. 3, pp. 21–31, September 2000.

[4] G. Antonelli, F. Caccavale, S. Chiaverini, and G. Fusco, “A novel adaptive control law for underwater vehicles,” IEEE Transaction on Control Systems Technology, vol. 11, no. 2, pp. 221–232, March 2003.

[5] Zhao and J. Yuh, “Experimental study on advanced underwater robot control,” IEEE Transactions on Robotics, vol. 21, no. 4, pp. 695–703, August 2005.

[6] Dana R. Yoerger and Jean-Jacques E. Slotine, “Robust trajectory control of underwater vehicles,” IEEE Journal of Oceanic Engineering, vol. OE-10, no.4, 1985.

[7] N. Olgac, B. E. Platin, J-M Chang, “Sliding-mode control of remotely operated vehicles for horizontal plane motions,” Control Theory and Applications, IEE Proceedings, vol.138, no. 5, pp. 469-473, 1991.

[8] Jos C. Paulo V. S. da Cunha, Ramon R. Costa, and Liu Hsu, “Design of a High Performance Variable Structure Position Control of ROV’s,” IEEE Journal of Oceanic Engineering, vol. 20, no.1, 1995.

[9] S. Soylu, B. J. Buckham and R. P. Podhorodeski, “A chattering-free sliding-mode controller for underwater vehicles with fault-tolerant infinity-norm thrust allocation,” Ocean Engineering, vol 35, no 16, pp. 1647-1659, 2008.

[10] Dai Ju, Zhao Xiaoguang, Tan Min, “Fuzzy logic control in autonomous ROV navigation,” In Proceedings of 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering,” vol.3 , pp. 1566-1569, 2002.

[11] W. J. Pepijn, F. Colin van de Ven, T. Daniel, “Neural network control of underwater vehicles,” Engineering Applications of Artificial Intelligence, vol.18, pp. 533–547, 2005.

[12] V. Kodogiannis, “Direct adaptive control of underwater vehicles using neural networks,” Journal of Vibration and Control, vol. 9, pp. 605–619, 2003.

[13] Soylu, S., Buckham, B. J. and Podhorodeski, R. P., “MIMO sliding-mode and H∞ controller design for dynamic coupling reduction in underwater-manipulator systems,” CSME Transactions, vol. 33, no. 4, pp. 731-743, 2009.

[14] A. Trebi-Ollennu, B.A. Stacey, and B.A White, “A multivariable decoupling design of an ROV depth control system: a direct adaptive fuzzy SMC approach,” Proceedings of the American Control Conference, vol. 5, pp. 3244-3248, 1995.

[15] H. Liu; Z. Gong; M. Li, “Sliding mode control of ROV based on RBF neural networks adaptive learning,” Proceedings of 3rd International Conference on Intelligent System and Knowledge Engineering, vol.1, pp. 590-594, 2008.

[16] M. Caccia and G. Veruggio, “Guidance and control of a reconfigurable unmanned underwater vehicle,” Control Engineering Practice, vol. 8, pp. 21-37, 2000.

[17] M. W. S. Lau, S. S. M. Swei, G. Seet, E. Low and P. L. Cheng, “Control of an underactuated remotely operated underwater vehicle,” Proceeding of the Institution of Mechanical Engineer Part 1: Journal of Systems and Control Engineering, vol. 217, 2003.

[18] M. Aicardi, A. Caiti, G. Cannata, and G. Casalino, “ Stability and robustness analysis of a two-layered hierarchical architecture for the closed loop control of robots in operational space” Proceedings of 1995 IEEE International Conference on Robotics and Automation, vol. 3, pp. 2771-2278, Nagoya, Japan.

[19] J. Zand, “Enhanced Navigation and Tether Management of Inspection Class ROVs.,” Master’s. Thesis, University of Victoria, 2010.

[20] T. Fossen, Guidance and Control of Ocean Vehicles, Wiley, 1994.

[21] F. L. Lewis, K, Liu, and A. Yesildirek, “Neural net controller with guaranteed tracking performance," IEEE Transactions on Neural Networks,” vol. 6, no. 3, 1995, pp. 703-715, 1995.

[22] R. Kalman, “A new approach to linear filtering and prediction problems,” Journal of Basic Engineering, pp. 35-36, March 1960.

[23] H. Cox, “On the estimation of state variables and parameters for noisy dynamic systems,” IEEE Transactions on Automatic Control, vol. AC-9, pp. 5-12, Jan 1964.

[24] D. Steinke and B. Buckham, “A Kalman filter for the navigation of remotely operated vehicles,” in OCEANS 2005 (IEEE Cat. No. 05CH37711), vol. 1, (Piscataway, NJ, USA), pp. 581–8, IEEE, 2005.

[25] Soylu, S., Buckham, B. J. and Podhorodeski, “Integrated teleoperation scheme for underwater mobile manipulator,” To be submitted to IEEE Journal of Oceanic Engineering.

[26] Slotine, J.J., Li, W., 1991. Applied Nonlinear Control. Prentice-Hall, Englewood Cliffs, NJ.

[27] D. R. Yoerger, J. G. Cooke, and J. J. E. Slotine, “The influence of thruster dynamics on underwater vehicle behaviour and their incorporation into control system design,” IEEE Journal of Oceanic Engineering, vol. 15, pp. 167-177, 1990.

[28] M. Caccia and G. Veruggio, “Modeling and identification of open-frame variable configuration unmanned underwater vehicles,” IEEE Journal of Oceanic Engineering, vol. 25, no.2, 2000.

Appendix E. Dexterous Task-Priority Based Redundancy Resolution for Underwater Manipulator Systems

(2009, CSME Transactions, volume 31, issue 4, pages 519-533)

DEXTEROUS TASK-PRIORITY BASED REDUNDANCY RESOLUTION FOR UNDERWATER MANIPULATOR SYSTEMS

Serdar Soylu, Bradley J. Buckham*, and Ron P. Podhorodeski
Department of Mechanical Engineering,
University of Victoria, P.O. Box 3055, Victoria, B.C., Canada, V8W 3P6
{serdar, bbuckham, podhoro}@me.uvic.ca
*Corresponding author: Fax: (250) 721-6035

Received June 2007, Accepted November 2007
No. 07-CSME-61, E.I.C. Accession 3030

Abstract

The problem of redundancy resolution for underwater remote vehicle-manipulator systems (**URVM**) is addressed in the current work. In URVM applications, it is beneficial to have the underwater remote vehicle (**URV**) hold station using its thrusters while a human pilot operates the serial manipulator. This provides a stable platform for the manipulator and eases the pilot's job drastically when current and/or tether disturbances are present. However, when following this objective, the redundancy of the URVM as a whole is wasted; the four actively controlled motions of the URV are not used to improve the efficacy of the manipulator task. In fact, this standard operating procedure frequently puts the manipulator into near singular configurations. This is not desirable from the manipulator controller standpoint since near singular configurations result in undesirably high joint velocities and oscillations. In this work, a new heuristic approach based on the task-priority redundancy resolution scheme is applied to the URVM. The proposed approach provides a means to avoid singular configurations of the manipulator, and provides dexterous manipulation by using the URV's mobility in an optimal, coordinated manner. This scheme is particularly useful for remote systems where an a priori trajectory generator is not applicable. Numerical case studies are developed to demonstrate the effectiveness of the technique.

Résumé

RÉSOLUTION REDONDANTE BASÉE SUR LA PRIORITÉ POUR LES SYSTÈMES DE MANIPULATEURS SOUS-MARINS POUR TÂCHES DEXTRES

Le problème de la résolution redondante pour les systèmes de véhicule sous-marin manipulateurs télé-commandés (VSMT) est résolu dans le travail présent. Dans les aplications des VSMT, il est bénéfique que le véhicule sous-marin télé-commandé (VST) maintienne sa position avec ses propulseurs tandis qu'un pilote humain actionne le manipulateur sériel. Cela fournit une plate-forme stable pour le manipulateur et facilite radicalement la tâche du pilote en présence de perturbations dues au courault et/ou au câble d'attache. Cependant, en poursuivant cet objectif, la redondance du VSMT est gâchée: les quatres déplacements activement controllés par le VST ne sont pas utilisés pour augmenter l'efficacité de la tâche manipulateur. En fait, cette procédure habituelle d'opération place fréquemment le manipulateur en configurations quasi-singulières. Cela n'est pas souhaitable du point de vue du opérateur du manipulateur car lorsque le système est proche d'une configuration singulière, des vitesses articulain élevées et des oscillations involontaires se produisent. Dans ce travail, une nouvelle approche heuristique basée sur le schéma de résolution redondante des priorités de tâchesest appliquée au VSMT afin de concilier différents objectifs avec les degrés de liberté collectifs du VSMT. La méthode permet d'augmenter la dextérité de la composante manipulateur du VSMT, en utilisant les déplacements du VST pour améliorer le positionnement du manipulateur, tout en maitenant les déplacements du VST à un minimum. Des études de cas numériques sont développées pour présenter l'efficacité de la technique proposée.

1 INTRODUCTION

Underwater remote vehicles (**URVs)** equipped with robotic manipulators play an important role in a number of shallow and deep-water missions for marine science, oil and gas extraction, exploration and salvage [1]. In these applications, the URV is used as a mobile platform that delivers the robotic manipulator to a subsea work site. The motions of the URV and the manipulator are guided independently by a human pilot on a surface support vessel through a long slender tether that provides power and telemetry. For detailed surveys, the URV motion can be accomplished by an on-board URV controller. These controllers use URV state feedback provided by acoustic and inertial positioning systems [2], and dynamic models to intelligently control the conventional thrusters arranged on the URV chassis. Manipulator units are generally add-on technologies produced by independent manufacturers, and hence the manipulator most often has an independent control system. The desired manipulator joint motions are created using a teleoperated master-slave arm configuration. Driving the passive master-arm, the human pilot sets the desired end-effector position and orientation that is to be duplicated by the submerged slave arm, provided the URV can hold station.

However, during the URV manoeuvre, the pilot encounters enormous difficulties. The URV thrusters rely on momentum transfer to a fluid and have an inherent lag in their response to pilot inputs. Furthermore, URV's are bluff bodies designed for omnidirectional operation and have poor drag characteristics, which slow the response of the URV to the pilot's command. As such, the manipulator joints are relied upon for detailed interaction with the subsea environment, while the URV thrusters are generally used to try and counter tether and current disturbances. However, this URV functionality is compromised by the limited visual and navigational feedback available to the pilot, and the subsequent inability to sense disturbances being exerted by the tether and the current. Furthermore, when the movement is replicated by the slave arm, the inertial and hydrodynamic drag associated with the manipulator links create reactions at the manipulator-URV junction. The reaction loads act as disturbances to the URV motion which in turn disturbs the placement of the end-effector [1-4]. These factors make it very difficult for an individual to synchronize the thrusters and manipulator commands, and often two pilots will work together during deployment.

It is proposed that the URV and the manipulator motion be coordinated such that a consolidated controller amalgamates URV navigation data, manipulator state feedback, and a single pilot command to achieve a desired end-effector motion. The consolidated system is referred to as an underwater remote vehicle-manipulator (**URVM**). By coordinating the collective degrees of freedom of the URVM in response to a single pilot input, the operator would only be concerned with driving the end-effector as opposed to operating the end-effector and the URV separately. As a consequence of this collective mode of operation, the URVM efficacy would be greatly improved, and the scope of the detailed operations required by large-scale projects such as VENUS [5] and NEPTUNE [6], which involves setting up a cable-linked seafloor observatory, would become possible.

In addition to the manipulator revolute joints, the URV itself contributes four active degrees of freedom, including surge (forward), sway (lateral) and heave (vertical) translations, and a yawing rotation about the vertical axis. Also, the URV is a free-floating body, and thus moves in two additional rotations: pitch and roll. However, these motions are not controlled, but rather are mitigated by a strong buoyant restoring moment. Due to its kinematic redundancy, the URVM system admits an infinite number of joint-space solutions for a given end-effector position and orientation. This allows one to use the available redundancy to achieve additional objectives besides the given end-effector task using a redundancy resolution technique. The primary objective is to coordinate the URV and the manipulator motion for a given end-effector path, and eliminate the need for the pilot intervention in the URV motion. In addition to that, various optimization criteria (e.g., reduction of fuel consumption, increase of system

manipulability), can be fulfilled by using those degrees of freedom not needed in the satisfaction of the primary objective.

In URVM applications, there are additional constraints not present in most land-based redundant manipulators. These constraints are mostly due to the dynamics of the URV: it is not always desirable to use the URV surge, sway, heave, and yaw motions extensively in placing the end-effector for several reasons. Firstly, the small URV movements cause significant changes in the end-effector location since the small rotational URV motions are amplified by the link lengths. Secondly, the URV consumes relatively more energy than the manipulator for a given motion due to its larger inertia [3, 7, 8]. The consideration of the large inertia of the URV is important since the redundancy resolution scheme could ask the URV to move in a manner that is not possible. Thirdly, the control of the URV is relatively more difficult since its response time is much slower [3]. For these reasons, approximately 35 per cent of the total work done in a typical subsea task requires keeping the URV stationary [9].

The implementations of redundancy resolution methods to the URVM systems have been documented in only a few existing works [7]. The singularity robust task-priority redundancy resolution [10], which was originally proposed in [11] and [12], was shown to be useful for a URVM in [8] due to its multitask capabilities. In [8], the secondary task was set to be holding the vehicle stationary, and the manipulator singularity avoidance was realized by constraining joint motion ranges. However, this limits the available workspace of the manipulator. In [13], the kinematic redundancy is utilized to minimize the total hydrodynamic drag forces experienced by a URVM system in an effort to reduce the energy consumption. However, the different dynamic characteristics of the URV and the manipulator were not addressed. In [7], the singularity robust task-priority redundancy resolution is merged with a fuzzy technique to resolve the URV-manipulator coordination. It was shown that the fuzzy method provides a versatile tool to handle multiple secondary tasks. However, the proposed scheme does not provide a means to hold the vehicle motionless when the commanded end-effector location is within the reach of the manipulator. A screw theory method, combined with the Davies method to represent the differential motion, was implemented to solve the inverse kinematics of URVM systems in [14]. The minimization of the URV motion was realized by imposing kinematic constraints on velocity magnitudes. However, the singularity problem was not addressed in the same paper. The problem of eliminating unnecessary URV motions while avoiding manipulator singular configurations was addressed in [15], where a fuzzy hybrid system was proposed as a solution to this problem. However, the method proposed in [15] requires complex fuzzy rules for guiding the URV motion in a manner consistent with the predefined “hysteresis” behaviour.

Among the existing redundancy resolution schemes for the URVM systems, the task-priority approach is prominent. However, the direct implementation of the task-priority approach presented in prior works consistently forces the manipulator to work in, or near, degenerate configurations resulting in undesirably high joint velocities and oscillations. The solution to this problem requires either complex fuzzy rules or a priori trajectory knowledge [7, 8, 15]. Also, unlike a redundant land-based manipulator, the redundancy of the URVM can not be relied upon when the manipulator hits degeneracy: the response time of the URV motions may not meet that of the subsequent pilot end-effector command. In addition, if the URVM finishes a task in a singular configuration such as when the manipulator is stretched out, and the next task is to lift a heavy object, it may be impossible to lift up the material without exceeding the actuator torque limit, or the righting moment provided by the URV buoyancy module. Furthermore, when the manipulator is at a singular configuration and the URV is held fixed, arbitrary motion of the end-effector may not be possible due to the loss of degree(s) of freedom. To circumvent these potential problems, the URV mobility can be freed to provide only the slow motion of the manipulator base in an attempt to increase the dexterity of the manipulator, and move away from singular configurations.

In order to solve the problem of eliminating unnecessary URV motions while avoiding manipulator singular configurations in URVM systems, a new heuristic approach is presented. The proposed approach is a variation of the task-priority redundancy resolution. In this method, low priority secondary tasks

attempt to exploit slow URV motions and an increase in the system manipulability is sought when necessary. An on-line solution for the occurrence of algorithmic and kinematic singularities is used. To this end, a mobility index that is used to gauge proximity to singular configurations is used. The mobility index supervises the URV movement and its coordination with the end-effector movement. This scheme is particularly useful for remote systems where an a priori trajectory generator is not applicable.

2 THEORETICAL BACKGROUND

2.1 Forward Velocity Problem

The task-space velocity vector $\dot{\mathbf{x}} \in \mathbf{R}^m$ and joint-space velocity vector $\dot{\mathbf{q}} \in \mathbf{R}^n$ are related by:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} \tag{1}$$

where $\mathbf{J} \in \mathbf{R}^{m \times n}$ is the Jacobian matrix. For a kinematically redundant manipulator, $(n > m)$ and there are an infinity set of joint rates solution that can complete the desired end-effector motion, $\dot{\mathbf{x}}$. The existing solution procedures for Eq. (1) apply additional constraints in order to create a deterministic set of equations. The methods are distinguished by the influence of these constraint equations.

2.2 Pseudo Inverse Solution

The minimum-norm solution to the inverse kinematics problem associated with Eq. (1) that satisfies $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$ and minimizes $\|\dot{\mathbf{q}}\|_2 = \sqrt{\dot{\mathbf{q}}^T \dot{\mathbf{q}}}$ is given by:

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{x}} \tag{2}$$

where $\mathbf{J}^{\dagger} = \mathbf{J}^T \left(\mathbf{J}\mathbf{J}^T\right)^{-1}$ is the right Moore-Penrose pseudoinverse [16]. However, this solution does not guarantee singularity avoidance.

2.3 Projected Gradient Method

A general solution to Eq. (1), that sacrifices the minimum norm properties, can be obtained by adding a null space solution to the minimum-norm solution [16]:

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{x}} + \left(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}\right)\dot{\mathbf{q}}_0 \tag{3}$$

where $\dot{\mathbf{q}}_0 \in \mathbf{R}^n$ is an arbitrary joint velocity vector. The term $\left(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}\right)$ is called the projection operator, and it projects the vector $\dot{\mathbf{q}}_0$ onto the null space of the end effector Jacobian matrix. The resulting null-space velocities generate motions within the serial manipulator that do not produce any end-effector motion. Thus, these "internal" motions can be exploited to achieve additional objectives. Liegeois [17] proposed that the arbitrary vector $\dot{\mathbf{q}}_0$ be the gradient of a scalar objective (potential) function $h(\mathbf{q})$:

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{x}} + \left(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}\right)\left(\lambda \nabla h(\mathbf{q})\right) \tag{4}$$

where negative values of the scalar gain λ minimize $h(\mathbf{q})$, and positive values maximize this objective function. Equation (4) is called the projected gradient method.

2.4 Task Priority Redundancy Resolution

The task priority redundancy resolution technique attempts to divide a required task into subtasks according to the order of priority [11, 12]. To resolve conflicting directives from the multiple tasks, a hierarchy is established such that subtasks with lower priority are realized using extra degrees of freedom that are not taken by higher priority subtasks. When conflicts between tasks arise, the solution that is in favour of the higher priority task is realized.

The task priority redundancy resolution technique can be viewed as a variation of the projection gradient method. Instead of projecting the gradient of a scalar objective function through the projection

operator, lower priority tasks are sequentially projected onto the null space of higher priority tasks. For the sake of simplicity, consider a double-task case in which the primary task $\dot{\mathbf{x}}_p \in \mathbf{R}^{m_1}$ has high priority, and the secondary task $\dot{\mathbf{x}}_s \in \mathbf{R}^{m_2}$ has low priority.

$$\dot{\mathbf{x}}_p = \mathbf{J}_p \dot{\mathbf{q}} \tag{5}$$

$$\dot{\mathbf{x}}_s = \mathbf{J}_s \dot{\mathbf{q}} \tag{6}$$

where $\mathbf{J}_p \in \mathbf{R}^{m_1 \times n}$ and $\mathbf{J}_s \in \mathbf{R}^{m_2 \times n}$ are the primary-task Jacobian matrix and the secondary task Jacobian matrix, respectively. The general solution of Eq. (5) using the pseudoinverse is as follows:

$$\dot{\mathbf{q}} = \mathbf{J}_p \dot{\mathbf{x}}_p + \left(\mathbf{I} - \mathbf{J}_p^{\dagger}\mathbf{J}_p\right)\dot{\mathbf{q}}_0 \tag{7}$$

Substituting Eq. (7) into the secondary task forward velocity kinematics in Eq. (6) yields:

$$\mathbf{J}_s\left(\mathbf{I} - \mathbf{J}_p^{\dagger}\mathbf{J}_p\right)\dot{\mathbf{q}}_0 = \dot{\mathbf{x}}_s - \mathbf{J}_s\mathbf{J}_p^{\dagger}\dot{\mathbf{x}}_p \tag{8}$$

Then the unknown $\dot{\mathbf{q}}_0$ that minimizes $\left\|\dot{\mathbf{x}}_s - \mathbf{J}_s\dot{\mathbf{q}}\right\|_2$ is given by:

$$\dot{\mathbf{q}}_0 = \bar{\mathbf{J}}_s^{\dagger}\left(\dot{\mathbf{x}}_s - \mathbf{J}_s\mathbf{J}_p^{\dagger}\dot{\mathbf{x}}_p\right) \tag{9}$$

where $\bar{\mathbf{J}}_s = \left(\mathbf{J}_s\left(\mathbf{I} - \mathbf{J}_p^{\dagger}\mathbf{J}_p\right)\right)$. The joint space velocities are obtained by substituting Eq. (9) into Eq. (7). Exploiting the fact that the null-space projection operator is Hermitian and idempotent [11-12], $\dot{\mathbf{q}}$ is found as:

$$\dot{\mathbf{q}} = \mathbf{J}_p^{\dagger}\dot{\mathbf{x}}_p + \left(\mathbf{I} - \mathbf{J}_p^{\dagger}\mathbf{J}_p\right)^{\dagger}\bar{\mathbf{J}}_s^{\dagger}\left(\dot{\mathbf{x}}_s - \mathbf{J}_s\mathbf{J}_p^{\dagger}\dot{\mathbf{x}}_p\right) \tag{10}$$

However, Eq. (10) is vulnerable to the occurrence of kinematic and algorithmic singularities. A kinematic singularity corresponds to a loss in manipulator degree of freedom and a drop in the level of redundancy in the system. The primary Jacobian pseudoinverse is given by:

$$\mathbf{J}_p^{\dagger} = \sum_{i=1}^{r}\mathbf{v}_i\mathbf{u}_i^T/\sigma_i \tag{11}$$

where r is the rank of the primary Jacobian, and $\mathbf{v}_i$ and $\mathbf{u}_i$ are the right and left singular vectors of $\mathbf{J}_p$, respectively [16]. Hence, one can detect the occurrence of kinematic singularity by monitoring the corresponding singular value σ_r. Given that $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r \geq 0$, at the singular configuration, σ_r becomes zero. This means that the joint-space velocities along $\mathbf{v_r}$ fall in the null-space of $\mathbf{J}_p$, and as a result the end-effector velocities along $\mathbf{u}_r$ become unrealizable [10]. Note that there could be cases where more than one singular value is zero. This situation corresponds to a further reduction in the range space of the primary Jacobian.

An algorithmic singularity occurs when the matrix $\bar{\mathbf{J}}_s = \left(\mathbf{J}_s\left(\mathbf{I} - \mathbf{J}_p^{\dagger}\mathbf{J}_p\right)\right)$ becomes rank deficient. At an algorithmic singularity, the null space of the primary task and the secondary task are linearly dependent, i.e., $N\left(\mathbf{J}_p\right) \cap N\left(\mathbf{J}_s\right) \neq 0$, reflecting the fact that the primary and secondary tasks can not be satisfied simultaneously. In this case, high joint velocities and oscillations occur [10]. In order to circumvent the algorithmic singularity problem, the components of the secondary task solution that conflict with the primary task solution must be removed.

2.5 Singularity Robust Task Priority Redundancy Solution

To alleviate the kinematic singularity problem in Eq. (10), the damped-least squares inverse given as $\mathbf{J}_p^* = \sum_{i=1}^{r}\left(\sigma_i/(\sigma_i^2+\eta^2)\right)\mathbf{v}_i\mathbf{u}_i^T$ is used in lieu of the pseudoinverse [18]. A non-null damping factor $\eta \in \mathbf{R}$ prevents the denominator from becoming zero. Thus, it provides continuity and good conditioning to the solution, but this improvement is obtained at the expense of an increased residual error.

To alleviate algorithmic singularities, the general solution for the primary task, Eq. (7), can be equalized to the minimum-norm solution of the secondary task, $\mathbf{J}_s{}^{\dagger}\dot{\mathbf{x}}_s$, to obtain a solution that solves both the primary and secondary task simultaneously [10]:

$$\mathbf{J}_s{}^{\dagger}\dot{\mathbf{x}}_s = \mathbf{J}_p{}^{\dagger}\dot{\mathbf{x}}_p + \left(\mathbf{I} - \mathbf{J}_p^{\dagger}\mathbf{J}_p\right)\dot{\mathbf{q}}_0 \tag{12}$$

Eq. (12) can be solved for $\dot{\mathbf{q}}_0$ yielding:

$$\dot{\mathbf{q}}_0 = \left(\mathbf{I} - \mathbf{J}_p^{\dagger}\mathbf{J}_p\right)^{\dagger}\left(\mathbf{J}_s{}^{\dagger}\dot{\mathbf{x}}_s - \mathbf{J}_p{}^{\dagger}\dot{\mathbf{x}}_p\right) \tag{13}$$

Since $\left(\mathbf{I} - \mathbf{J}_p^{\dagger}\mathbf{J}_p\right)^{\dagger} = \left(\mathbf{I} - \mathbf{J}_p^{\dagger}\mathbf{J}_p\right)$ and $\mathbf{J}_p^{\dagger} = \mathbf{J}_p^{\dagger}\mathbf{J}_p\mathbf{J}_p^{\dagger}$, Eq. (13) can be simplified to:

$$\dot{\mathbf{q}}_0 = \left(\mathbf{I} - \mathbf{J}_p^{\dagger}\mathbf{J}_p\right)\mathbf{J}_s^{\dagger}\dot{\mathbf{x}}_s \tag{14}$$

Substituting Eq. (14) into Eq. (7) and using the idempotence of $\left(\mathbf{I} - \mathbf{J}_p^{\dagger}\mathbf{J}_p\right)$ yields:

$$\dot{\mathbf{q}} = \mathbf{J}_p^{\dagger}\dot{\mathbf{x}}_p + \left(\mathbf{I} - \mathbf{J}_p^{\dagger}\mathbf{J}_p\right)\mathbf{J}_s^{\dagger}\dot{\mathbf{x}}_s \tag{15}$$

In Eq. (15) the minimum-norm solutions to the primary and secondary inverse kinematic problems, Eq. (5) and Eq. (6) respectively, are obtained using the pseudoinverse of the corresponding Jacobian. The joint velocities computed for the secondary task are then projected onto the null space of the primary task to remove components in conflict with the primary task. This provides robustness in the presence of algorithmic singularities. To incorporate the robustness to the kinematic singularities, $\mathbf{J}^*$ is used instead of $\mathbf{J}^{\dagger}$ in Eq. (15). However, in URVM applications, $\mathbf{J}_p^{\dagger}$ is used as it is, since the primary Jacobian of the URVM system will always exhibit full rank due to the mobility of the URV [7]. The extension of Eq. (15) to highly redundant systems performing more than two tasks is presented in [19].

The direct implementation of Eq. (15) is prone to a numerical drift occurring when the joint rates are integrated forward in time to obtain URV/manipulator position values. In order to avoid this problem, the closed-loop version of Eq. (15) can be employed [7]:

$$\dot{\mathbf{q}} = \mathbf{J}_p^{\dagger}\left(\dot{\mathbf{x}}_p + \mathbf{K}_P\mathbf{e}_P\right) + \left(\mathbf{I} - \mathbf{J}_p^{\dagger}\mathbf{J}_p\right)\mathbf{J}_s^{\dagger}\left(\dot{\mathbf{x}}_s + \mathbf{K}_s\mathbf{e}_s\right) \tag{16}$$

where $\mathbf{K}_p$ and $\mathbf{K}_s$ are user-defined positive definite matrix gains, and $\mathbf{e}_p$ and $\mathbf{e}_s$ are the numerical construction errors defined as $\mathbf{x}_{p,d} - \mathbf{x}_p$ and $\mathbf{x}_{s,d} - \mathbf{x}_p$ with *d* denoting the desired values.

3 DEXTEROUS TASK PRIORITY APPROACH

3.1 Overview

It was shown in [8] that the direct implementation of the singularity robust task priority approach might drive the manipulator component into its singular configurations when the secondary task is defined as station keeping of the URV; it fails to coordinate the URV and the manipulator motion in an effective manner. Therefore, prevention of manipulator singularities must be included in the framework of the redundancy resolution method. As such, the functionality of the robust-task priority approach in accommodating multiple kinematic tasks can be fully utilized. In the following sections, the method

proposed to circumvent the singularity problem will be presented. The method provides a means to coordinate the URV and the manipulator motion in such a way that the manipulator singularity is avoided.

3.2 Mobility of URVM Systems

There is a strong connection between the kinematic singularities of the sole manipulator and the algorithmic singularities of the URVM system for the current application. The algorithmic singularities occur when $\mathbf{J}_p$ and $\mathbf{J}_s$ have common linearly dependent columns meaning the primary and secondary tasks are in conflict [20]. Given that the secondary task is to keep the URV at the current location, any conflict indicates that the end-effector can not be placed at a desired location without using the URV's mobility. This happens when a loss in the degree of freedom of the manipulator, a kinematic singularity, is experienced during the URV station keeping. In an effort to monitor the kinematic singularity of the sole manipulator, it is proposed that the collective Jacobian matrix $\mathbf{J}_a = \begin{bmatrix} \mathbf{J}_p^T & \mathbf{J}_s^T \end{bmatrix}^T \in \mathbf{R}^{(m_1+m_2)\times n}$ be used. The rank deficiency of $\mathbf{J}_a$ indicates a conflict in the primary and the secondary task since $\mathbf{J}_p$ and $\mathbf{J}_s$ always have full rank. Consequently, the rank of $\mathbf{J}_a$ can be used as the indicator of the sole manipulator singularity.

In an attempt to track manipulator singularity, it is useful to use a parameter that quantifies the closeness to the manipulator singular configurations. The following index based on Yoshikawa's measure of manipulability [21] can be used for that purpose:

$$\rho = \sqrt{\det\left(\mathbf{J}_a \mathbf{J}_a^T\right)} \tag{17}$$

Differently from Yoshikawa's measure of manipulability where the end-effector Jacobian matrix is used, the collective matrix $\mathbf{J}_a$ is used in Eq. (17) instead. To distinguish this difference, Eq. (17) will be called mobility index in the subsequent sections. With this arrangement, Eq. (17) can be used not only to gauge the level of proximity of the manipulator to the singular configurations, but also to specify the extent to which the URV should move to prevent the manipulator from hitting its near singular configurations. Using the Singular Value Decomposition (**SVD**) of the Jacobian matrix, $\mathbf{J}_a = \mathbf{U}\sum\mathbf{V}^T$, it can be shown that Eq. (17) is merely the product of the singular values, $\rho = \prod_{i=1}^{m} |\sigma_i|$ [21]. Given that $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_n \geq 0$ always holds, the mobility index is lower and upper bounded by $\sigma_n^n \leq \rho \leq \sigma_1^n$; leading to $\rho/\sigma_1^n \leq 1$. As such, the mobility index can be defined as ρ/σ_1^n for cases in which the normalization of the mobility index is necessary. Since the mobility index is dependent on the physical size of the mechanism, this normalization would be necessary if the proposed approach is used on different systems for a comparison purpose. The mobility index becomes zero only when the collective Jacobian matrix is not full rank.

In order to monitor how the mobility index changes with respect to the joint vector, the following analytical formula can be used [22]:

$$\frac{\partial \rho}{\partial q_i} = \rho \cdot trace\left(\frac{\partial \mathbf{J}_a}{\partial q_i}\mathbf{J}_a^{\dagger}\right), \quad i = 1 \ldots n \tag{18}$$

3.3 Mobility Index Based Dexterous Redundancy Resolution

The following redundancy resolution is proposed with $\gamma = \left(1 - sign\left(d\rho/dt\right)\right)/2$:

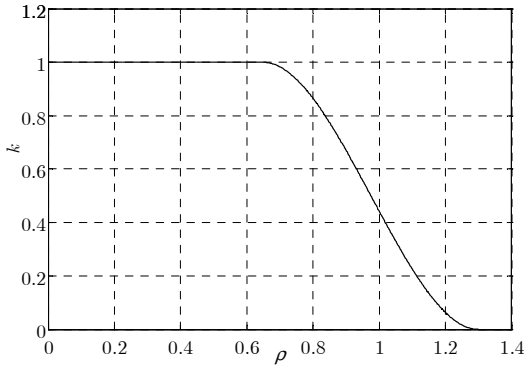
$$\dot{\mathbf{q}} = \mathbf{J}_{w,p}^{\dagger}\left(\dot{\mathbf{x}}_p + \mathbf{K}_P\mathbf{e}_P\right) + \left(\mathbf{I} - \mathbf{J}_{w,p}^{\dagger}\mathbf{J}_p\right)\left(\left(1-\gamma k\right)\mathbf{J}_{w,s}^{\dagger}\left(\dot{\mathbf{x}}_s + \mathbf{K}_s\mathbf{e}_s\right) + \gamma k\left(\lambda\mathbf{W}^{-1}\frac{\partial\rho\left(\mathbf{J}_a\right)}{\partial\mathbf{q}}\right)\right) \tag{19}$$

which combines the gradient projection method given in Section 2.3 and the task-priority approach given in Section 2.5. The task-priority component is in charge of realizing minimal URV motion, whereas the gradient projection component is in charge of avoiding manipulator near singular configurations. In Eq. (19), the term $\mathbf{J}_w^{\dagger}$ is the weighted pseudoinverse that instantaneously solves the problem of minimizing $\dot{\mathbf{q}}^T\mathbf{W}\dot{\mathbf{q}}$ subject to $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$ in the form of $\dot{\mathbf{q}} = \mathbf{J}_w^{\dagger}\dot{\mathbf{x}}$:

$$\mathbf{J}_w^{\dagger} = \mathbf{W}^{-1}\mathbf{J}^T\left(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^T\right)^{-1} \tag{20}$$

with the inverse of the weighting matrix being defined as $\mathbf{W}^{-1} = diag\left(w_1, w_2, \ldots, w_n\right)$. This weighting matrix contains weight factors for each degree of freedom in a robotic system. The term λ is a positive constant that determines the convergence rate to the local minimum of the mobility index function. The limiting factors on the value of λ can be found in [23]. The term *k* is the sigmoid form shaping function [24] illustrated in Figure 1.

The success of the Gradient Projection Method relies on the fact that the projection operator $\left(\mathbf{I} - \mathbf{J}_p^{\dagger}\mathbf{J}_p\right)$ is semi-definite [25]. However, when the weighted pseudoinverse is used, the resulting matrix, $\left(\mathbf{I} - \mathbf{J}_{w,p}^{\dagger}\mathbf{J}_p\right)$, is no longer symmetric, and therefore is non-definite. For this reason, the gradient projection method must be modified. This can be done by postmultiplying the weighted null-space projection operator, $\left(\mathbf{I} - \mathbf{J}_{w,p}^{\dagger}\mathbf{J}_p\right)$, by the inverse of the positive-definite weighting matrix, $\mathbf{W}^{-1}$ [22]. The resulting matrix, $\left(\mathbf{I} - \mathbf{J}_{w,p}^{\dagger}\mathbf{J}_p\right)\mathbf{W}^{-1}$, is positive semi-definite. The proof can be found in [26]. The proof requires decomposing the positive definite weighting matrix as $\mathbf{W}^{-1} = \mathbf{U}^T\mathbf{U}$. Therefore, the weighting matrix must be a positive-definite matrix.



$$k = \begin{cases} 0 & \bar{\rho} \le \rho \\ 4\left(\dfrac{4\rho^3 - 9\rho^2\bar{\rho} + 6\bar{\rho}^2\rho - \bar{\rho}^3}{\bar{\rho}^3}\right), & \dfrac{\bar{\rho}}{2} \le \rho < \bar{\rho} \\ 1 & \rho < \dfrac{\bar{\rho}}{2} \end{cases}$$

Figure 1: Shaping function definition (right) and corresponding plot for $\bar{\rho} = 1.3$ (left).

An activation threshold value $\bar{\rho}$, determined based on the desired proximity from singular configurations, is set on the mobility index. As mentioned earlier, Eq. (19) can be thought of as the combination of the gradient projected method and the singularity robust task-priority approach. When activated, it works to move the manipulator away from the singular configurations. The task-priority component works as in Section 2.5, keeping the URV at the current location in an effort to provide a stable platform for the end-effector task.

In order to blend the task priority and projected gradient contributions, a shaping function is used to distribute the secondary task demand over the station-keeping and the singularity avoidance tasks. If the

singularity index is bigger than the designated threshold, k becomes zero, and Eq. (19) becomes equivalent to the task-priority approach of Section 2.4. In this case the URV could be still kept at rest while the manipulator is performing a task; hence, unnecessary URV motion is avoided. In cases where $\bar{\rho}/2 \le \rho < \bar{\rho}$, the shaping function distributes the secondary demand over the URV and the manipulator. When this happens, the URV is moved in proportion to the mobility index in an attempt to keep the manipulator away from a singular configuration. When the mobility index drops below half of the threshold, the emphasis is placed fully on the singularity avoidance. In this case, the URV-related secondary task is fully disregarded, and the available redundancy including the URV's mobility, is committed fully to the manipulator's singularity avoidance. However, the intervention of the projected gradient solution is not necessary when ρ is naturally increasing. As such, a discontinuous switching term, using the *sign* function defined as for $\dot{\rho} \le 0$, $sign(\dot{\rho}) = -1$ and for $\dot{\rho} > 0$, $sign(\dot{\rho}) = 1$, is added such that the singularity avoidance only intervenes in times of worsening manipulability.

In order to commit the full redundancy to the minimal URV motion criterion when the mobility index function is already an increasing function, the term γ must be included in the task-priority part of Eq. (19). If this term is not included, the full capacity of available redundancy can not be utilized by the system since it will be still weighted by the shaping function.

Equation (19) can be executed in real time since the primary and secondary Jacobians always exhibit full rank. As such, the time consuming SVD operation can be avoided when computing the weighted pseudoinverses. By exploiting this property, the analytical formula given in Eq. (20) can be used instead of the full SVD; leading to the suitability of Eq. (19) to real-time URVM applications.

4 SIMULATION

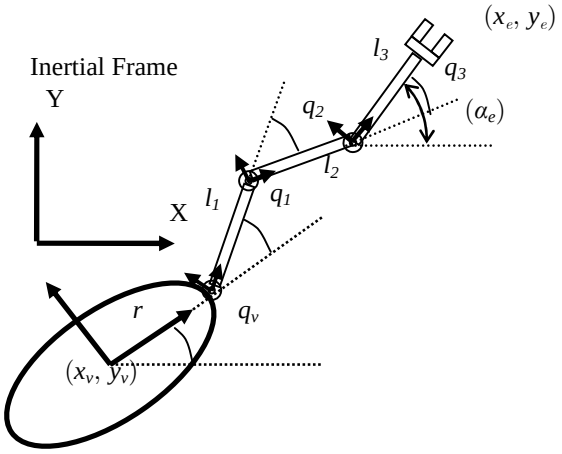


Figure 2: The kinematic chain of the planar underwater vehicle system with a 3 joint-DOF manipulator.

The URV system considered in this work is the Canadian Scientific Submersible Facility's ROPOS equipped with a 3-DOF manipulator as shown in Figure 2. For the sake of simplicity, only a planar end-effector task is considered. The state vector is given as $\mathbf{q} = [x_v \quad y_v \quad q_v \quad q_1 \quad q_2 \quad q_3]^T$, where x_v and y_v are the coordinates of the vehicle's center of mass expressed in the inertial frame, q_v is the yaw angle of the vehicle, and q_i is the joint position with respect to the body-fixed link coordinate system. The terms x_e and y_e are the coordinates of the end-effector, and α_e is the orientation of the end-effector. The distance from the center of mass of the vehicle to the first joint is $r = 1$ m. The distance from one link to another is denoted by l_i, and its values are 1m as well. The primary task is to make the end-effector

follow a predetermined trajectory. Therefore, the corresponding manipulation vector for the primary task is defined as $\mathbf{x}_p = [x_e \quad y_e]^T$ with $\mathbf{J}_p \in \mathbf{R}^{2\times 6}$. A series of waypoints were set for the end-effector and a continuous set of end-effector values were generated using a third order polynomial function with zero initial and final velocities. The system starts from the initial configuration of $\mathbf{q} = [0 \quad 0 \quad 0 \quad \pi/3 \quad -\pi/3 \quad \pi/3]^T$ m, rad. that corresponds to the end-effector position of $\mathbf{x}_p = [3, \quad 1.7321]^T$ m. The final end-effector location is $\mathbf{x}_p = [5, \quad 5.1962]^T$ m.

The secondary task variables are defined as $\mathbf{x}_s = [x_v \quad y_v \quad q_v]^T$ with $\mathbf{J}_s \in \mathbf{R}^{3\times 6}$. Note that this definition always leads to a full rank secondary task Jacobian matrix, $\mathbf{J}_s$. Provided that the primary task can be tracked using only the manipulator joints, it is desired to keep the URV location constant. To this end, the current URV position value is entered as the desired position values for the next step. This guarantees zero-URV motion as long as the desired end-effector location is within the current reach of the manipulator. In addition, this arrangement makes the URV follow the end-effector.

For the current implementation, the inverse of the weight matrix was chosen to be $\mathbf{W}^{-1} = diag(0.1 \quad 0.1 \quad \varepsilon \quad 1 \quad 1 \quad 1)$ with ε being a small positive number such as $\varepsilon \leq 10^{-8}$. Note that the third component of the weighting matrix corresponding to the yaw orientation of the URV is set to nearly zero. This forces the yawing motion of the URV to become zero. This arrangement makes the weighting matrix positive-definite. In the weighting matrix, bigger diagonal values require larger movements, whereas smaller values require smaller movements for the associated degree of freedom. In other words, it sets the motion preferences for each degree of freedom of the URVM system.

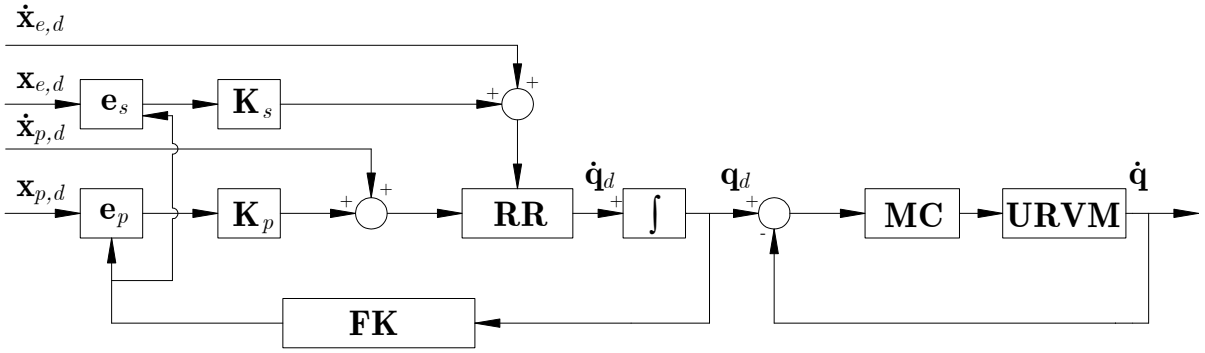


Figure 3: The simulation scheme consisting of the kinematic loop (left) and dynamic loop (right). In the figure, RR, MC, and FK stand for redundancy resolution, motion control and forward kinematics, respectively.

The time-domain simulation scheme that was used is illustrated in Figure 3. In the figure, the closed-loop redundancy resolution, Eq.(19), generates desired joint-space reference values for the dynamic control loop. The dynamic control loop computes the corresponding actuators forces, i.e. the URV thrusts and the manipulator torques, from the requested joint position and velocity values. For the current study, the computed-torque method of [27] is used.

The implementation of the computed torque method requires a dynamic model of the system given as:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{G} = \tau \tag{21}$$

where $\mathbf{M} \in \mathbf{R}^{6\times 6}$ is the inertia matrix including the added mass effects, $\mathbf{C} \in \mathbf{R}^{6\times 6}$ is the matrix of centrifugal and Coriolis terms, $\mathbf{D} \in \mathbf{R}^{6\times 6}$ is the drag matrix, $\mathbf{G} \in \mathbf{R}^{6\times 1}$ is the vector of gravity and buoyancy forces and moments, and finally $\tau \in \mathbf{R}^{6\times 1}$ is the force vector acting on the URVM system. The dynamic parameters for ROPOS are taken from [28]. The computed-torque control law is given as:

$$\tau = \hat{\mathbf{M}}\left(\ddot{\mathbf{q}}_d + \mathbf{K}_v\left(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}\right) + \mathbf{K}_c\left(\mathbf{q}_d - \mathbf{q}\right)\right) + \hat{\xi} \tag{22}$$

where $\hat{\mathbf{M}}$ and $\hat{\xi} = \hat{\mathbf{C}}\dot{\mathbf{q}} + \hat{\mathbf{D}}\dot{\mathbf{q}} + \hat{\mathbf{G}}$ are the estimates of the model parameters, and $\mathbf{K}_v$ and $\mathbf{K}_c$ are the positive gain matrices. Substituting Eq. (22) into Eq. (21) with the assumption of $\hat{\mathbf{M}} = \mathbf{M}$ and $\hat{\xi} = \xi$ yields:

$$\ddot{\mathbf{e}} + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_c\mathbf{e} = \mathbf{0} \tag{23}$$

where "**e**" denotes errors defined as $\ddot{\mathbf{e}} = \ddot{\mathbf{q}}_d - \ddot{\mathbf{q}}, \dot{\mathbf{e}} = \dot{\mathbf{q}}_d - \dot{\mathbf{q}},$ and $\mathbf{e} = \mathbf{q}_d - \mathbf{q}.$ Equation (23) guarantees the asymptotic reduction of the error.

The followed simulation scheme is illustrated in Figure 3. Equation (19) is implemented with λ=10, $\bar{\rho}$=1.3, $\mathbf{K}_p = [10 \quad 10]^T$ and $\mathbf{K}_s = [10 \quad 10 \quad 100]^T$, which are determined by trial and error. The simulation results are reported in Figure 4. The scheme produced joint positions and velocities that are realizable, as they show a smooth pattern, with no jerk. As can be seen from Figure 4, the primary task was successfully executed. The velocities of the state variables became null at the end of the simulation consistently with the desired initial and final velocities.

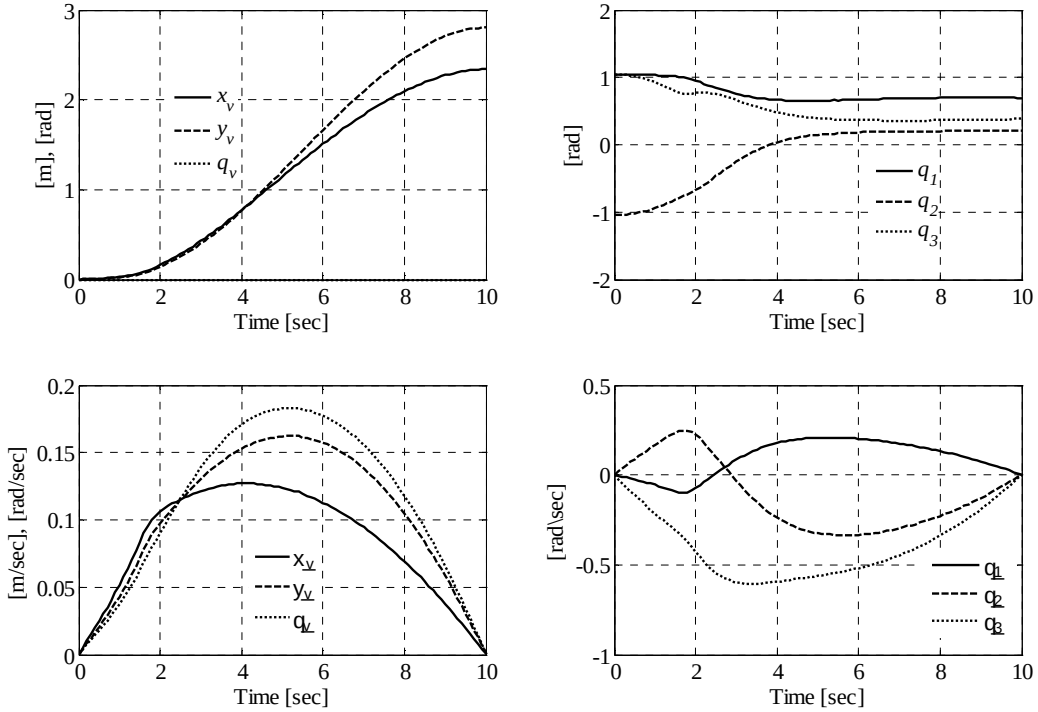


Figure 4: Time history of state variables for the URV (left figures) and the manipulator (right figures).

The robust task-priority approach has also been implemented with the same simulation parameters. The corresponding URVM manoeuvres produced by each method are illustrated in Figure 5. Both approaches successfully fulfilled the primary task. Despite the secondary task constraint, both methods caused the URV to move during the manoeuvre; it indicates that the trajectory goes out of the workspace of the sole manipulator. As can be seen from the left figure in Figure 5, the singularity robust task-priority approach drives the sole manipulator into a singular configuration, i.e., a stretched-out configuration for the current simulation case. On the other hand, the dexterous task-priority scheme does not force the manipulator component to hit its singular straight arm configuration. With the proposed approach, when the manipulator approaches a singular configuration, the system reconfigures itself in a dexterous

configuration using the URV's mobility with a minimal effort. The reconfiguration takes place while the URV provides a stable platform for the manipulator as much as possible; the URV moves along translational directions since it is unavoidable, but the rotational motion is prevented in agreement with the secondary task. Therefore, Eq. (19) provides a means to increase the dexterity of the manipulator by guiding the URV motion in an optimal, coordinated manner.

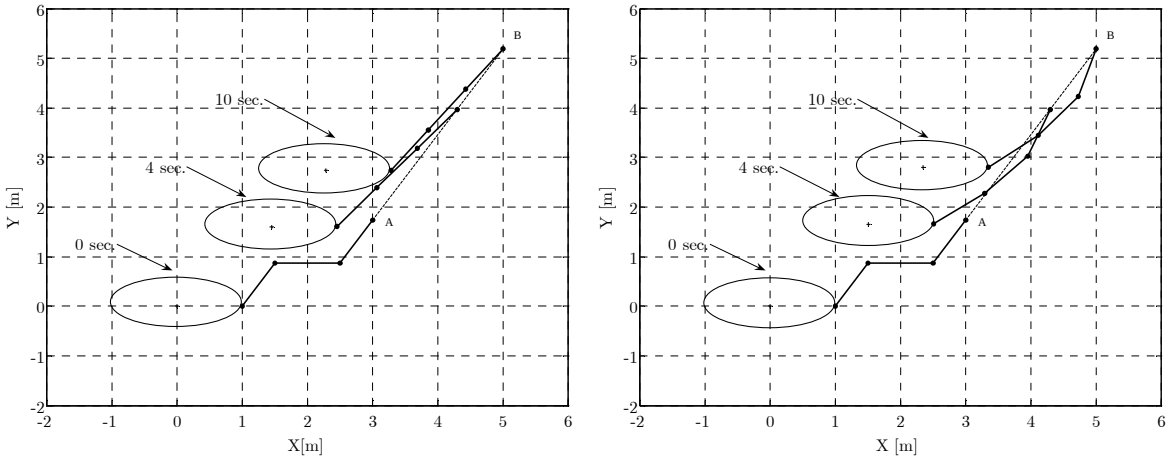


Figure 5: With the direct implementation of the task-priority approach, the manipulator falls into near singular configurations, i.e. a stretched-out configuration (shown on the left). The dexterous task-priority approach prevents the manipulator from falling into near singular configurations (shown on the right) by applying URV motions.

The left figure in Figure 6 reveals that the robust task-priority redundancy resolution, Eq. (16), leads to poor dexterity performance; it falls into near singularity at approximately 5 seconds. The mobility index never reaches the zero value due to the mobility of the URV, but the system dexterity is still very poor. The right figure in Figure 6 is obtained from the implementation of Eq. (19). As the figure reveals, the dexterity of the system is significantly improved. The lower bound $\overline{\rho} = 1.3$ determines the extent to which the system reconfigures itself in an effort to avoid manipulator singularities. This value can be adjusted on-line depending on the specific requirements of a given task.

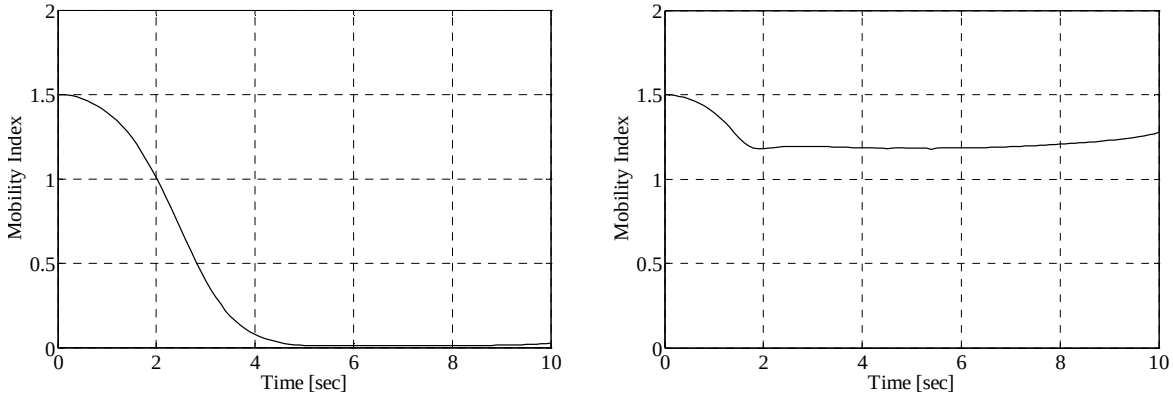


Figure 6: Measure of manipulability for the task priority approach (left) and the task priority approach augmented with manipulability monitoring.

5 DISCUSSION

The success of the model-based controller approach relies heavily upon the accuracy of the dynamic model of the URVM system. Given the nonlinear nature of the underwater environment, having a reliable dynamic model for a URVM system is hard to obtain. By virtue of this, other control strategies such as sliding-mode control, adaptive control, and neural-fuzzy control may be more effective. However, since the main focus of the current work is redundancy resolution, a simple control algorithm is implemented with the assumption of knowing the exact dynamics of the system and having a good response time. The inclusion of the control framework not only provides a complete redundancy resolution scheme, but also demonstrates the capability of the proposed method in generating dynamically realizable joint-space vectors for the URVM system.

The collective Jacobian can be decomposed as:

$$\mathbf{J}_a = \begin{bmatrix} \mathbf{J}_{u,p} & \mathbf{J}_{m,p} \\ \mathbf{J}_{u,s} & \mathbf{J}_{m,s} \end{bmatrix} \tag{24}$$

where $\mathbf{J}_{u,p} \in \mathbf{R}^{m_1 \times n}$ and $\mathbf{J}_{m,p} \in \mathbf{R}^{m_1 \times l}$ are the submatrices containing column vectors defining the contribution to the primary task velocities (end-effector velocities) made by the URV and the manipulator state velocities, respectively. Likewise, $\mathbf{J}_{u,s} \in \mathbf{R}^{m_2 \times u}$ and $\mathbf{J}_{m,s} \in \mathbf{R}^{m_2 \times l}$ are the submatrices containing column vectors defining the contribution to the secondary task velocities made by the URV and the manipulator state velocities with $u + l = n$, respectively. When the manipulator is at a singular configuration, the submatrix $\mathbf{J}_{m,p}$ loses rank, and so does the last l columns of $\mathbf{J}_a$, since $\mathbf{J}_{m,s}$ is a zero matrix, i.e., the manipulator joint velocities do not contribute to the URV velocities. Therefore, manipulator singularities can also be captured by observing the submatrix $\mathbf{J}_{m,p}$ of the primary task Jacobian; this implies that $\mathbf{J}_{m,p}$ can be used in lieu of $\mathbf{J}_a$ in Eq. (17). This choice is computationally more efficient, as it involves smaller size matrix computations. Note that $\mathbf{J}_a$ does not have homogeneous units. Therefore, when the mobility index is normalized according to ρ / σ_1^n, this normalization leads to different units for the index depending on the units associated to σ_1. This problem can be solved by using $\mathbf{J}_{m,p}$ instead of $\mathbf{J}_a$ since $\mathbf{J}_{m,p}$ has homogeneous units. In the current study, this is not necessary since the normalization is not used. As well, Eq. (17) is presented herein since it provides a versatile tool to quantitatively define the degree of conflict between general tasks provided the Jacobians of the defined tasks exhibit full rank. This piece of information has the potential to be used to coordinate different URVM tasks for different URVM applications.

6 CONCLUSION

This work has addressed the problem of kinematic redundancy in URVM systems. It has been shown that the robust task-priority approach is prone to manipulator singularities, when the secondary task is defined as station keeping of the URV. The singular configurations of the manipulator cause the same significant complications for the manipulator controller as land-based non-redundant cases. In this work, a new heuristic method has been proposed to remove these complications. The method provides a means to combine two different redundancy resolution techniques in a novel way. In order to gauge not only the singular configurations but also the need for the URV motion, a mobility index number is proposed that is derived from a modification of Yoshikawa's measure of manipulability. Using the mobility index, the URV motions are applied in an optimal, coordinated manner through a simple shaping function. This mobility index is a versatile tool that can be used to coordinate different URVM tasks. The proposed method removes the complexity associated with existing solutions to the singularity problem of the

URVM system. As well, the method eliminates unnecessary URV motions when the demanded end-effector location is within the reach of the manipulator, and hence provides energy efficient control for the URVM application. The method presented produces kinematically and dynamically realizable URV-manipulator state values that significantly improve the dexterity of the system, thus ensuring dexterous manipulation. This scheme makes the prevailing task-priority approach more suitable, and hence improves the coordinated control of the URVM systems.

ACKNOWLEDGEMENT

The authors wish to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for providing financial support for this research.

REFERENCES

[1] M. W. Dunnigan, and G. T. Russell, "Evaluation and reduction of the dynamic coupling between a manipulator and an underwater vehicle," *IEEE J. Oceanic Eng.*, vol. 23, no. 3, pp. 260-273, July 1998.

[2] G. Grenon, P. An, S. Smith, and A. Healey, "Enhancement of the inertial navigation system for the Morpheus autonomous underwater vehicles," *IEEE J. Oceanic Eng.*, vol. 26 no. 4, pp. 548-560, Oct. 2001.

[3] J. Kim and W. K. Chung, "Dynamic analysis and two-time scale control for underwater vehicle-manipulator systems," *Proc. 2003 IEEE/RSJ Int. Conf. on Intelligent Robot. and Syst., (IROS 2003)*, vol.1, pp. 577–582, 2003.

[4] S. Soylu, B. J. Buckham and R. P. Podhorodeski, "Using articulated-body algorithm within sliding mode control to compensate dynamic coupling in underwater manipulator systems," *CSME Trans.* vol. 29, no. 4, pp. 629-643, 2005.

[5] K. Kawaguchi, H. Momma, and R. Iwase, "VENUS PROJECT-submarine cable recovery system," *1998 Int. Symp. on Underwater Technology (Cat. No.98EX101)*, pp. 448-452, 1998.

[6] J. Delaney, P. Beauchamp, M. McNutt, C. Barnes, A. Chave, and J. Madden, "Project NEPTUNE: an interactive, regional cabled ocean observatory in the Northeast Pacific," *Oceans Conf. Record (IEEE)*, vol. 2, pp. 1038-1042, 2003.

[7] G. Antonelli, and S. Chiaverini, "Fuzzy redundancy resolution and motion coordination for underwater vehicle-manipulator systems, "*IEEE T. Fuzzy Syst.*, vol. 11, no. 1, pp. 109-120, 2003.

[8] G. Antonelli and S. Chiaverini, "Task priority redundancy resolution for underwater-manipulator systems," *1998 IEEE Int. Conf. Robotic. Autom.*, pp. 768-733, May 1998.

[9] K. R. Goheen and R. Jeffreys, "Multivariable self-tuning autopilots for autonomous and remotely operated vehicle, " *IEEE J. Oceanic Eng.*, vol. 15, no. 3, pp. 144-151, July 1990.

[10] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators, "*IEEE T. Robotic. Autom.*, vol. 13, pp. 398-410, June 1997.

[11] A. A. Maciejewski, and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robot. Res.*, vol. 4, no.3, pp. 109-117, 1985.

[12] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 3-15, 1987.

[13] N. Sarkar and T. K. Podder, "Motion coordination of underwater vehicle-manipulator systems subject to drag optimization," *Proc. 1999 IEEE Int. Conf. on Robotic. Autom.*, pp. 387-392, 1999.

[14] C. H. dos Santos, R. Guenther, D. Martins, and E. R. De Pieri, "Virtual kinematic chains to solve the underwater vehicle-manipulator systems redundancy," *J. of the Brazilian Society of Mechanical Sciences and Eng.*, vol. 28, no. 3, pp. 354-361, 2006.

[15] C. H. dos Santos, G. Bittencourt, and R. Guenther, "Motion coordination for underwater vehicle-manipulator systems using a fuzzy hybrid strategy," *Int. Conf. on Intell. Robot. and Syst.*, Beijing, China, pp. 3018-3023, Oct. 2006.
[16] D. S. Watkins, *Fundamentals of Matrix Computations*, 2nd Edition, John Wiley&Sons, 2002.
[17] A. Liegeois, "Automatic supervisory control of the configuration and behaviour of multibody mechanisms," *IEEE T. Syst. Man. Cyb.*, vol. SMC-7, pp. 868-871, 1977.
[18] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Trans. ASME J. Dynam. Syst., Meas., Control*, vol. 108, pp. 163-171, 1986.
[19] B. Siciliano and J. J. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," *Proc. Int. Conf. Adv. Robotic.*, Pisa, Italy, pp. 1211-1216, June 1991.
[20] H. Seraji and R. Colbaugh, "Singularity-robustness and task-prioritization in configuration control of redundant robots," *29th IEEE Conf. on Decision and Control*, pp. 3089-3095, 1990.
[21] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. J. Robot. Res*, vol. 4, no.2, pp.3-9, 1985.
[22] J. Park, "*Analysis and control of kinematically redundant manipulators: An approach based on kinematically decoupled joint space decomposition*," PhD thesis, Pohang University of Science and Technology (POSTECH), 1999.
[23] L. Li, W. A. Gruver, Q. Zhang, and Z. Yang, "Kinematic control of redundant robots and the motion optimizability measure," *IEEE T. Syst. Man. Cyb.*, vol. 31, no. 1, pp. 155-160, 2001.
[24] G. Marani, J. Kim, J. Yuh, and W. K. Chung, "Algorithmic singularities avoidance in task-priority based controller for redundant manipulators," *2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Las Vegas, pp.3570-3574, Oct. 2003.
[25] T. Yoshikawa, *Foundations of robotics: analysis and control*, MIT Press, 1990.
[26] B. Nemec, and L. Zlajpah, "Implementation of force control on redundant robots," *1998 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1314–1319, 1998.
[27] J. J. Craig, *Introduction to Robotics*, 2nd Edition. Addison-Wesley, 1989.
[28] D. Steinke, "*Design and Simulation of a Kalman filter for ROV Navigation*," Master's Thesis, University of Victoria, 2006.

Appendix F. Redundancy Resolution for Underwater Mobile Manipulators

(2009, Ocean Engineering, volume 37, pages 325-343)

Ocean Engineering 37 (2010) 325–343

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Redundancy resolution for underwater mobile manipulators

Serdar Soylu *, Bradley J. Buckham, Ron P. Podhorodeski

Department of Mechanical Engineering, University of Victoria, P.O. Box 3055, Victoria, BC, Canada V8W 3P6

ARTICLE INFO

Article history:
Received 28 October 2008
Accepted 23 September 2009
Available online 2 October 2009

Keywords:
Redundancy resolution
Kinematics of underwater vehicle–manipulator systems
Fault-tolerant systems

ABSTRACT

A new fault-tolerant redundancy resolution scheme is presented that allows a single six degree of freedom (DOF) command to be distributed over a small remotely operated underwater vehicle–manipulator (ROVM) system. ROVM systems are composed of a remotely operated underwater vehicle (ROV) and serial manipulator. The combined system is often kinematically redundant for the six-DOF end-effector command, and such a ROVM system admits an infinite number of joint-space solutions for a commanded end-effector state. In the current work, the primary objective is to follow the desired end-effector velocities commanded by a human pilot. The primary objective is realized using the right Moore–Penrose pseudoinverse solution that minimizes the two-norm of the collective joint velocities. Secondary objectives considered are: avoiding manipulator joint limits, avoiding singularities and high joint velocities, keeping the end-effector in sight of the on-board camera, minimizing the ROV motion, and minimizing the drag-forces on the ROV. Each criterion is defined within the framework of the gradient projection method (GPM). The hierarchy for the secondary tasks is established by a low-level artificial pilot that determines a weighting factor for each criterion based on if–then-type fuzzy rules that reflect an expert human pilot's knowledge. The resulting weight schedule yields a self-motion (null-space motion) that emulates how a skilled operator would utilize the redundancy of the ROVM to achieve the secondary objectives. In addition, the proposed method has a fault-tolerant property that enforces joint-velocity limits and also redistributes the end-effector velocity command in the case of faulty joints. To demonstrate the efficacy of the proposed scheme, a numerical simulation case study is performed. The results illustrate that complex spatial end-effector manoeuvres that are otherwise not possible with a stationary ROV can be accomplished in real-time via the coordination of the ROV and the manipulator. The on-line nature of the proposed scheme makes it suitable for remote systems where the desired end-effector state is not known *a priori*.

1. Introduction

Remotely operated underwater vehicles (ROVs) equipped with robotic manipulators are used in a variety of underwater applications such as oil and gas extraction, installation of underwater telecommunication cables, inspections and maintenance of offshore structures and military services. The combined system is referred to as an remotely operated underwater vehicle–manipulator (ROVM).

To date, the motions of the ROV and manipulator are guided independently by a human pilot on a surface support vessel through a long, slender tether that provides power and telemetry. A master–slave arm configuration is used to drive the manipulator joints. This mode of operation decouples the manipulator and the ROV degrees of freedom (DOFs), and depends on the ability of the ROV to hold station such that the absolute motions of the master arm are replicated at depth by the slave manipulator. While ROV motion can be eliminated by automatic control, by lodging the ROV against an immovable object, or by latching on to the environment with an additional arm, this practice eliminates the redundancy inherent in the ROVM system provided by the ROV surge, sway, heave and yaw motions. Furthermore, for smaller ROVs, the serial manipulator is usually underactuated for six-DOF tasks and the elimination of the ROV DOFs results in a very constrained end-effector workspace that is not capable of arbitrary six-DOF motion.

If the collective DOF of the ROVM are fully exploited during manipulator operation, secondary objectives could be achieved. As a consequence, the efficacy of the pilot–ROVM interaction can be improved and spatial operations can be completed with a small inexpensive ROVM system. However, due to the extra DOF, the ROVM system admits an infinite number of joint-space solutions for a commanded end-effector position and orientation, and a redundancy resolution algorithm must be applied to ensure automatic coordination of the ROVM DOF.

As pointed out by Antonelli and Chiaverini (2003), the implementation of redundancy resolution methods to ROVM systems has been documented in only a few existing works.

* Corresponding author. Tel.: +1 250 472 4065; fax: +1 250 721 6035.
E-mail addresses: serdar@me.uvic.ca (S. Soylu), bbuckham@me.uvic.ca (B.J. Buckham), podhoro@me.uvic.ca (R.P. Podhorodeski).

0029-8018/$ - see front matter Crown Copyright © 2009 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.oceaneng.2009.09.007

Frequently, a single secondary objective, or task, is added to the end-effector trajectory following task to ensure a deterministic solution to the inverse kinematics, and the various methods differ in the choice of this secondary task. The singularity robust task-priority redundancy resolution (Chiaverini, 1997) was shown to be effective for a ROV station keeping task of a ROVM by Antonelli and Chiaverini (1998). In Sarkar and Podder (2001), ROVM kinematic redundancy was utilized to minimize the total hydrodynamic drag-forces experienced by a ROVM system in an effort to reduce energy consumption. A unified dynamics-based motion planning method that can produce both kinematically admissible and dynamically realizable joint-space solutions was proposed by Podder and Sarkar (2004). Podder and Sarkar's method took into account the dynamic characteristics of the ROV and manipulator by resolving the redundancy such that the two subsystems produced motions at the end-effector that were within the respective bandwidths. Santos et al. (2006a) used screw theory, and a secondary objective of minimizing ROV motion whenever possible, to solve the inverse kinematics of ROVM systems. Marani et al. (2006) discussed a task reconstruction method with task priority for redundant manipulators. In their work, Marani et al. (2006) focuses mostly on the singularity avoidance and discusses the potential applications of their method for collision and joint limit avoidance. Han and Chung (2007) exploit the redundancy to minimize the restoring moments of ROVM systems.

In actual ROVM operations a human pilot manages a series of secondary objectives with the ROV while executing an end-effector task. These could include: avoiding contact with the environment, station keeping the ROV, adjusting the ROV camera view through ROV motions, and moving the ROV to ensure that the manipulator joints do not reach the limits of travel. Unfortunately for the pilot, secondary objectives are often not complementary. For example, perfect station keeping eliminates any ability to move the ROV to help avoid manipulator joint limits. Given that most secondary objectives can easily be formulated as kinematic criteria, redundancy resolution offers opportunity to automatically pursue multiple secondary considerations if a blending of the secondary tasks' requirements can be automated. To this end, fuzzy-logic has proven to be very useful.

In Antonelli and Chiaverini (2003), the singularity robust task-priority redundancy resolution is merged with a fuzzy technique to resolve the ROV–manipulator coordination. It was shown that the fuzzy method provides a versatile tool to handle multiple secondary tasks. A fuzzy hybrid system was proposed by Santos et al. (2006b) for the redundancy resolution of ROVM systems. The work of Santos et al. (2006b) requires fuzzy rules for guiding the ROV motion in a manner consistent with the predefined "hysteresis" behaviour. Among all of these works, only the work by Antonelli and Chiaverini (2003) addresses the issue of multiple redundancy resolution criteria in ROVM systems in detail. However, none of the previous works accounts for the possibility of faulty manipulator joints, or prevents the occurrence of excessive joint velocities.

In the current work, as a first contribution, the gradient projection method (GPM) (Liegeois, 1977) has been merged with a fuzzy-logic approach in an effort to afford on-line generation of ROVM null-space joint rates that satisfy secondary kinematic criteria. The advantage of using the GPM approach over the existing multitask redundancy resolution approach (a task-priority approach) is that the GPM does not require specification of desired secondary task values or the generation of a Jacobian matrix that translates these desired values into a set of joint rates. As such, the GPM yields a simpler means for the inclusion of secondary task objectives in the redundancy resolution algorithm.

As a second contribution, the current work develops a fault-tolerance property for the GPM framework. The fault-tolerance strategy proposed here for ROVM operation is based on the work of Chen et al. (1999). In the current work, a joint fault can have three interpretations: first, it occurs when a joint completely fails; second, a joint can lose a portion of its driving capacity; and third, the redundancy resolution scheme may produce joint velocities that are beyond the capacity of the corresponding joints for the current pose. The resulting scheme allows one to put velocity limits on respective DOF, and thus prevents the occurrence of excessive joint velocities. The fault-tolerance property exploits the system redundancy to prevent interruption of a mission due to failing manipulator joints, and is of vital importance in reducing mission durations and costs.

Within the proposed scheme, each secondary task is quantified by an objective function, and the proper null-space joint rates are calculated through optimization of the objective function values. To reflect the intuition and skill of an experienced human pilot, the hierarchy of the secondary tasks is established by a low-level artificial pilot that determines a weighting factor for each objective based on if–then-type fuzzy rules. A Mamdani fuzzy inference system (Driankov et al., 1995) is used to interpret the fuzzy rules based on the sensory knowledge. The resulting weight schedule yields a self-motion (null-space motion) that emulates how a skilled operator would utilize the redundancy of the ROVM. A preliminary discussion about the fault-tolerant redundancy resolution scheme is given by the authors of the current work in Soylu et al. (2007a).

To demonstrate the efficacy of the proposed GPM and fault-tolerance methods, a numerical simulation case study is performed using a kinematic model of a small ROVM system. The examples illustrate how secondary objectives are included within the GPM framework, how the fuzzy rules evolve the hierarchy of these objectives, and how the proposed approach successfully distributes a single six-DOF end-effector command over a small ROVM system composed of an otherwise under-actuated ROV and underactuated serial manipulator.

The remainder of the paper is outlined as follows: Section 2 describes the system state vector as well as the frame attachment. Section 3 explains the basis of the multiple-objective GPM method employed in the current work. Section 4 presents a series of additional secondary objectives as well as their incorporation into the GPM approach. Section 5 addresses the basics of the fuzzy-logic algorithm used in the current work, and explains the set of linguistic rules used in the fuzzy-logic. Section 6 extends the multiple-objective GPM to include fault-tolerant properties. Section 7 summarizes the fault-tolerant multiple-objective GPM approach and provides a general scheme for the proposed method. Sections 8 and 9 present a case study and conclusions, respectively.

2. ROVM kinematics

2.1. State vector and coordinate frames

The state vector of a ROVM system illustrated in Fig. 1 can be given as $\mathbf{q}=[q_1 q_2 \ldots q_n]^T \in \Re^n$, where q_1, q_2 and q_3 are the liner displacements of the vehicle in terms of the inertial reference frame, q_4, q_5 and q_6 are the Euler angles (roll-pitch-yaw) of the vehicle, and finally q_7, q_8 to q_n are the joint displacements of the manipulator in terms of the body-fixed frames attached to each respective link (it could be either prismatic or revolute.)

The differential kinematic relationship between the task-space velocities $\dot{\mathbf{x}} \in \Re^m$ and the joint-space velocities $\dot{\mathbf{q}} \in \Re^n$ is given by

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} \tag{1}$$

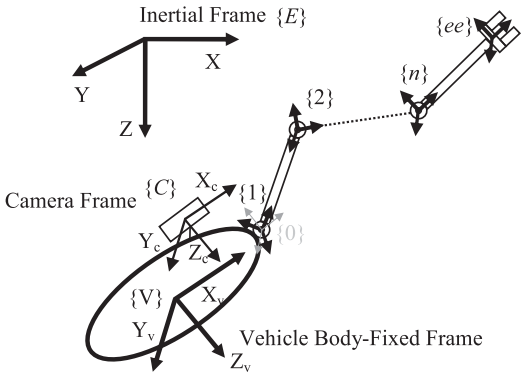


Fig. 1. Coordinate frames for a ROVM system.

where $\mathbf{J} \in \Re^{m \times n}$ is the end-effector Jacobian matrix with m and n being the dimension of the task space and the total number of DOF in the ROVM system, respectively. Given that a typical ROVM mission requires position/orientation control of the end-effector frame, the dimension of the task space is six; three for the position and three for the orientation, i.e., $\dot{\mathbf{x}} \in \Re^6$.

2.2. Derivation of Jacobian matrix

2.2.1. ROVM Jacobian matrix

In order to derive the Jacobian matrix given in Eq. (1), the joint directions and derivatives method can be used. According to this method, the Jacobian matrix for a six-dimensional task space, i.e., $\mathbf{J} \in \Re^{6 \times n}$, is given as

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \mathbf{p}_{I \to ee}}{\partial q_1} & \dfrac{\partial \mathbf{p}_{I \to ee}}{\partial q_2} & \cdots & \dfrac{\partial \mathbf{p}_{I \to ee}}{\partial q_n} \\ \mathbf{z}_1 & \mathbf{z}_2 & & \mathbf{z}_n \end{bmatrix} \tag{2}$$

where $\mathbf{p}_{I \to ee} \in \Re^{3 \times 1}$ is the position vector from the inertial reference frame to the end-effector (ee) expressed in the inertial reference frame, and $\mathbf{z} \in \Re^{3 \times n}$ is the direction matrix containing the joint direction vectors in terms of the inertial reference frame. Note that the calculation of roll–pitch–yaw rates is singular when the ROVM system operates close to the singular configuration (a nose down or up stance of the ROV). However, due to the strong buoyant restoring moment, ROVM systems naturally avoid these singular configurations. If there is a requirement for a ROVM system to work close to singular configurations, the quaternion method demonstrated in Antonelli and Chiaverini (2003) can be employed. The next two subsections explain the derivation of the position vector and the direction vectors.

2.2.2. Derivation of the position vector $\mathbf{p}_{I \to ee}$

The homogeneous transformation matrix from the ROV's body-fixed frame to the inertial frame ${}^{I}_{v}\mathbf{T} \in \Re^{4 \times 4}$ is given by

$${}^{I}_{v}\mathbf{T} = \begin{bmatrix} {}^{I}_{v}\mathbf{R} & \mathbf{p}_{I \to v} \\ 0_{1 \times 3} & 1 \end{bmatrix} \tag{3}$$

where $\mathbf{p}_{I \to v}$ is the position vector of the ROV in terms of the inertial reference frame, and ${}^{I}_{v}\mathbf{R} \in \Re^{3 \times 3}$ is the rotation matrix describing the ROV's body-fixed frame with respect to (w.r.t.) the inertial frame and is defined as

$${}^{I}_{v}\mathbf{R} = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix},$$
$$s \equiv \sin(*), c \equiv \cos(*), t \equiv \tan(*). \tag{4}$$

The homogeneous transformation matrix from the inertial frame to the end-effector can be obtained by multiplying the intermediate transformations in the following order:

$${}^{I}_{ee}\mathbf{T} = {}^{I}_{v}\mathbf{T}\,{}^{v}_{0}\mathbf{T}\,{}^{0}_{1}\mathbf{T}\,{}^{1}_{2}\mathbf{T} \ldots {}^{n-1}_{n}\mathbf{T}\,{}^{n}_{ee}\mathbf{T} \tag{5}$$

where ${}^{v}_{0}\mathbf{T}$ is the homogeneous transformation matrix between the vehicle and the 0th link containing the relevant rotation matrix as well as the position vector, ${}^{i-1}_{i}\mathbf{T}, i = 1 \ldots n$, is the homogeneous transformation matrix relating the coordinate frame i with respect to $i-1$ based on the modified D&H parameters (Craig, 1989). The last column of the resulting homogeneous transformation matrix denotes the position vector $\mathbf{p}_{I \to ee}$ and the 3×3 matrix of the first three columns and rows is the rotation matrix describing the orientation of the end-effector wrt the inertial reference frame, i.e., ${}^{I}_{ee}\mathbf{R}$:

$${}^{I}_{ee}\mathbf{T} = \begin{bmatrix} {}^{I}_{ee}\mathbf{R} & \mathbf{p}_{I \to ee} \\ 0_{1 \times 3} & 1 \end{bmatrix} \tag{6}$$

2.2.3. Derivation of the joint direction vector

The joint directions can be extracted from the rotation matrices as follows:

$$\mathbf{z}_k = {}^{I}_{k}\mathbf{r}_3 \tag{7}$$

where ${}^{I}_{k}\mathbf{r}_3$ is the third column of the rotation matrix ${}^{I}_{k}\mathbf{R}$, i.e., ${}^{I}_{k}\mathbf{R} = \begin{bmatrix} {}^{I}_{k}\mathbf{r}_1 & {}^{I}_{k}\mathbf{r}_2 & {}^{I}_{k}\mathbf{r}_3 \end{bmatrix}$, defined as ${}^{I}_{k}\mathbf{R} = {}^{I}_{1}\mathbf{R}\,{}^{1}_{2}\mathbf{R} \ldots {}^{k-1}_{k}\mathbf{R}$.

Finally, the Jacobian matrix can be formed by taking the derivative of the position vector $\mathbf{p}_{I \to ee}$ from (6) w.r.t. the relevant joint variables and finding the joint direction vectors $\mathbf{z}_k$ from (7) and finally substituting into (2).

3. Redundancy resolution

For a kinematically redundant manipulator ($n > m$) and there are an infinite set of joint rates solutions that can complete the desired end-effector motion, $\dot{\mathbf{x}}$. The predominant joint rate solution seeks to reduce the collective joint rates by minimizing the two-norm $\|\dot{\mathbf{q}}\|_2 = \sqrt{\dot{\mathbf{q}}^T \dot{\mathbf{q}}}$. The minimum-norm solution forms a particular solution to the inverse velocity problem (finding the system state values for a given end-effector velocity (Craig, 1989)) associated with (1) and is given by

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger} \dot{\mathbf{x}} \tag{8}$$

where $\mathbf{J}^{\dagger} = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T)^{-1}$ is the right Moore–Penrose pseudoinverse (see Watkins (2002) for details). A general solution to (1), that sacrifices the minimum-norm properties and satisfies secondary objectives, can be obtained by adding a homogeneous solution—a collection of joint velocities that lie within the null-space of $\mathbf{J}$:

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger} \dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}) \dot{\mathbf{q}}_0 \tag{9}$$

where $\dot{\mathbf{q}}_0 \in \Re^n$ is a vector, and the projection operator $(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})$ projects $\dot{\mathbf{q}}_0$ onto the null-space of $\mathbf{J}$. The resulting null-space vectors generate motions within the ROVM system that do not produce any end-effector motion and these internal motions can be exploited to achieve additional, or secondary, objectives.

Liegeois (1977) proposed that the vector $\dot{\mathbf{q}}_0$ be the gradient of a scalar objective function $h(\mathbf{q})$ yielding the GPM solution:

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger} \dot{\mathbf{x}} + \lambda (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}) \left(\nabla h(\mathbf{q}) \right) \tag{10}$$

In (10), the null-space joint velocities are used to maximize the change in $h(\mathbf{q})$, for positive values of the scalar gain λ, or minimize $h(\mathbf{q})$, for negative values of λ.

To include multiple objectives in the GPM the following equation can be used:

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{x}} + \lambda(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})(\sum_{j=1}^{s} \alpha_j \nabla h_j / \|\nabla h_j\|_2) \tag{11}$$

where s is the total number of kinematic criteria, and α_j is a weight that determines the emphasis on the jth objective of Section 4.3. In order to prevent one single criterion from dominating the null-space velocities, the normalized gradient is used in (11). If the norm of the jth gradient is equal to, or near, zero, the manipulator is in a configuration which cannot affect the jth objective and it is eliminated from consideration. A proof of the decrease of the multiple-criteria objective function through the GPM approach is given in Appendix A.

For ROVM systems, the degrees of freedom are subject to kinematic and dynamic constraints that can limit their utility in producing the desired end-effector motion. For example, the manipulator joints of some ROVM systems have severe joint limits while the ROV drag and inertia are large and ROV rates requested by the redundancy resolution may not be achieved. To this end, the Jacobian weighted-pseudoinverse $\mathbf{J}_w^{\dagger}$ can be used in place of $\mathbf{J}^{\dagger}$ to set preference for certain joints in the ROVM:

$$\mathbf{J}_w^{\dagger} = \mathbf{W}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^T)^{-1} \tag{12}$$

where

$$\mathbf{W}^{-1} = \mathrm{diag}(w_1,\ w_2, \ldots, w_n) \tag{13}$$

is a positive-definite matrix of weight factors with w_j corresponding the jth entry in $\dot{\mathbf{q}}$. The weighted-pseudoinverse minimizes $\dot{\mathbf{q}}^T\mathbf{W}\dot{\mathbf{q}}$ subject to $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$. As shown in Appendix B, the GPM solution relies on the fact that the projection operator is semi-definite. However, when the weighted-pseudoinverse is used, the weighted projection operator, $(\mathbf{I} - \mathbf{J}_w^{\dagger}\mathbf{J})$, is not symmetric, and therefore is non-definite (Yoshikawa, 1990). For this reason, the gradient projection method must be modified. This can be done by premultiplying the weighted projection operator by the inverse of the positive-definite weighting matrix, $\mathbf{W}^{-1}$. The weighted GPM approach becomes

$$\dot{\mathbf{q}} = \mathbf{J}_w^{\dagger}\dot{\mathbf{x}} + \lambda(\mathbf{I} - \mathbf{J}_w^{\dagger}\mathbf{J})\mathbf{W}^{-1}(\sum_{j=1}^{s} \alpha_j \nabla h_j / \|\nabla h_j\|_2) \tag{14}$$

A proof of the decrease of the multiple-criteria objective function through the weighted GPM approach is given in Appendix B. Detailed discussions regarding the pseudoinverse and weighted-pseudoinverse-based redundancy resolution along with several extended examples can be found in Nakamura (1991).

The magnitude of a self-motion within the optimal $\dot{\mathbf{q}}$ is directly affected by the magnitude of the projection of the corresponding gradient into the null-space and the user specified gain λ. If the product of λ and the projection magnitude is too small, then the system will respond slowly to poor objective values, h_j. If λ is increased too much to compensate this tendency, oscillations may result as shown by Euler et al. (1989). In addition, if λ is too large, unnecessary self-motions can drive the ROVM into configurations that border the joint limits and decrease the system's ability to exploit redundancy. In light of these concerns, a systematic approach to determine the λ gain is given by Chen et al. (1995). In addition, the calculation of the bounds on the scalar gain λ, in consideration of joint-velocity limits, is outlined by Li et al. (2001). In the current work, the problem of slow convergence, or oscillating self-motion is managed primarily by proper selection of α_j.

4. Motion objectives

4.1. Definition of secondary objectives

In this section, a series of secondary objectives is presented —including prevalent ROVM objectives presented in past studies and new objectives introduced to expand the hierarchy of secondary tasks. The series is by no means exhaustive—application specific objectives could be added to those listed here. The joint limit avoidance and minimal vehicle motion are included through the weighting matrix defined in (13). In this implementation, w_i, i=1…6 are reserved for the minimal vehicle motion objective, whereas w_i, i=7…n are reserved for the joint limit avoidance. The singularity avoidance, the camera view, and the drag force minimization objectives are included through the GPM.

4.2. Through the weighting matrix

4.2.1. Joint limit objective

To account for the ith manipulator joint's physical limits, the weighting matrix defined in (13) is evaluated dynamically to ensure that the joint weight factors discourage, or prevent, use of a joint when it nears, or is at its limit. As shown by Chan and Dubey (1995), a pertinent objective function can be chosen as

$$s = \sum_{i=7}^{n} (q_{i,max} - q_{i,min})^2 / \left[C_i(q_{i,max} - q_i)(q_i - q_{i,min})\right] \tag{15}$$

where $q_{i,max}$ and $q_{i,min}$ are the upper and lower bounds on the displacements for the ith joint. C_i is a positive number defining the strictness of the constraint for the ith joint. Note that only the manipulator joints are weighted since there are no physical limits on the ROV's DOFs. Differentiating (15) with respect to the joint variables yields:

$$\frac{\partial s}{\partial q_i} = \frac{(q_{i,max} - q_{i,min})^2(2q_i - q_{i,max} - q_{i,min})}{C_i(q_{i,max} - q_i)^2(q_i - q_{i,min})}, i = 7 \ldots n. \tag{16}$$

The components of the diagonal weighting matrix given in (13) are defined as

$$w_i = 1/(1 + |\partial s/\partial q_i|), i = 7 \ldots n \tag{17}$$

According to this definition, the value of diagonal elements varies within the interval [0,1]: when the ith joint is at the middle of its range, w_i is equal to one (i.e., the motion of the corresponding joint is allowed), and becomes zero at either limit (i.e., the motion of the corresponding joint is not allowed).

If the diagonal component w_i is already approaching one, or in other words, if the joint is moving away from its limits, the corresponding joint should not be penalized. That way, the available redundancy can be committed to other tasks. To this end, w_i is redefined for i=7…n as

$$\begin{array}{ll} \text{if } \Delta|\partial s/\partial q_i| \geq 0, & w_i = 1/(1 + |\partial s/\partial q_i|) \\ \text{otherwise,} & w_i = 1 \end{array} \tag{18}$$

where $\Delta|\partial s/\partial q_i| \geq 0$ correspond to the case in which the joint moves closer to its limits with the sign Δ referring to the difference between the two consecutive values of $\partial s/\partial q_i$.

4.2.2. Minimal vehicle motion objective

While the redundancy resolution scheme seeks to exploit the ROV DOF, the responsiveness of the ROV may be compromised by several factors including: its relatively large inertia (both real and added), the time delay in the generation of control forces by conventional propeller type thrusters, and large (and possibly erratic) tether disturbances. Hence, while a ROV DOF may enable

the spatial operation of the end-effector, the demands placed on the DOF must be held within reason.

In addition to avoiding joint limits, the weighting matrix defined in (13) is used to discourage excessive ROV velocities. The ROV-related diagonal elements determine the availability of the corresponding ROV DOF. For instance, a zero value indicates that the corresponding ROV DOF cannot be committed to the task. To allow the weighting on each ROV degree of freedom to vary during a manoeuvre, an additional time variant parameter, $\alpha_v \in [0,1]$, which is generated by the fuzzy-logic described in Section 5, is used to modulate the weighting factor w_i:

$$w_i = (1-\alpha_v) w_{max,i}, i = 1 \ldots 6 \tag{19}$$

where the positive scalar $w_{max,i}$ is the user defined, maximum value of weigh factor for the corresponding DOF of the vehicle.

When a particular ROV DOF is to be restricted, α_v approaches 1, and in cases where it cannot be utilized at all $\alpha_v = 1$. This case mimics the case of a faulty manipulator joint—the faulty DOF must be removed from consideration within the execution of (14), and Section 5 outlines the procedure for removing faulty joints. In the context of minimizing the ROV motion, the elimination of a failing, or unusable, ROV DOF eliminates the occurrence of a $w_i = 0$ within the weighting matrix. Thus the weighting matrix used in (14) remains positive-definite despite the possibility of $\alpha_v = 1$.

4.3. Through the GPM

4.3.1. Singularity avoidance objective

To avoid singular configurations of the manipulator links, Yoshikawa's measure of manipulability (MOM) (Yoshikawa, 1985) is used to monitor the manipulator joints' proximity to the singular configurations:

$$\varepsilon_{sin} = \sqrt{\det(\mathbf{J}_{sub}\mathbf{J}_{sub}^T)} \tag{20}$$

where $\mathbf{J}_{sub} \in \Re^{3\times(n-6)}$ is the submatrix of the end-effector Jacobian matrix. The submatrix $\mathbf{J}_{sub}$ is composed of the first three rows and the last $(n-6)$ column vectors of $\mathbf{J}$ representing the contribution of the manipulator joints to the end-effector translational velocities. This term is similar to the one presented by Soylu et al. (2007b). However, in Soylu et al. (2007b), a collective Jacobian containing the end-effector and secondary task Jacobian were used. To monitor the mobility index, the following analytical formula can be used (Park, 1999):

$$\nabla h_1 = \frac{\partial \varepsilon_{sin}}{\partial q_i} = \varepsilon_{sin}\, tr\left(\frac{\partial \mathbf{J}_{sub}}{\partial q_i}\mathbf{J}_{sub}^{\dagger}\right), i = 1 \ldots n. \tag{21}$$

where $tr(\bullet)$ refers to the trace of a matrix.

4.3.2. Camera angle objective

To keep the end-effector in sight of the camera, the tilt angle of the camera should be coordinated with the end-effector motion. The camera field-of-view (FOV) can be defined as a cone with a vertex at the camera lens and oriented with the camera frame of reference. The camera's coordinate system $(X_C - Y_C - Z_C)$ is attached to the center of the lens of the camera as shown in Fig. 1. The camera tilt angle about the Y_C axis defined as β, which completely defines the relative orientation of the camera with respect to the ROV's body-fixed frame.

To keep the end-effector in sight of the camera

$$ay_e(\mathbf{q},\beta)^2 + bz_e(\mathbf{q},\beta)^2 \le x_e(\mathrm{q},\beta)^2 \tag{22}$$

where $x_e(\mathbf{q},\beta)$, $y_e(\mathbf{q},\beta)$, and $z_e(\mathbf{q},\beta)$ are the position coordinates of the end-effector in terms of the camera's coordinate frame, and a and b are the positive constants defining the opening angle of the cone. An objective function consistent with the camera angle constraint is

$$\varepsilon_{cam} = x_e(\mathbf{q},\beta)^2 - \left(ay_e(\mathbf{q},\beta)^2 + bz_e(\mathbf{q},\beta)^2\right) \tag{23}$$

where the index ε_{cam} indicates the end-effector's proximity to the boundary of the camera cone: $\varepsilon_{cam} \ge 0$ indicates that the end-effector is inside the camera view, and $\varepsilon_{cam} < 0$ indicates that the end-effector is outside.

The camera index is a function of the manipulator joint angles and the tilt angle of the camera, and the gradient is given by

$$\nabla \varepsilon_{cam} = \left[\left[\partial \varepsilon_{cam}/\partial \mathbf{q}\right]^T \quad \partial \varepsilon_{cam}/\partial \beta\right]^T \tag{24}$$

Since the redundancy resolution scheme guides only the end-effector motion, only the end-effector related gradient information is included in the GPM:

$$\nabla h_2 = \partial \varepsilon_{cam}/\partial \mathbf{q} \tag{25}$$

To expand the visible workspace of the manipulator as much as possible, the camera tilt angle is driven to follow the end-effector location.

A change in β along the gradient direction is defined as $\Delta\beta = k(\partial \varepsilon_{cam}/\partial \beta)$, where k is a positive constant that indicates how aggressive the camera moves. Finally, the location of the camera tilt angle that coordinates the tilt angle with the end-effector location can be given by

$$\dot{\beta} = k(\partial \varepsilon_{cam}/\partial \beta)/\Delta t \tag{26}$$

where Δt is the sampling time for the camera motion controller. The desired camera angle can be found by integrating (26), i.e. $\int \dot{\beta}\, dt$. The camera tilt rates calculated via (26) are truncated to remain within saturation limits. When the tilt angle reaches its upper or lower limit, the tilt rate is set to zero. If the camera tilt motion alone is enough to keep the end-effector in the FOV, then the camera angle constraint can be excluded from the additional objectives.

4.3.3. Drag minimization objective

The ROV component of an ROVM system has different hydrodynamic characteristics in the surge, lateral, and heave directions primarily due to differences in the geometry. A comprehensive drag force minimization algorithm must consider the complete ROVM system dynamics, as is done by Sarkar and Podder (2001). In Sarkar and Podder (2001), the ROVM drag is given in terms of the complete ROVM state $\mathbf{q}$ and drag minimization is the only secondary objective considered in a ROVM redundancy resolution method. In the current work, a simple means to minimize ROVM lateral drag is presented only to expand the range of secondary objectives considered, and thus help demonstrate the efficacy of the fuzzy determination of the secondary objective hierarchy. The approach adopted herein is solely kinematic and similar to that of Antonelli and Chiaverini (1998) in which the surge direction of the ROVM is aligned with a known incident water velocity. For the remainder of the current work, this practice will be referred to as yaw angle alignment. This secondary objective ensures that the larger surge thrust of the ROV is used to combat hydrodynamic drag, but is also at odds with the other secondary objectives.

With ψ_c denoting the angle between the Y_v axis vector of the ROVs body-fixed frame and the vector obtained by projecting the water current vector onto the ROVs body-fixed $X_v - Y_v$ plane, the objective function can be defined as: $\max: -\varepsilon_3^2$, where $\varepsilon_3 = (\psi - \psi_c)$ is the yaw angle index. Finally, the gradient of the objective function is given as

$$\nabla h_3 = -\partial(\varepsilon_3^2)/\partial \mathbf{q} \tag{27}$$

5. Fuzzy inference system

5.1. Overview

The additional objectives defined in Section 4 do not create a cohesive set of internal ROVM motions, and it remains to establish a hierarchy for the additional objectives to resolve expected conflicts. As mentioned in Section 3, the weight factors $\alpha_j \in [0,1]$ define the relative importance of each objective. An evolutionary process for the weight factors is established in this section using a Mamdani fuzzy inference method. In the current work, the fuzzy inference system is not used as an optimization tool. Rather, it is used as a decision-making tool through which each objective is prioritized with respect to the others at a given time. The fuzzy inference affords a dynamic hierarchy of secondary objectives. The design of the inference system has components of fuzzification, definition of knowledge base, decision making and defuzzification.

5.2. Fuzzification process

In the fuzzification process, the fuzzy-logic block takes a crisp, or precise, input x in the interval of the universe of discourse Ω, or range of a fuzzy set A, and determines the degree to which the input belongs to the fuzzy set via the membership function: $\mu_A(x):\Omega \rightarrow [0,1]$: 1 represents full membership, any value between 1 and 0 represents partial membership, and 0 indicates non-membership. This process is repeated for each crisp input.

For the current work, the monitored crisp values for each task are the mobility index (ε_{sin}), the camera angle index (ε_{cam}), and the yaw angle index (ε_{yaw}). These crisp values are first normalized based on their expected maximum values ($\varepsilon/\varepsilon_{max}$), and then are fuzzified via the corresponding membership functions. Note that fuzzifying a crisp value means determining the degree to which the crisp value belongs to one particular input fuzzy set. The input linguistic variables and the corresponding input fuzzy sets are: *singularity*={*close*, *not close*}, *camera angle*={*good*, *not good*}, *yaw angle*={*aligned, not aligned*}, and *vehicle motion*={*minimal, not minimal*}. The output linguistic variables, and the corresponding fuzzy sets, are set to be α_s={*low*, *high*} for $s=1\ldots3$ and α_v={*low*, *high*}. The product of the fuzzification process is a value within the interval of 0 and 1 indicating the degree to which the input belongs to the fuzzy set.

The corresponding membership functions are shown in Fig. 2.

5.3. Knowledge base

The fuzzified crisp values enter into the fuzzy rule block that mimics a human expert's knowledge. This logic is in the general linguistic form of, "if p is A then r is B," where p and r are the linguistic fuzzy variables, and A and B are the linguistic fuzzy sets. The "if p is A" part is called the antecedent (premise), whereas the

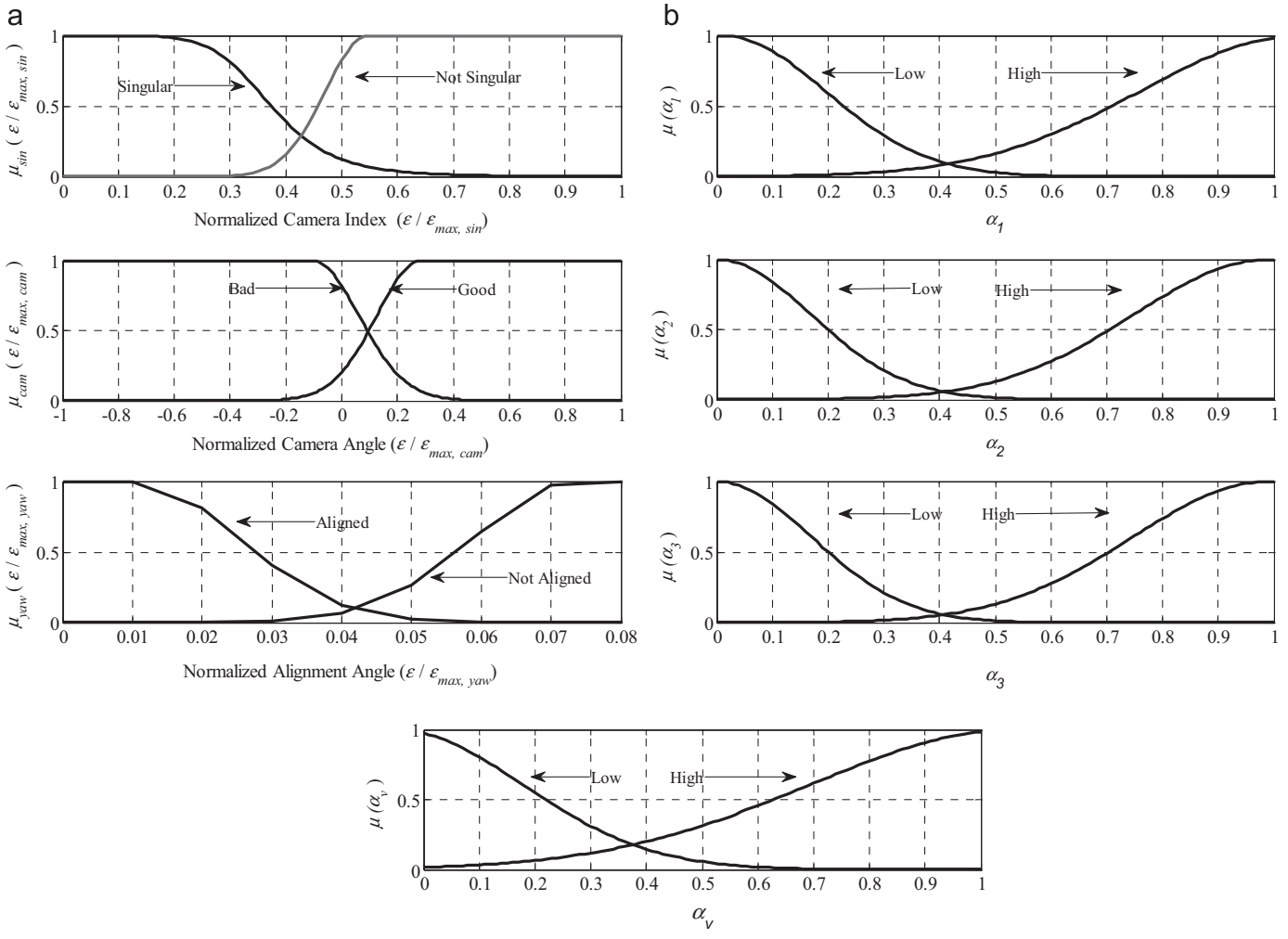


Fig. 2. (a) Input fuzy sets and their membership functions (b) output fuzzy sets and their membership functions.

"then r is B" part is called the consequent (conclusion). For the current study, the following set of rules has been implemented:

- if the manipulator is singular then α_1 is high;
- if the manipulator is not singular then α_1 is low;
- if the manipulator is not singular and the camera angle is bad then α_2 is high;
- if the manipulator is singular or the camera angle is good then α_2 is low;
- if the manipulator is not singular and the camera angle is good and the yaw angle is not aligned then α_3 is high;
- if the manipulator is singular or the camera angle is bad or the yaw angle is aligned then α_3 is low;
- if the manipulator is not singular and the camera angle is good and the yaw angle is aligned then α_v is high;
- if the manipulator is singular or the camera angle is bad or the yaw angle is not aligned then α_v is low.

Using four linguistic variables and two fuzzy sets per variable normally requires a total of 64 rules for four variables. In order to reduce the number of rules, a hierarchical structure was established. The order of priority from the highest to the lowest is in the form of *singularity avoidance→camera angle→yaw angle→vehicle motion*. This arrangement prevents possible conflicts between the secondary tasks by solving in favour of the higher priority task. In addition, this arrangement simplifies the knowledge base by reducing the number of rules to 8. A similar rule construction can be found in Antonelli and Chiaverini (2003). Note that the proposed set of linguistic fuzzy rules do not use the fuzzy-set of *vehicle motion*={*minimal*, *not minimal*} since the minimal vehicle objective has the least priority.

5.4. Decision making

The decision-making procedure includes the application of the fuzzy operators, the implication method, and the aggregation of the outputs. The fuzzy operator is used to interpret *and-or* logic operators. In the current work, the *and-or* logic statements have been carried out based on *min–max* operations, respectively. The fuzzy operator is applied for each rule to translate the outputs of the antecedents into a single number for each rule for the implication step. In the implication step, the single numbers given by the antecedents are the inputs, and the outputs are the truncated output fuzzy sets. The implication function modifies the output fuzzy sets to the extent determined by the outputs (single numbers) of the antecedent of each rule. In the current work, the *min* operation has been used for the implication phase. In the aggregation process, the modified fuzzy sets representing the outputs of each rule are combined into a single fuzzy set. In the current work, the *max* operation has been used for the aggregation phase.

5.5. Defuzzification interface

The resulting fuzzy set is defuzzified (resolved) to a single number that represents the decision output of the fuzzy-logic. In terms of the defuzzification, the centroid method has been followed according to:

$$\alpha_s = \sum_{k=1}^{8} \mu_{k,s}(p) b_{k,s} / \sum_{k=1}^{8} \mu_{k,s}(p), s = 1 \ldots 3,\ v \tag{28}$$

where $\mu_{k,s}(p)$ is the output of the consequent fuzzy sets for the kth rule related to α_s, and $b_{k,s}$ are corresponding centroids of each consequent fuzzy set related to α_s.

6. Fault-tolerant redundancy resolution

A fault-tolerance property prevents the redundancy resolver from requesting intolerable joint motions in the event of joint failure. In the current work, there are three interpretations of joint failure. First, it occurs when a joint breaks down. Second, it happens when a joint has partly lost its driving capacity. Finally, it takes place when the redundancy resolution scheme produces joint velocities that are beyond the current capacity of the corresponding joints. Note that the fault definition can be readily extended to the vehicle's DOF.

When a joint failure occurs, the demanded end-effector velocities must be realized with the remaining non-faulty joints. Therefore, the end-effector velocity $\bar{\dot{\mathbf{x}}} \in \Re^m$ that must be generated by the non-faulty joints can be given as

$$\bar{\dot{\mathbf{x}}} = \dot{\mathbf{x}} - \mathbf{J}_f \dot{\mathbf{q}}_f \tag{29}$$

where $\mathbf{J}_f = [\mathbf{j}_{f,1}\ \mathbf{j}_{f,2} \cdots \mathbf{j}_{f,(n_f+n_s)}] \in \Re^{m\times(n_f+n_s)}$ is the faulty Jacobian matrix consisting of the columns vectors of the Jacobian matrix $\mathbf{J}$ that correspond to the faulty joints, and $\dot{\mathbf{q}}_f = [\dot{q}_{f,1}\ \dot{q}_{f,2} \cdots \dot{q}_{(n_f+n_s)}]^T \in \Re^{(n_f+n_s)}$ is the faulty velocity vector that is a collection of joint velocity values assigned to faulty joints. The number of faulty joints caused by a physical break-down is denoted by n_f and the total number of faulty joints caused by an excessive velocity demand and a loss in driving ability is denoted by n_s, and finally, $\dot{q}_{f,i}\ i = 1 \ldots n_f + n_s$ is the faulty velocity term that is a joint-velocity value assigned to a faulty joint by the designer.

The faulty velocity term $\dot{q}_{f,i}$ is set to zero for joints whose faulty condition is caused by physical break-down. This corresponds to the faulty joint being locked. Regarding the joints that suffer from a lack of driving ability, the faulty velocity term is set to a value lower than its maximum velocity limit. Finally, the faulty velocity term is assigned to a joint-velocity limit for joints whose faulty condition is created by the excess joint-velocity demand requested by the redundancy resolver.

It is assumed that when a full fault occurs due to the physical break-down, the corresponding joint is locked; allowing the system to remain functional. Under this assumption, the condition required for the redundancy resolver to redistribute the joint velocities over the available joints is that the number of available joints must be equal or greater than the dimension of the task-space, i.e., $m \le n - n_f$. Note that since the Jacobian of ROVM systems always exhibit full rank (Antonelli and Chiaverini, 2003), this becomes the only condition. When $n_s \neq 0$ and $m \le n - n_f$, the velocity distribution can be still realized with some joint-velocity constraints imposed by the corresponding joint-velocity limits.

Since the joint velocities are preallocated for the faulty joints, the redundancy resolution distributes joint velocities only over the non-faulty joints. Therefore, the corresponding Jacobian for a faulty ROVM system is obtained by removing the columns from the end-effector Jacobian of the non-faulty ROVM that are associated with the faulty joints. Likewise, the associated components must be removed from the weighting matrix and the joint-velocity vector. This column subtraction process yields the modified end-effector Jacobian, $\bar{\mathbf{J}} \in \Re^{m\times(n-(n_f+n_s))}$, the modified joint-velocity vector $\bar{\dot{\mathbf{q}}} \in \Re^{n-(n_f+n_s)}$ and finally the modified weighting matrix $\overline{\mathbf{W}} \in \Re^{(n-(n_f+n_s))\times(n-(n_f+n_s))}$. For instance, the modified Jacobian $\bar{\mathbf{J}}$, joint-velocity vector and weighting matrix for a ROVM system whose ith joint is in failure can be given as

$$\begin{aligned}
&\bar{\mathbf{J}} = [\mathbf{j}_1\ \mathbf{j}_2\ \cdots\ \mathbf{j}_{i-1}\ \mathbf{j}_{i+1}\ \cdots\ \mathbf{j}_n],\\
&\bar{\dot{\mathbf{q}}} = [\dot{q}_1\ \dot{q}_2\ \cdots\ \dot{q}_{i-1}\ \dot{q}_{i+1}\ \cdots\ \dot{q}_n]^T,\\
&\overline{\mathbf{W}} = diag\,[w_1\ w_2\ \ldots\ w_{i-1}\ w_{i+1}\ \ldots\ w_n]
\end{aligned} \tag{30}$$

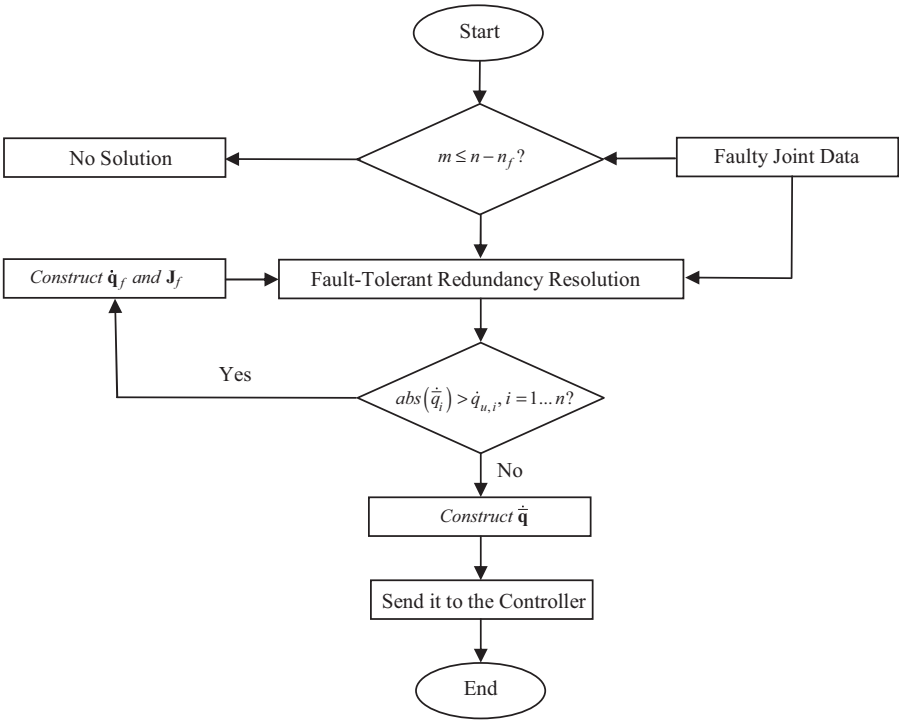


Fig. 3. Fault-tolerant scheme.

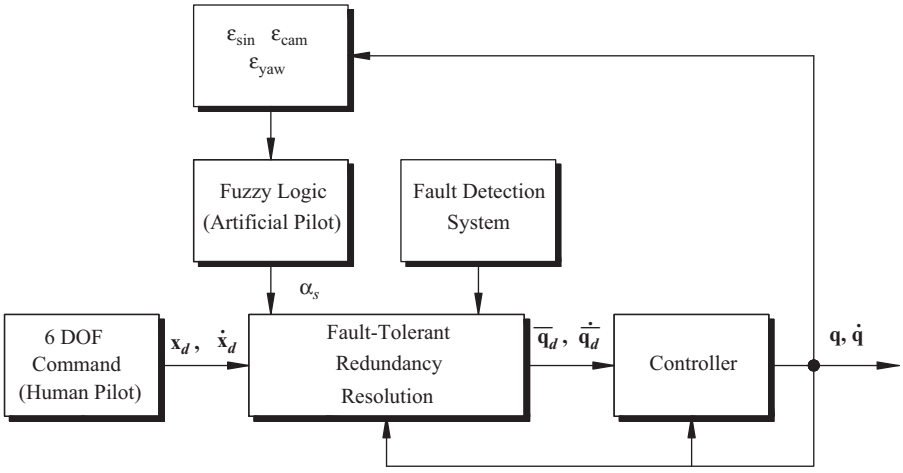


Fig. 4. General scheme of the proposed fuzzy based, fault-tolerant redundancy resolution.

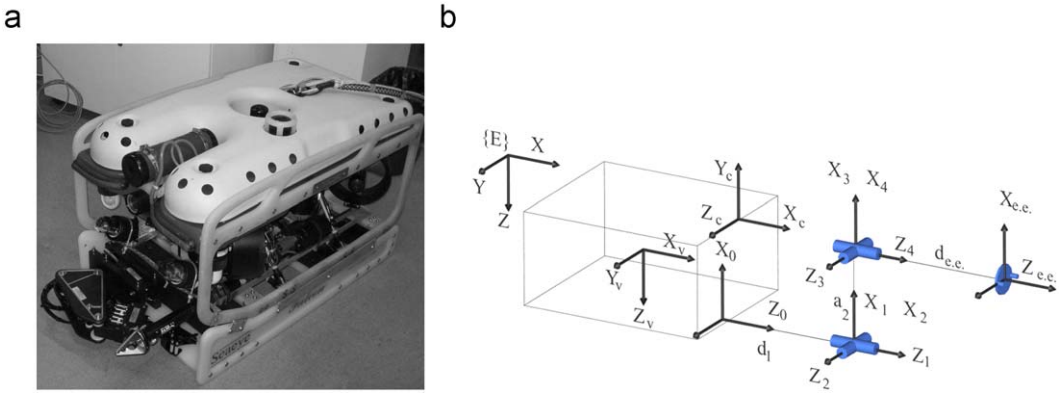


Fig. 5. (a) FALCON^{TM} ROVM with a 4DOF Hydrolek^{TM} manipulator and (b) corresponding zero-displacement configuration.

Finally, the redundancy resolver generates a joint-velocity vector for a faulty ROVM system according to

$$\dot{\bar{\mathbf{q}}} = \bar{\mathbf{J}}_w^{\dagger}(\dot{\mathbf{x}} - \mathbf{J}_f \dot{\mathbf{q}}_f) + \lambda(\bar{\mathbf{I}} - \bar{\mathbf{J}}_w^{\dagger}\bar{\mathbf{J}})\overline{\mathbf{W}}^{-1}\left(\sum_{j-1}^{s} \alpha_j \nabla\bar{h}_j / \|\nabla\bar{h}_j\|_2\right), \; j = 1...3, \; v \tag{31}$$

where $\bar{\mathbf{J}}_w^{\dagger} = \overline{\mathbf{W}}^{-1}\bar{\mathbf{J}}^T(\bar{\mathbf{J}}\overline{\mathbf{W}}^{-1}\bar{\mathbf{J}}^T)^{-1}$ is the modified pseudoinverse of the end-effector Jacobian, and $\bar{\mathbf{I}} \in \Re^{n-(n_f+n_s)\times n-(n_f+n_s)}$ is the modified unit matrix, and finally $\nabla\bar{h}$ is the gradient vector of the objective function excluding the direction contribution from the faulty joints.

Table 1
D&H parameters of the Hydrolek™ manipulator.

$i-1$	α_{i-1}	a_{i-1}	d_i	θ_j	i
0	0	0	d_1=0.15 m	q_1	1
1	$-\pi/2$	0	0	q_2	2
2	0	a_{22}=0.22 m	0	q_3	3
3	$\pi/2$	0	d_{ee}=0.15 m	q_4	*ee*

With this arrangement, the redundancy resolver does not induce any motions for the faulty joints. Finally, the joint-space vector is reconstructed by adding faulty velocity terms in the corresponding places in the joint-velocity vector produced by (31); e.g., for the ROVM system with its *i*th joint being in failure, the joint-velocity vector is given as $\dot{\bar{\mathbf{q}}} = [\dot{\bar{q}}_1 \;\; \dot{\bar{q}}_2 \;\; \dots \;\; \dot{\bar{q}}_{i-1} \;\; \dot{q}_{f,1} \;\; \dot{\bar{q}}_{i+1} \;\; \dots \;\; \dot{\bar{q}}_n]^T$. The execution of general fault-tolerant scheme is presented in Fig. 3.

First, the condition $m \le n - n_f$ is checked to see whether or not there is enough healthy joints to realize the task. If the condition $m \le n - n_f$ is satisfied, then the faulty joint data is fed into the redundancy resolution block. The redundancy resolution block then generates a desired joint rate vector based on (31). Then, each component of the generated joint rate vector is checked according to $abs(\dot{\bar{q}}_i) > \dot{q}_{i,u}$, $i = 1 \dots n$ (The $\dot{q}_{i,u}$ is the joint velocity limit) to see whether or not the generated velocities are within the acceptable range of each relevant joint. If all the joints rates are within the acceptable range, then the generated joint vector is sent to the controller. If they are not, then the fault-tolerant scheme sets the velocity of that particular joint to its current maximum allowable velocity and calls the redundancy resolution scheme once more with only the remaining joints considered available.

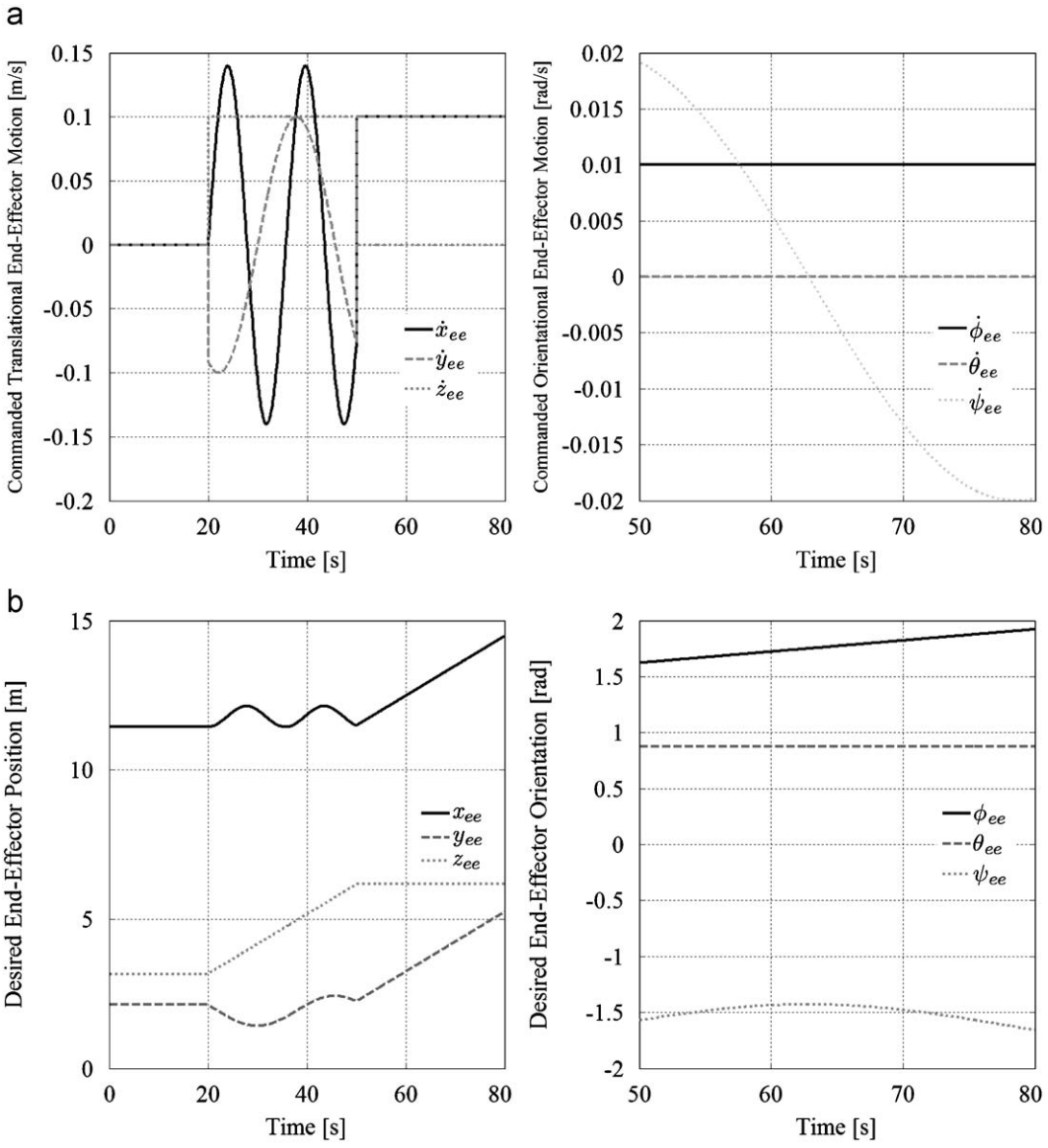


Fig. 6. (a) Commanded end-effector velocities and (b) corresponding end-effector locations.

7. Fault-tolerant gradient projection method with fuzzy-logic

The general scheme of the proposed fault-tolerant fuzzy-based redundancy resolution is given in Fig. 4.

The human pilot provides a six-DOF input using a master arm. The corresponding joint velocities are then calculated by the redundancy resolution block according to (14); factoring in the directives provided by the low-level fuzzy-logic-based artificial pilot. These directives are created based on the indices of ε_{sin}, ε_{cam}, and ε_{yaw} that are calculated using the sensory knowledge. The computed joint-velocity vector along with the faulty joint knowledge (the current FALCON^{TM} system does not have a fault detection system. However, we assume in this work that joint faults can be detected.), a new joint-space vector is computed based on (31). This step fills two roles: first, it redistributes the joint-velocity vector generated by (14) such that every joint has a velocity within its allowable range; second, the faulty joints are not assigned any velocity values. The resulting joint-space vector along with the current joint position vector is then fed into the controller block. The controller block is responsible for computing corresponding actuators forces, (i.e., the ROV thrusters and the manipulator torques) that yield the desired joint-velocity vector.

8. Simulation results

8.1. Simulation conditions

The ROVM system considered in this work is a Saab Seaeye FALCON^{TM} ROV equipped with a 4-DOF Hydrolek^{TM} manipulator as shown in Fig. 5a. As illustrated in Fig. 5b, the inertial frame is labelled

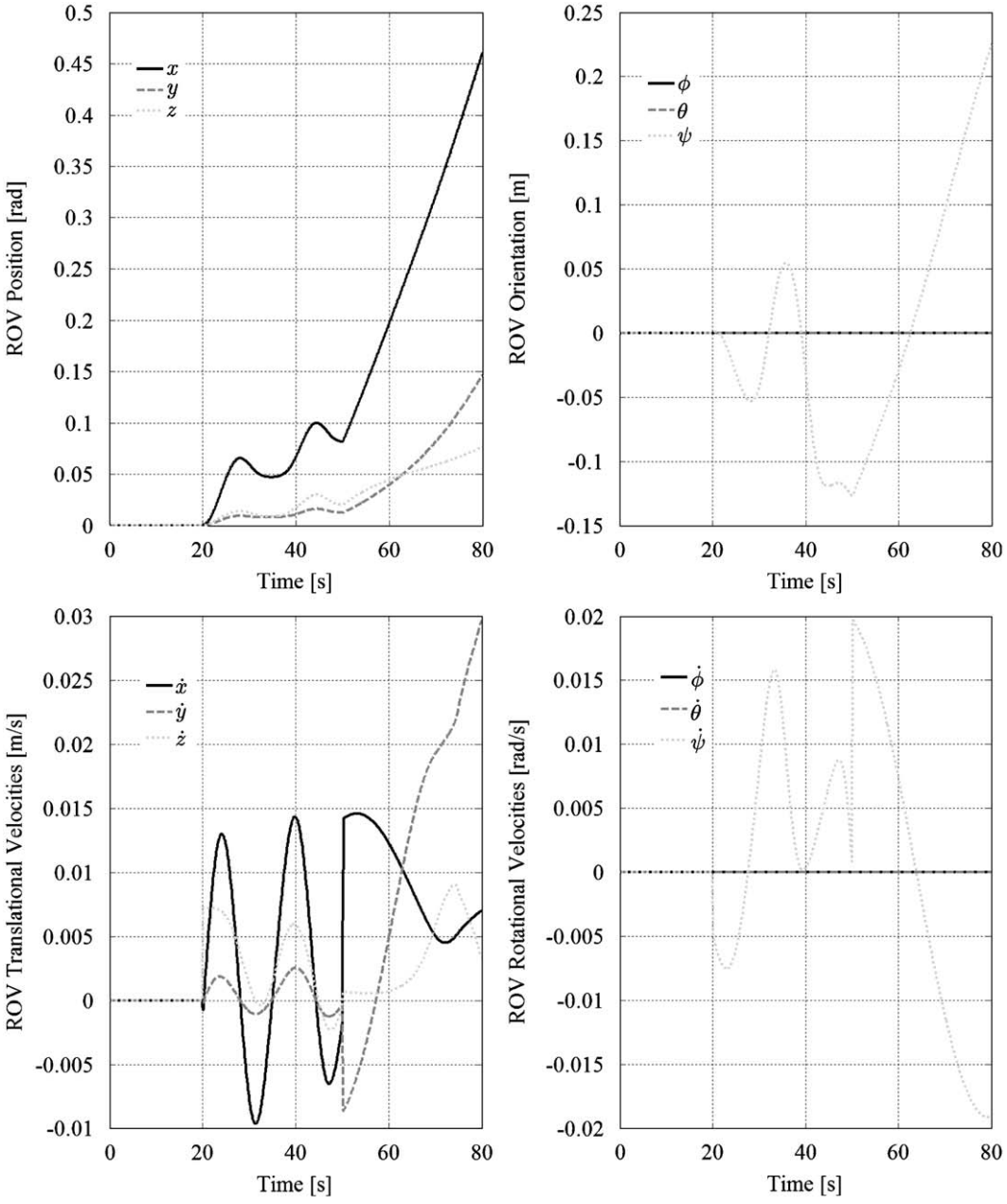


Fig. 7. ROV position and velocity values with weighted-pseudoinverse (case study 1).

as "E", and the body-fixed frame (X_v, Y_v, Z_v) is attached to the center of mass of the ROV. Regarding the manipulator, the body-fixed frames are attached to a base joint of each successive link using the modified Denavit–Hartenberg convention (Craig, 1989) shown in Table 1. The state vector is given as $\mathbf{q}=[x\ \ y\ \ z\ \ \phi\ \ \theta\ \ \psi\ \ q_1\ \ q_2\ \ q_3\ \ q_4]^T$, where x, y, and z are the position coordinates of the ROV's center of mass expressed in the inertial frame; ϕ, θ, and ψ are roll, pitch, and yaw Euler angles defining the orientation of the ROV with respect to the inertial frame, and finally, q_i, $i=1\ldots4$ are the scalar joint position values with respect to the body-fixed link coordinate systems. The ROVM is bounded by a rectangular prism with length, width, and height being 1, 0.6 and 0.5 m, respectively. Each manipulator link is modeled as cylinders with the length of link 1, 2, and 3 being 0.15, 0.22, and 0.15 m, respectively.

In field operations the pilot drives the end-effector with a joystick to generate velocity commands, i.e., $\dot{\mathbf{x}}$ in (14). As a case study, a spatial end-effector task has been set including time series of the velocity commands for the end-effector (corresponding to a six-DOF human pilot input), which are demonstrated in Fig. 6a. The corresponding desired end-effector position and orientation values w.r.t. the inertial reference frame are illustrated in Fig. 6b. The proposed fuzzy-logic based, fault-tolerant scheme given in (31) is used to follow the synthetic pilot end-effector velocity commands while fulfilling the secondary tasks (joint limits, singularity, camera view, yaw angle, and minimal ROV motion tasks) whenever possible.

The Hydrolek™ manipulator has the following joint limits with respect to the zero displacement ($q_i=0$) configuration shown in Fig. 5; $180^\circ \le q_1 \le 290^\circ$, $-95^\circ \le q_2 \le -5^\circ$, $90^\circ \le q_3 \le 135^\circ$, $-180^\circ \le q_4 \le 180^\circ$. In the simulation, the joint-velocity limits are assumed to be 0.14 rad/s for the manipulator joints. The second joint stops functioning after 50 s. Water current is assumed to flow along the direction of $[0.95\ \ 0.3\ \ 0]^T$ w.r.t. the inertial reference frame. This corresponds to a vector that is on the X–Y plane of the inertial frame making the angle of 0.2 rad (11.45°) w.r.t. the X-axis of the inertial frame.

In the simulation study, the system starts from the initial configuration of $\mathbf{q}=[0\ \ 0\ \ 0;\ 0\ \ 0\ \ 0;\ 230\ \ 91\ \ -92\ \ 0]^T$m; rad; deg, which is a stretched-arm (singular) configuration. The corresponding initial end-effector position and orientation are $[1.14\ \ 0.21\ \ 0.32]^T$m and $[1.62\ \ 0.87\ \ -1.57]^T$rad, respectively. The total manoeuvre consists of three portions covering the time intervals of 0–20, 20–50 and 50–80 s, respectively. In the first portion of the manoeuvre (0–20 s), the end-effector is commanded to maintain its initial position. In the second portion of the manoeuvre (20–50 s), only the end-effector translational velocities of $\dot{\mathbf{x}}_{(20-50\,s),t}=[0.14\sin(2t/5-8)\ \ 0.1\sin(0.2t-6)\ \ 0.1]^T$ m/s are commanded. Finally, in the last portion of the manoeuvre (50–80 s), both the translational and rotational velocities are commanded to move according to $\dot{\mathbf{x}}_{(50-80\,s),t}=[0.1\ \ 0.1\ \ 0]^T$ m/s and $\dot{\mathbf{x}}_{(50-80\,s),r}=[0.01,\ \ 0\ \ -0.02\sin(0.1t)]^T$ rad/s, respectively. The corresponding desired end-effector position and orientation

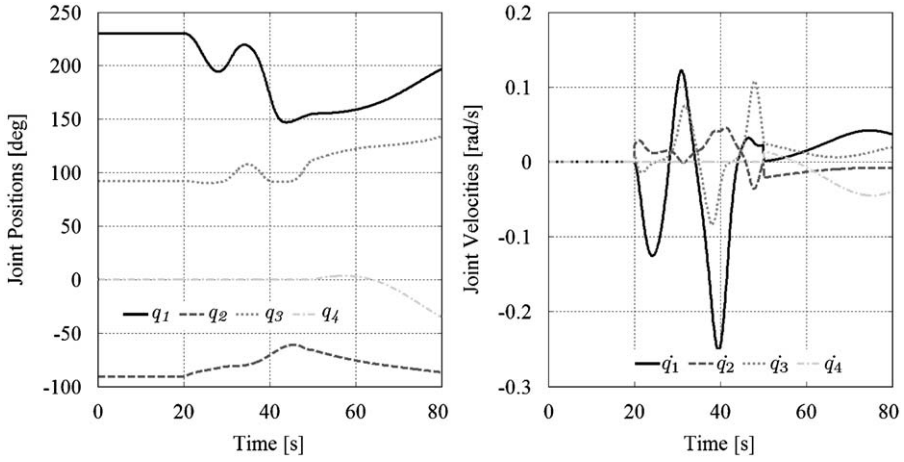


Fig. 8. Manipulator position and velocity values without the fuzzy-logic (case study 1).

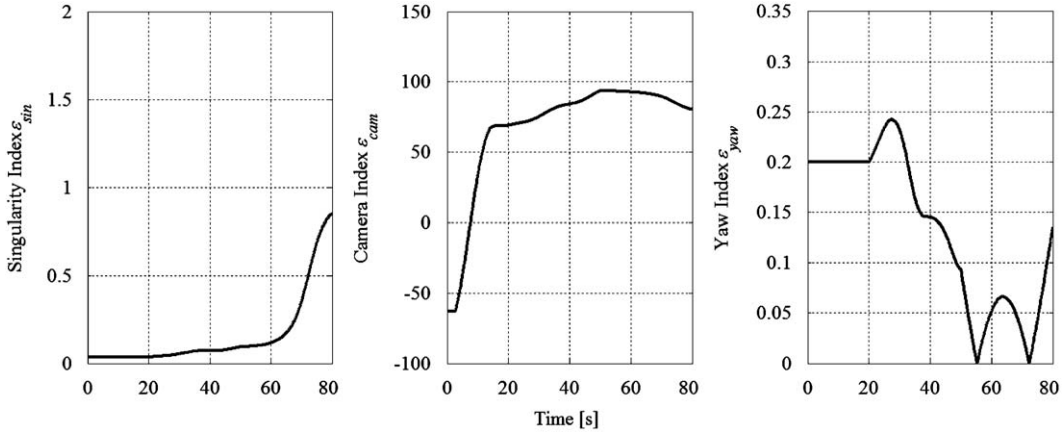


Fig. 9. Time history of singularity camera and yaw indices (case study 1).

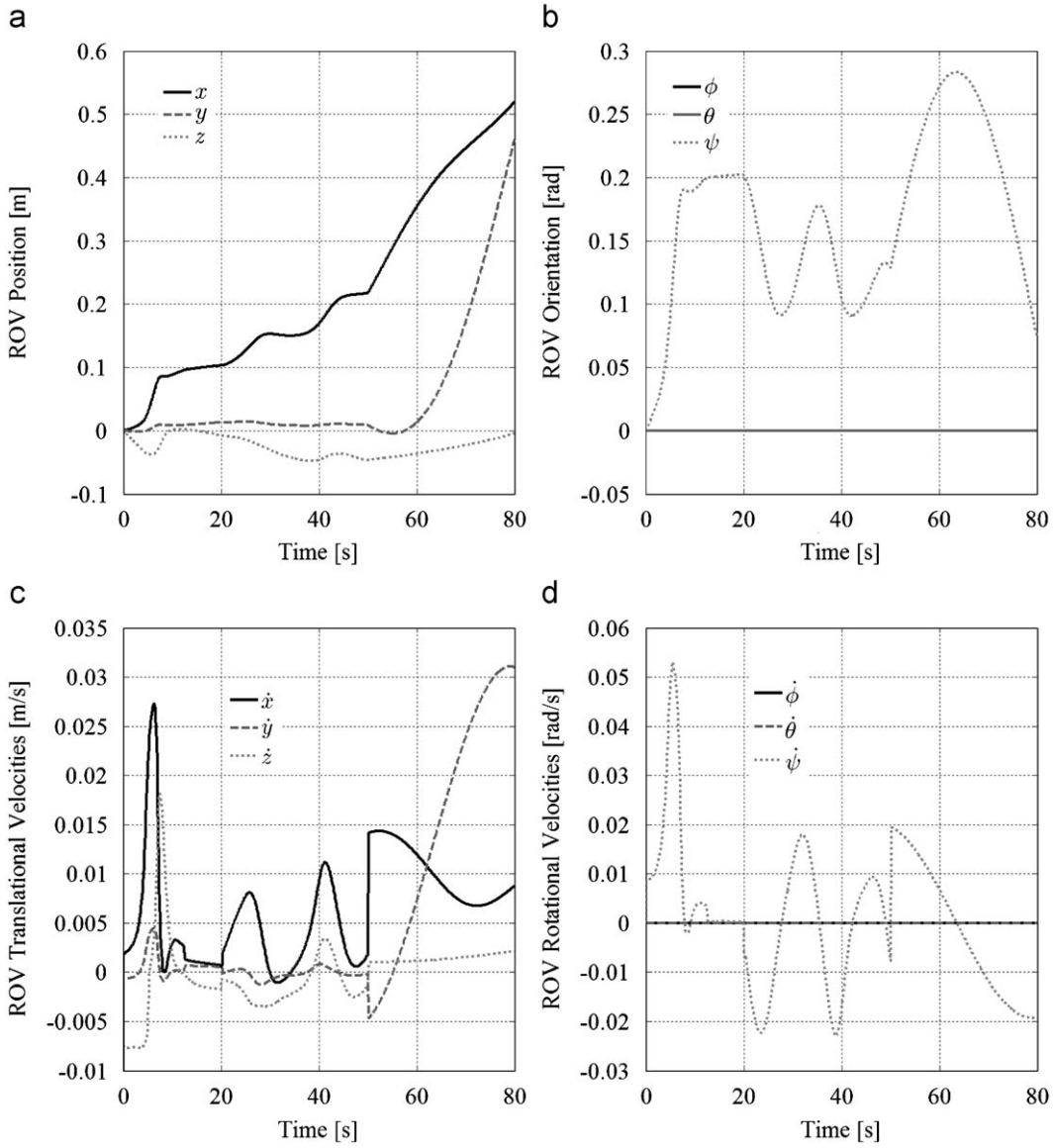


Fig. 10. ROV position and velocity values (case study 2).

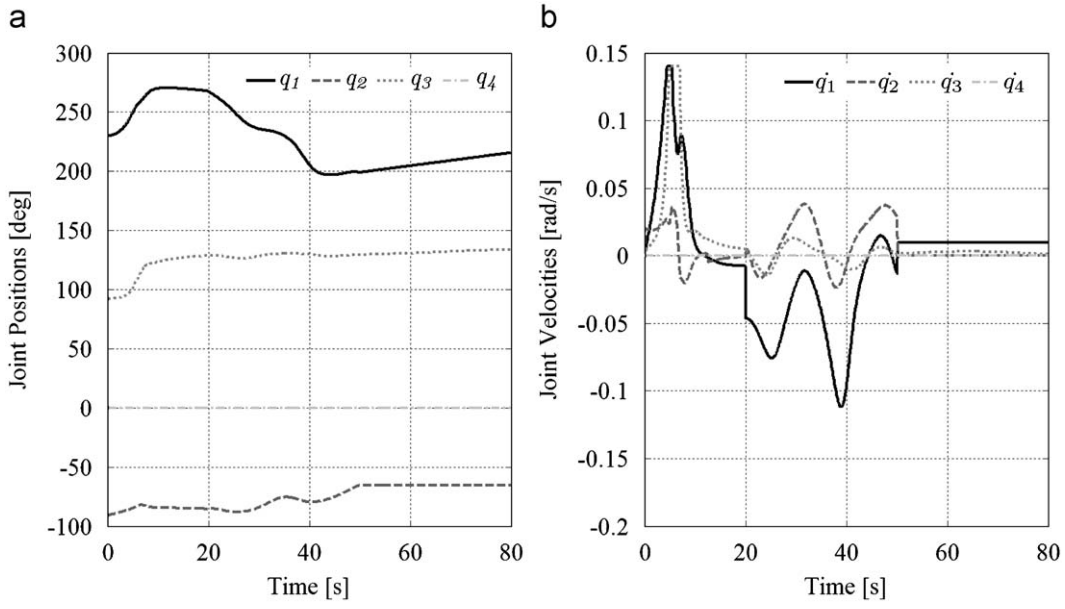


Fig. 11. Manipulator position and velocity values (case study 2).

values w.r.t. the inertial reference frame are illustrated in Fig. 6b. The initial camera angle was set to 45° around the Z-axis of the camera frame.

8.2. Case study 1

In the first case study, the task is to make the end-effector follow the pilot's velocity command. To this end, a weighted-pseudoinverse without the fuzzy-logic was used, i.e., $\dot{\mathbf{q}} = \mathbf{J}_w^{\dagger}\dot{\mathbf{x}}$. The weighting matrix was used not only to distribute the end-effector motion over the vehicle and the manipulator in proportion to their inertias but also to eliminate the roll and pitch motion of the vehicle. The corresponding weighting matrix is

$$\mathbf{W} = diag[0.1 \quad 0.1 \quad 0.1 \quad 0 \quad 0 \quad 0.1 \quad 1 \quad 1 \quad 1 \quad 1] \tag{32}$$

Fig. 7 demonstrates the time history of the vehicle position and velocity values. The final yaw angle of the vehicle is approximately 0.22 rad, and as a result of this, the vehicle is subject to large drag force leading to high power consumption. Fig. 8 shows the time history of the manipulator joint position and velocities. As can be seen from the left figure of Fig. 11, the redundancy resolution scheme generated joint position values that violate the joint displacement limits of 180° for the first joint. In addition, the weighted-pseudoinverse generated joint velocity that exceeds the joint-velocity limit of 0.14 rad/s for the first joint. Also, the generated manipulator state values for the second joint cannot be realized by the system since this joint stops functioning after 50 s. Fig. 9 is the time history of the index values. As can be seen, the system is always in a near singular configuration—indicated by the near zero value of the singularity index. The camera index value is negative in the beginning of the operation and the end-effector is not viewable by the human pilot. The camera index becomes positive at the instant of 8 s since the camera's tilt motion is coordinated with the end-effector motion. Note that this happens with no contribution from the vehicle motion since only the camera angle is coordinated with the end-effector motion in the first case study. The yaw index value is generally high demonstrating poor alignment with the ocean current. In light of these observations, the weighted-pseudoinverse alone is seen to be ineffective, or even inapplicable.

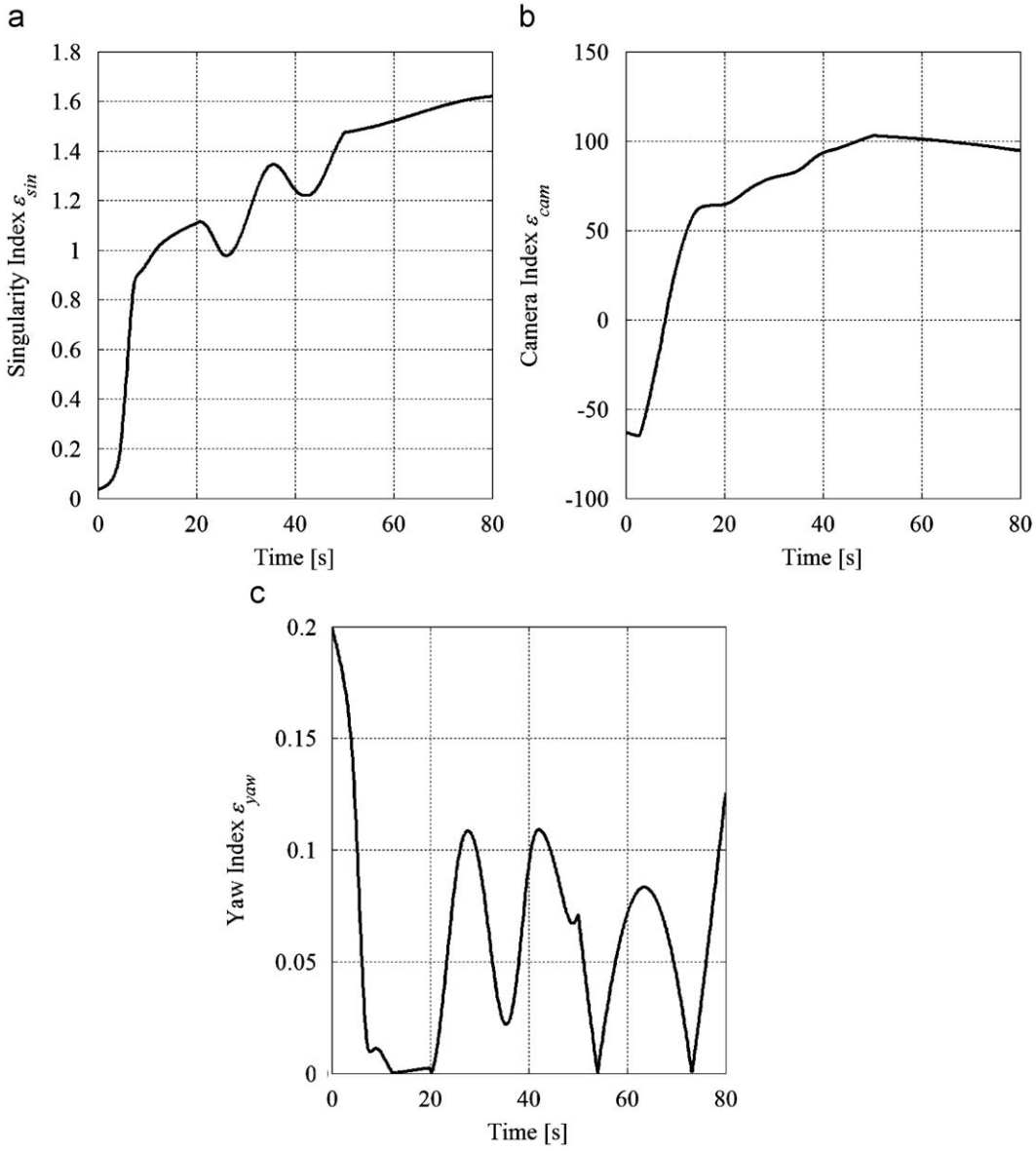


Fig. 12. Time history of singularity, camera, and yaw indices (case study 2).

8.3. Case study 2

In this case study, the proposed fault-tolerant redundancy resolution scheme was implemented for the end-effector manoeuvre described in Section 8.1 Fig. 10 demonstrates the time history of the vehicle position and velocity values. Throughout the manoeuvre the redundancy resolution scheme produced ROV positions and velocities that are realizable, as they show a smooth pattern with no jerk. The redundancy resolver did not induce any motion around the roll and pitch axis of the ROV since the corresponding weighting factors are set to zero. With regards to the joint positions and velocities, the joint positions were kept within their physical joint limits throughout the manoeuvre as shown in Fig. 11. Figs. 12 and 13 are the time history of the index and fuzzy-logic output values, respectively. Finally, Fig. 14 is snapshots of the ROVM manoeuvre for specific moments.

In the first portion of the manoeuvre (0–20 s), the total number of degrees of freedom for the task is eight: the manipulator and the ROV contribute four degrees of freedom each. Note that the ROV's pitch and roll motion do not contribute to the total available degrees of freedom, as they are not controllable. In the first portion, the end-effector is commanded to keep its initial position. Since the demanded task requires only three degrees of freedom (keeping the end-effector position in its initial configuration), the degree of redundancy to be exploited is five.

As can be seen from Fig. 12, the system starts to move from an ill-conditioned configuration as illustrated in Fig. 14a: the singularity index is zero (see Fig. 12a) indicating that the system is in the singular configuration and the reduced dexterity reduces the effectiveness of standard manipulator control; the camera index value is initially negative (see Fig. 12b) since the end-effector is outside the camera view and the pilot does not have a view of the end-effector; and finally, the yaw index value is initially high (see Fig. 12c) demonstrating poor alignment with the ocean current leading to high power consumption due to the poor alignment with the ocean current. With the redundancy resolution scheme turned on, the system immediately starts to move internally, while keeping the end-effector at its initial position, in an attempt to find a preferred system posture as defined by the secondary objectives. As seen in Fig. 12a, the reconfiguration increases the zero singularity index leading to a more dexterous system posture for subsequent tasks. Similarly, the poor camera index value steadily increases from a negative value indicating the end-effector gradually moves towards the camera cone. The camera index value eventually becomes

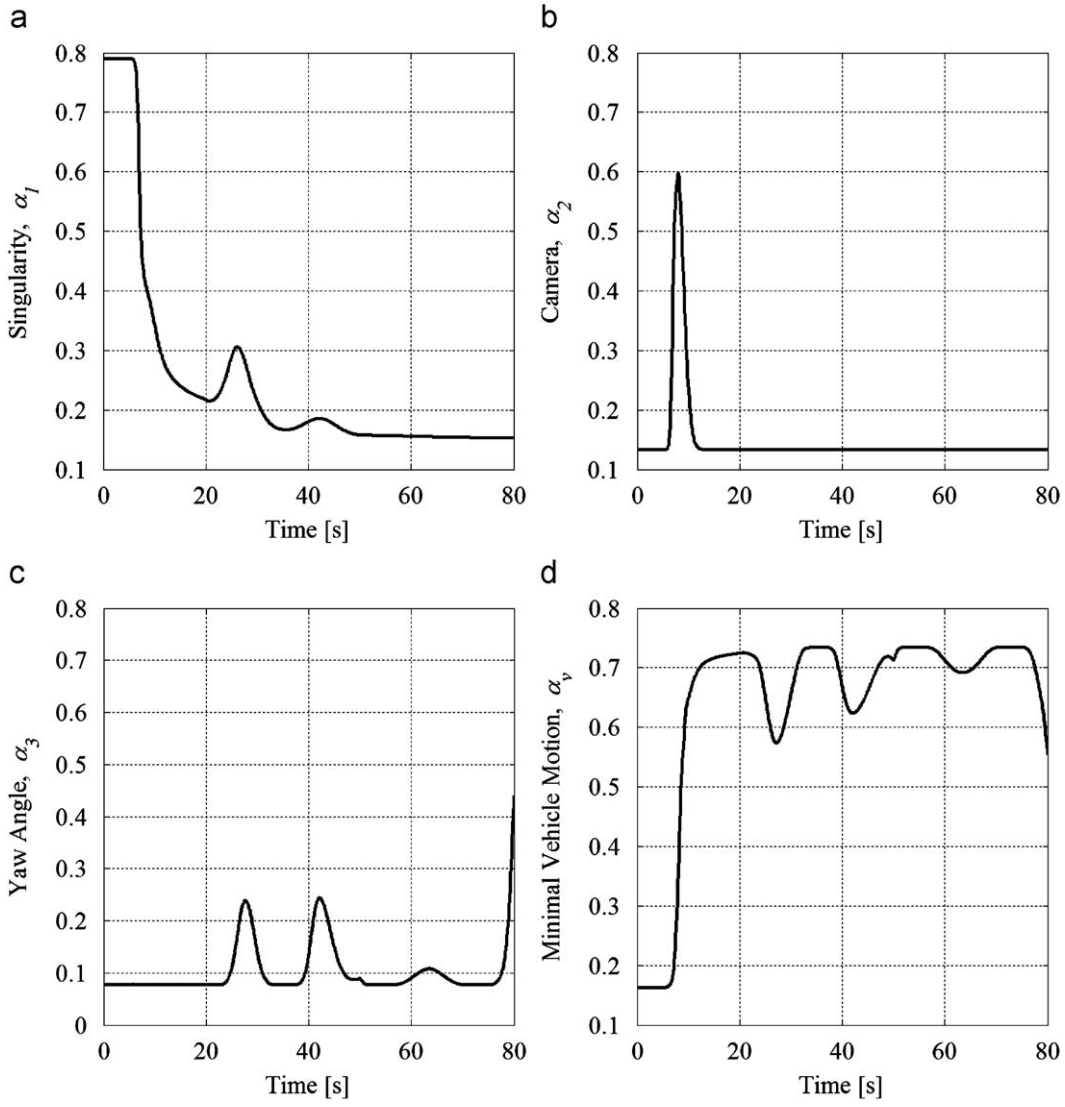


Fig. 13. The fuzzy block decision outputs (case study 2).

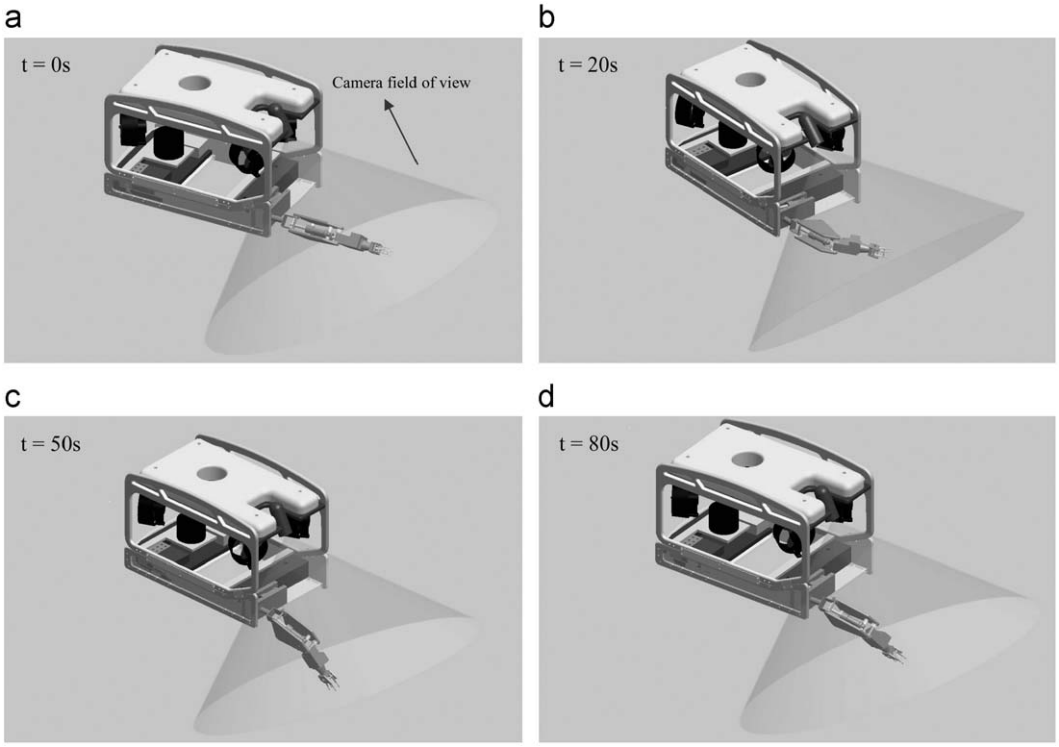


Fig. 14. Snapshots of the ROVM manoeuvre: (a) $t=0$ s, (b) $t=20$ s, (c) $t=50$ s, and (d) $t=80$ s (case study 2).

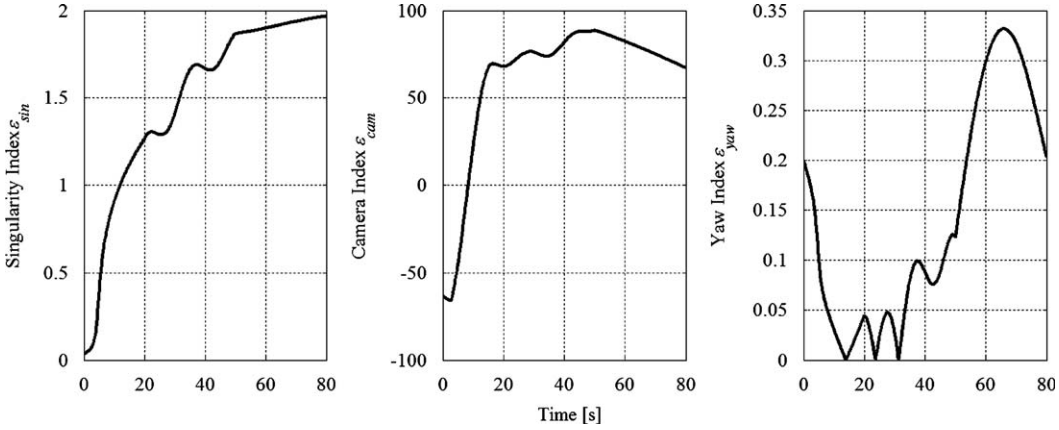


Fig. 15. Index values with singularity avoidance (case study 3).

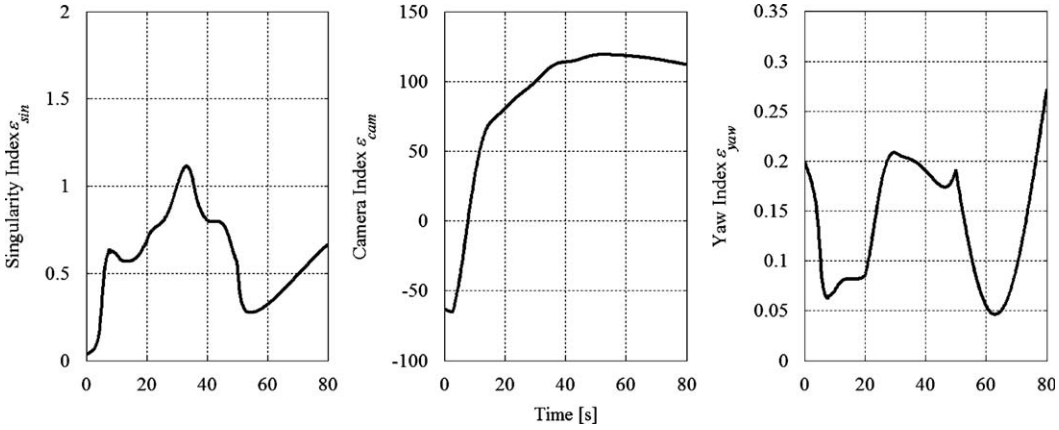


Fig. 16. Index values with singularity avoidance and camera angle (case study 3).

positive at approximately 8 s where the end-effector first enters in the camera view. The yaw index starts with a high value and rapidly drops as the yaw angle of the ROV gradually becomes aligned with the water current. The internal motion continuous to take place until the "best" posture is accomplished. The final reconfigured initial posture at t=20 s is illustrated in Fig. 14b.

The decision outputs of the fuzzy system are reported in Fig. 13. In the beginning of the manoeuvre, the system is in a singular posture, and thus α_1 is high whereas the rest of the α values i.e., α_i, i=2,3 and α_v, are low due to the hierarchy. As the system moves out of the singular configuration, α_1 becomes lower (see Fig. 13a) and α_2 increases (see Fig. 13b) indicating that the highest priority is gradually placed on the camera angle objective. As the end-effector moves into the sight of the camera, the highest priority shifts from the camera angle objective to the minimum vehicle motion objective as can be inferred from the decrease in α_2 and the increase in α_v. Note that during this reorganization of the secondary objective hierarchy, the yaw angle objective is not prioritized despite its higher status in the hierarchy with respect to the camera angle objective. This is due to the fact that yaw angle is fairly aligned with the ocean flow during the priority transition.

In addition, during the first portion of the manoeuvre, the first and third joints were driven to their joint-velocity limits of 0.14 rad/s, demonstrating how the fault-tolerant property can be used to impose velocity constraints on the solution as illustrated in Fig. 11b. The high joint velocity demand can be attributed to the minimal vehicle motion objective; the fuzzy-logic tries to keep the vehicle as stationary as possible.

As a consequence, the initial reconfiguration phase guided by the proposed method makes the system proceed to a joint configuration that satisfies the secondary objectives. Note that since the primary task does not heavily constrain the joint motions, the ROVM establishes a configuration that adheres to the secondary objectives.

During the second portion of the manoeuvre (20–50 s), the pilot attempts to follow a desired locus of end-effector positions and the orientations are unconstrained. As Fig. 12a and b reveal, the singularity and camera index values remain fairly large during this second stage of the manoeuvre indicating the relevant objectives are satisfied successfully. With regards to the yaw index value, some fluctuations occur due to the constant changes in the system yaw orientation with respect to the direction of the ocean flow (see Fig. 12c). The configuration of the system at the instant of 50 s is illustrated in Fig. 14c.

During the third portion of the manoeuvre (50–80 s), the pilot commands not only the end-effector translational velocities but also the end-effector orientations. In addition, the second joint is in failure dropping the available degrees of redundancy to only one. As can be seen from Fig. 11a and b, the redundancy resolver does not demand any motion from the second joint after 50 s, demonstrating the fault-tolerant property. Given the six-DOF task, the manoeuvre execution must involve the ROV mobility; the manipulator alone contributes only four-DOF. This third phase of the manoeuvre illustrates how intelligent coordination of the ROV and the manipulator makes otherwise unrealizable tasks possible while also fulfilling secondary

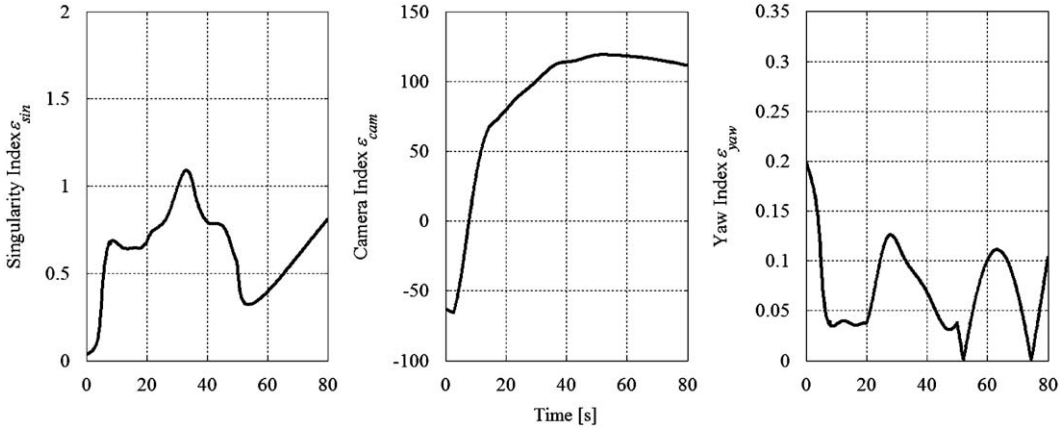


Fig. 17. Index values with singularity avoidance, camera angle and yaw alignment (case study 3).

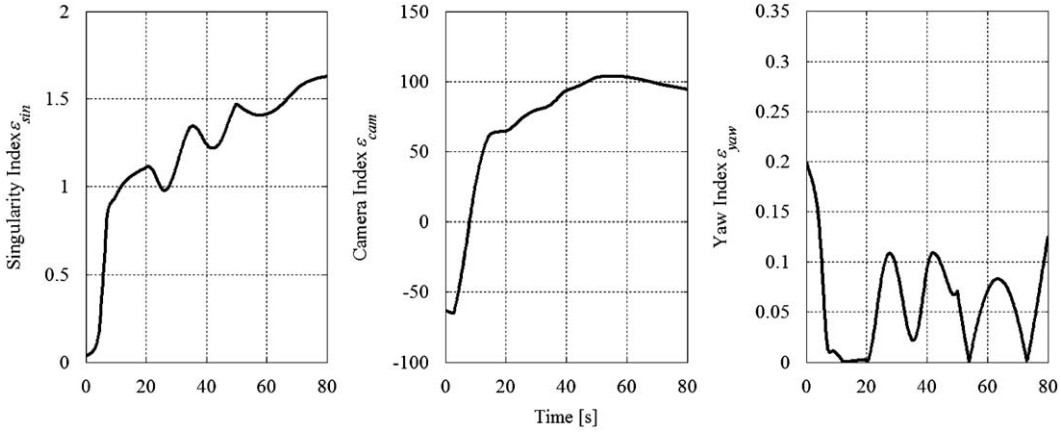


Fig. 18. Index values with the full set of objectives (case study 3).

objectives proving the value of the proposed coordination scheme for ROVM applications. The final configuration of the system is illustrated in Fig. 14d. Given that all the secondary objectives are realized to a satisfactory extent, it can be concluded that system responds quickly enough to keep pace with the desires of the human pilot.

8.4. Case study 3

This section explores the differences in system behaviour observed as the secondary objectives are added in stages. All simulations in case study 3 consider the same end-effector manoeuvre defined in Section 8.1. As in case study 1, no physical break-down fault occurs for the manipulator joints. This allows us to better observe the changes in the system behaviour attributed to the different objectives since the degree of redundancy is higher compared to the faulty-joint scenarios. Four different simulation studies were performed in this section. In the simulation, the joint limit avoidance and the fault-tolerant scheme were implemented, and the secondary objectives, i.e., singularity, camera, yaw angle, and minimal vehicle motion, are added in stages.

In the first simulation, the singularity avoidance objective was considered with the first two rules from the fuzzy rule list given in Section 5.3. In the second simulation, the camera angle objective was included in the objective set with the first four rules from the fuzzy rule list. The yaw angle objective was added to the objective list with the first six rules from the fuzzy rule list Section 5.3 in the third study. And finally, the fourth simulation of this case study 3 considers the full set of secondary objectives. Figs. 15–18 illustrate the corresponding index values for each case study.

Fig. 19 shows the normalized area values under the index curves for each suite of secondary objectives considered. In order to normalize, the area values under the index curves of each objective is determined for each set of objectives, and are divided by the area of the best performance index curve. In Fig. 19, S, C, Y and V denote singularity avoidance, camera view, yaw alignment and minimal vehicle motion, respectively. The vertical axes indicate the normalized performance while the horizontal axis indicates which set of objectives were considered, i.e., S+C corresponds to the case where only the singularity avoidance and camera view objectives are considered.

As can be seen in Fig. 19a bar S, the singularity avoidance performance is the best when only this particular objective was considered. A significant decrease in the singularity performance occurred when the camera angle objective was included in the objective set (Fig. 19a, bar S+C). This can be attributed to the fact that some of the available redundancy is committed to improve the camera view objective. This improvement can be seen by comparing bar S with bar S+C in Fig. 19b. With the inclusion of the new objectives, the singularity avoidance performance slightly increased (see Fig. 19a, bar S+C, S+C+Y and S+C+Y+V), suggesting that the motion dictated by the combination of yaw and minimal vehicle motion objectives also improved the singularity avoidance performance. This is not true for the camera view, as the inclusion of new objectives slightly worsened the performance index (see Fig. 19b, bar S+C+Y and S+C+Y+V).

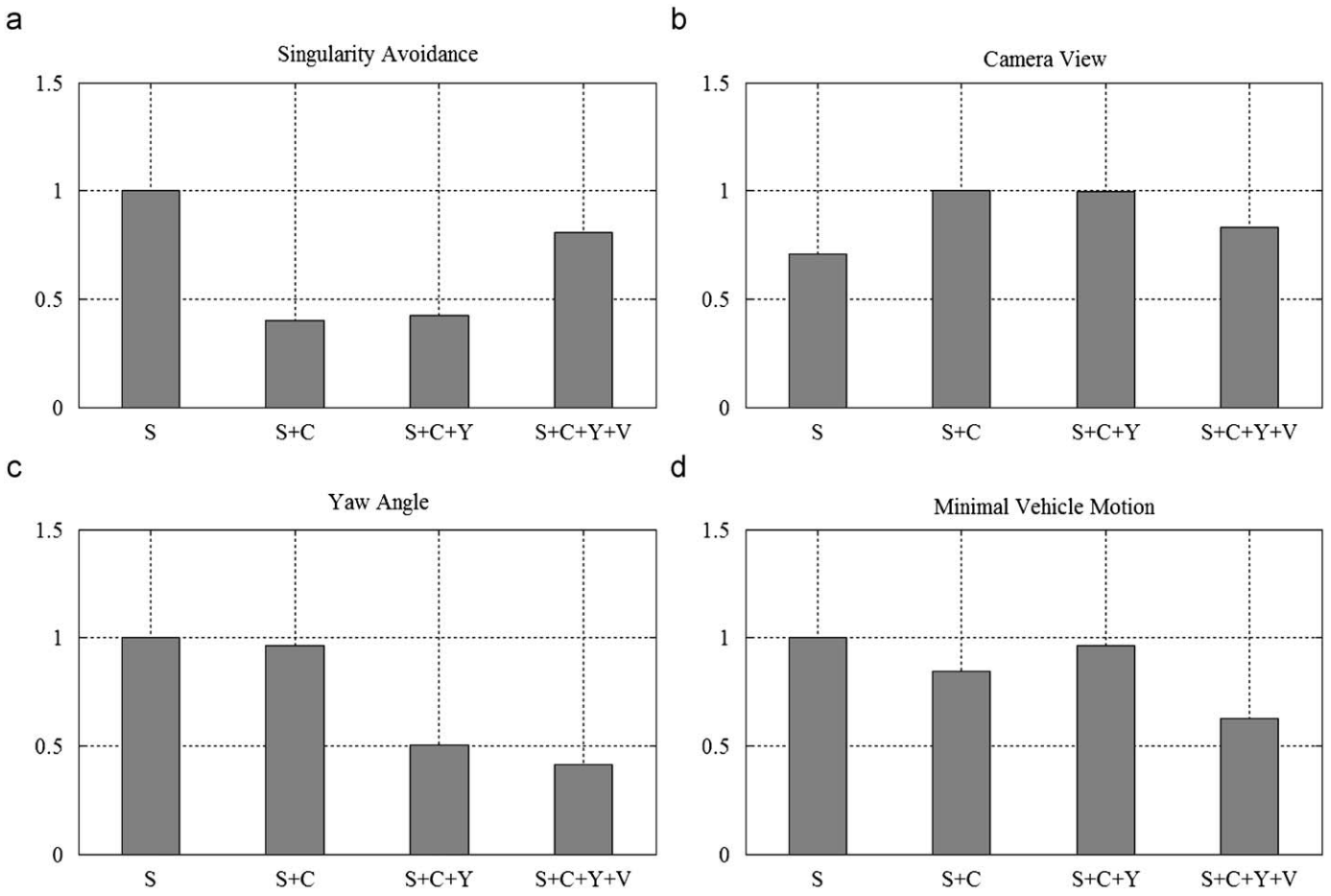


Fig. 19. Normalized area values under the index curves of each objective with different set of secondary objectives. In the two top plots, higher values indicate better performance whereas in the two bottom plots, lower values indicate better performance (case study 3).

With regards to the yaw and minimal vehicle motion, their performances are illustrated in Fig. 19c and d, respectively. As opposed to Fig. 19a and b, lower values indicate better performance. As can be seen from Fig. 19c, when the yaw angle objective was excluded from the secondary objective set, the system exhibits very poor alignment with the ocean current (see bar S and S+C). A significant performance improvement was achieved in the yaw angle alignment by adding this particular objective into the consideration (see bar S, S+C and S+C+Y). A further performance achievement was obtained for the S+C+Y+V case indicating that the minimal vehicle objective incurred motion around the yaw that improves the yaw alignment.

The best performance for the minimal vehicle motion objective was obtained with the inclusion of this particular objective in the objective set as illustrated in Fig. 19d. All the case studies performed indicate the system response is fast enough with the proposed scheme, as the secondary objectives were accomplished to a satisfactory extent.

9. Conclusion

A coordinated ROVM operation scheme has been proposed. A fault-tolerant fuzzy-based redundancy resolution method has been proposed to distribute the human pilot end-effector command over the ROV and the manipulator while satisfying a hierarchy of secondary objectives. Within the redundancy resolution, the GPM has been merged with a fuzzy technique such to produce an evolving hierarchy of secondary objectives that reacts to secondary objective performance index fluctuations that are easily evaluated in an on-line implementation. It is concluded that the fuzzy-logic implementation provides a very effective means to prioritize the secondary task in a satisfactory manner. In addition, the resulting method has been extended to include a fault-tolerant property that has been demonstrated to be a versatile tool for the redundancy resolution of ROVM systems: it can be used to tolerate faulty joints, to impose dynamic joint-velocity constraints, and to ensure that the GPM works when some of the diagonal elements of the weighting matrix become zero during ROVM manoeuvres. The imposition of the velocity constraints provide more controllable ROVM motion. A simulation case study of the FALCON^{TM} ROV with a 4-DOF Hydrolek^{TM} manipulator demonstrated that detailed spatial end-effector manoeuvres that are otherwise not possible can be completed in real-time through the coordination of the ROV and the manipulator with the fault-tolerant capacity. The on-line nature of the proposed scheme makes it suitable for remote systems where the desired end-effector state and secondary task values are not known *a priori*.

Acknowledgments

The authors would like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for providing financial support for this work. Also the authors wish to extend their thanks to Suboceanic Sciences Canada Ltd. in Duncan, Canada, for their participation and support in the research.

Appendix A. Proof of the decrease of the multiple-criteria objective function through the GPM approach

In the current work, a multiple-criteria GPM approach is defined as

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{x}} + \lambda(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\mathbf{p} \tag{A.1}$$

where the arbitrary vector $\mathbf{p}$ is defined as the sum of the normalized gradients of scalar objective functions, i.e., $\mathbf{p} = \sum_{j=1}^{s} \alpha_j \nabla h_j / \|\nabla h_j\|_2$. Consider the case in which the minimization of the scalar objective function, $z(\mathbf{q})$, is desired, i.e., $\min\left(z(\mathbf{q}) = \sum_{j=1}^{s} h_j\right)$, and that λ is a strictly negative number. Therefore, when $\dot{\mathbf{x}} = 0$, it can be shown that (A.1) guarantees the decrease of the scalar objective function:

$$\dot{z} = \frac{\partial z}{\partial \mathbf{q}}\dot{\mathbf{q}} = \nabla z\dot{\mathbf{q}} = \lambda \nabla z^T(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\nabla z = \lambda \nabla z^T(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})^T\nabla z \le 0 \tag{A.2}$$

since $(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})$ is positive semi-definite. Note that in (A.2), the idempotency and symmetry of $(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})$ is used, i.e., $(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}) = (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})^2$ and $(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}) = (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})^T$. Although, for nonzero $\dot{\mathbf{x}}$, (A.2) does not imply the nonpositiveness of $\dot{z}$, the second term of (A.1) still works to reduce the objective function z. This example also illustrates that the positive semi-definitiveness of the projection operator is crucial to make the scalar objective function decrease (Nakamura, 1991).

Appendix B. Proof of the decrease of the multiple-criteria objective function through weighted multiple-criteria GPM approach

The weighted GPM is defined as

$$\dot{\mathbf{q}} = \mathbf{J}_w^{\dagger}\dot{\mathbf{x}} + \lambda(\mathbf{I} - \mathbf{J}_w^{\dagger}\mathbf{J})\mathbf{W}^{-1}\mathbf{p} \tag{B.1}$$

It must be shown that the scalar objective function $\dot{z}$ defined in Appendix A decreases, i.e., $\dot{z} \le 0$. Without loss of generality, assume that $\dot{\mathbf{x}} = 0$ as in Appendix A. Therefore, the time derivative of the objective function z with the weighted GPM can be given as

$$\dot{z} = \frac{\partial z}{\partial \mathbf{q}}\dot{\mathbf{q}} = \nabla z\dot{\mathbf{q}} = \lambda \nabla z^T(\mathbf{I} - \mathbf{J}_w^{\dagger}\mathbf{J})\mathbf{W}^{-1}\nabla z \tag{B.2}$$

Since the weighting matrix is defined as a diagonal, positive-definite matrix, it can be decomposed as $\mathbf{W}^{-1} = \mathbf{U}^T\mathbf{U}$ (Watkins, 2002). Substituting $\mathbf{J}_w^{\dagger} = \mathbf{W}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^T)^{-1}$ and $\mathbf{W}^{-1} = \mathbf{U}^T\mathbf{U}$ into (B.2) yields:

$$\dot{z} = \lambda \nabla z^T\left(\mathbf{U}^T\mathbf{U} - \mathbf{U}^T\mathbf{U}\mathbf{J}^T(\mathbf{J}\mathbf{U}^T\mathbf{U}\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{U}^T\mathbf{U}\right)\nabla z = \lambda \nabla z^T\mathbf{U}^T(\mathbf{U} - \mathbf{U}\mathbf{J}^T(\mathbf{J}\mathbf{U}^T\mathbf{U}\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{U}^T\mathbf{U})\nabla z \tag{B.3}$$

Substituting $\mathbf{P} = \mathbf{J}\mathbf{U}^T$ and $\mathbf{y} = \mathbf{U}\nabla z$ yields

$$\dot{z} = \lambda y^T(\mathbf{U} - \mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1}\mathbf{P}\mathbf{U})\nabla z = \lambda \mathbf{y}^T\left(\mathbf{U}\nabla z - \mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1}\mathbf{P}\mathbf{U}\nabla z\right) = \lambda \mathbf{y}^T(\mathbf{I} - \mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1}\mathbf{P})\mathbf{U}\nabla z = \lambda \mathbf{y}^T(\mathbf{I} - \mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1}\mathbf{P})\mathbf{y} \tag{B.4}$$

Given that Moore–Penrose pseudoinverse of $\mathbf{P}$ is defined as $\mathbf{P}^{\dagger} = \mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1}$, (B.4) becomes

$$\dot{z} = \lambda \mathbf{y}^T(\mathbf{I} - \mathbf{P}^{\dagger}\mathbf{P})\mathbf{y} \le 0 \tag{B.5}$$

since $\lambda \le 0$ and $(\mathbf{I} - \mathbf{P}^{\dagger}\mathbf{P})$ is idempotent and symmetrical, and therefore positive semi-definite (Nemec, 1997).

References

Antonelli, G., Chiaverini, S., 2003. Fuzzy redundancy resolution and motion coordination for underwater vehicle–manipulator systems. IEEE Transactions on Fuzzy Systems 11 (1), 109–120.

Antonelli, G., Chiaverini, S., 1998. Task priority redundancy resolution for underwater-manipulator systems. In: Proceedings of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, pp. 768–733.

Chan, T.F., Dubey, R.V., 1995. A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. IEEE Transactions on Robotics and Automation 11 (2), 286–292.

Chen, W., Chen, I. M., Wang, T., 1999. Kinematic fault tolerant control for redundant robot based on joint velocities redistribution. In: Proceedings of the 9th International Conference on Advanced Robotics, '99 ICAR, Tokyo, Japan, pp. 397–402.

Chen, W., Zhang, Q., Yang, Z., Gruver, W.A., 1995. Optimizing multiple performance criteria in redundant manipulators by subtask-priority control. In: Intelligent Systems for the 21st Century, IEEE International Conference on Systems, Man and Cybernetics, vol. 3, 1995, pp. 2528–2533.

Chiaverini, S., 1997. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. IEEE Transactions on Robotics and Automation 13, 398–410.

Craig, J.J., 1989. Introduction to Robotics, second ed Addison-Wesley.

Driankov, D., Hellendoorn, H., Reinfrank, M., 1995. An Introduction to Fuzzy Control. Springer-Verlag.

Euler, J.A., Dubey, R.V., Babcock, S.M., 1989. Self-motion determination based on actuator velocity bounds for redundant manipulators. Journal of Robotic Systems 6 (4), 417–425.

Han, J., Chung, W.K., 2007. Redundancy resolution for underwater vehicle–manipulator systems with minimizing restoring moments. In: Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, pp. 3522–3527.

Li, L., Gruver, W.A., Zhang, Q., Yang, Z., 2001. Kinematic control of redundant robots and the motion optimizability measure. IEEE Transactions on Systems, Man, and Cybernetics 31 (1), 155–160.

Liegeois, A., 1977. Automatic supervisory control of the configuration and behaviour of multibody mechanisms. IEEE Transactions on Systems, Man, and Cybernetics 7, 868–871.

Marani, G., Yuh, J., Choi, S.K., 2006. Autonomous manipulation for an intertvention AUV. In: Roberts, G., Sutton, B. (Eds.), Guidance and Control of Unmanned Marine Vehicles. IEE's Control Engineering Series, pp. 217–237.

Nakamura, Y., 1991. Advanced Robotics: Redundancy and Optimization. Addison-Wesley, Reading, MA.

Nemec, B., 1997. Pseudoinverses and null space velocity controller. In: Proceedings of IEEE International Conference on Intelligent Engineering Systems. Budapest, Hungary, pp. 107–111.

Park, J., 1999. Analysis and control of kinematically redundant manipulators: an approach based on kinematically decoupled joint space decomposition. Ph.D. Thesis, Pohang University of Science and Technology (POSTECH), South Korea.

Podder, T.K., Sarkar, N., 2004. Unified dynamics-based motion planning algorithm for autonomous underwater vehicle–manipulator systems. Robotica 22, 117–128.

Sarkar, N., Podder, T.K., 2001. Motion coordination of underwater vehicle–manipulator systems subject to drag optimization. In: Proceedings of the 1999 IEEE International Conference on Robotics and Automation, Detroit, MI, USA, pp. 387–392.

Santos, C.H.F., Guenther, R., Martins, D., Pieri, E.R., 2006a. Virtual kinematic chains to solve the underwater vehicle–manipulator systems redundancy. Journal of the Brazilian Society of Mechanical Sciences and Engineering 28 (3), 354–361.

Santos, C.H., Bittencourt, G., Guenther, R., 2006b. Motion coordination for underwater vehicle–manipulator systems using a fuzzy hybrid strategy. In: Proceedings of the International Conference on Intelligent Robots and Systems, Beijing, China, pp. 3018–3023.

Soylu, S., Buckham, B.J., Podhorodeski, R.P., 2007a. A fault-tolerant fuzzy logic based redundancy resolution method for mobile underwater manipulators. In: Proceedings of the OCEANS 2007 MTS/IEEE, Vancouver, BC, Canada.

Soylu, S., Buckham, B.J., Podhorodeski, R.P., 2007b. Dexterous task-priority based redundancy resolution for underwater–manipulator systems. CSME Transactions 4 (31), 519–533.

Watkins, D.S., 2002. Fundamentals of Matrix Computations, second ed John Wiley & Sons.

Yoshikawa, T., 1990. Foundations of Robotics: Analysis and Control. MIT Press.

Yoshikawa, T., 1985. Manipulability of robotic mechanisms. International Journal of Robotic Research 4 (2), 3–9.

Appendix G. Exploiting Redundancy in Underwater Vehicle-Manipulator Systems

(2009, International Journal of Offshore and Polar Engineering, volume 19, issue 2, pages 115-123)

International Journal of Offshore and Polar Engineering (ISSN 1053-5381)

Vol. 19, No. 2, June 2009, pp. 115–123

Exploiting Redundancy in Underwater Vehicle-Manipulator Systems

Serdar Soylu, Bradley J. Buckham* and Ron P. Podhorodeski
Department of Mechanical Engineering, University of Victoria
Victoria, British Columbia, Canada

The current work focuses on the development of a comprehensive scheme for the coordinated control of remotely operated vehicle–manipulator (ROVM) systems, and it proposes a novel mode for their operation. The proposed scheme consists of 2 main stages: redundancy resolution, and and robust model based control. In the redundancy resolution stage, the end-effector input commanded by a human pilot is distributed over the vehicle and manipulator. The redundant degrees of freedom (DOF) are used to accomplish secondary objectives. To this end, the Gradient Projection Method (GPM) is merged with a fuzzy logic based weighting scheme. Regarding the control of the system, a dynamic model is derived using the energy-based quasi–Lagrange approach. As opposed to a classic Lagrangian derivation, the quasi–Lagrange approach generates the equations in terms of body-fixed frames. As well, an adaptive sliding mode controller is implemented that constantly compensates for unknown dynamics throughout the vehicle-manipulator system. To demonstrate the efficacy of the scheme, a numerical case study is performed. Results illustrate that a complex end-effector spatial maneuver defined by a single 6-DOF pilot input can be accomplished with a 4-DOF manipulator mounted on a small ROV.

INTRODUCTION

Underwater remotely operated vehicles (ROVs) equipped with robotic manipulators are used in a variety of underwater applications, such as oil and gas extraction, installation of underwater telecommunication cables, and inspections and maintenance of offshore structures. The combined system is referred to as an underwater remotely operated vehicle-manipulator (ROVM). To date, the motions of the ROV and the manipulator are guided independently by a human pilot on a surface support vessel through a long slender tether that provides power and telemetry. In current practice, the desired manipulator joint motions are created using a teleoperated master-slave arm configuration. This mode of operation depends on the ability of the ROV to hold station, and it decouples the manipulator and ROV degrees of freedom (DOF). In most cases, pilots attempt to eliminate any ROV motion by lodging the ROV against an immovable object using thrusters or by an additional arm. This simplifies the pilot's task, but it eliminates the redundancy inherent in the ROVM system. (In addition to the manipulator's DOF, the ROV itself contributes 6 active DOF.) Further, for smaller ROVs, the serial manipulator is usually underactuated for 6 DOF tasks, and so the elimination of the ROV DOF results in a very constrained end-effector workspace. If the ROV DOF are used during an end-effector task, then there are many possible ways to achieve a desired end-effector motion. A redundancy resolver determines the optimal combination of ROV and arm motions that yields the required end-effector motion while also realizing the additional secondary objectives.

The implementation of redundancy resolution methods within ROVM systems has been documented in only a few existing works. A singularity robust task-priority redundancy resolution was shown to be useful for an ROVM by Chiaverini (1997) due to its multitask capabilities. In Sarkar and Podder (2001), the kinematic redundancy is utilized to minimize the total hydrodynamic drag forces experienced by a ROVM system in an effort to reduce the energy consumption. In Antonelli and Chiaverini (2003), the singularity robust task-priority redundancy resolution is merged with a fuzzy technique to resolve the ROV-manipulator coordination. It was shown that fuzzy logic is an effective means for handling multiple kinematic constraints. A fault-tolerant redundancy resolution method was proposed by Soylu et al. (2007). In that work, the Gradient Projection Method (GPM) was used to blend a series of secondary objectives using a fuzzy weighting scheme with a fault-tolerant property. The current study follows Soylu et al. (2007).

While the redundancy resolution scheme translates the pilot's intent into desired ROVM joint rates, it falls on a robust control strategy to realize these joint rates. Robust control methods generally depend on dynamic models of the system. Existing dynamics models of ROVM include that of Ioi and Itoh (1990), who extended the Newton-Euler formulation to include the requisite hydrodynamics terms. Sagatun and Fossen (1991) derived the equations of motion for an ROVM system using the Lagrange method. McMillan et al. (1995) extended the articulated-body formulation to develop a computationally efficient dynamic simulation of an ROVM system. Tarn et al. (1996) developed a dynamic model of an ROVM based on Kane's method.

For multi-body systems such as ROVMs, the Lagrange approach is preferred since it provides the equations of motion in an analytical form—a requirement for any model-based controller. A drawback of the Lagrange approach is that it yields the equations of motion in terms of generalized coordinates specific to an inertial frame. However, for ROVM dynamics modeling, it is convenient to work in the body-fixed frames. To this end, the quasi-Lagrange approach is used to model the system dynamics in the current work. The differences between the classic Lagrangian and the quasi-Lagrangian will be elaborated in the dynamic modeling section.

Controller development has been largely applied to ROVs, and it is rare that the manipulator DOF are considered. The unified control of ROVM systems has been addressed by Antonelli and Chiaverini (1998), who designed an adaptive controller for an

*ISOPE Member.

Received November 6, 2008; revised manuscript received by the editors April 9, 2009. The original version (prior to the final revised manuscript) was presented at the 18th International Offshore and Polar Engineering Conference (ISOPE-2008), Vancouver, July 6–11, 2008.
KEY WORDS: Underwater vehicle, manipulator, redundancy, nonlinear control.

ROVM. A robust control technique based on the singular perturbation method was demonstrated by Canudas de Wit et al. (2000). The computed torque control was applied by Sarkar and Podder (2001) on an ROVM system. A model-based sliding mode controller was proposed by Antonelli (2003). Xu et al. (2005) designed a sliding-mode controller with a fuzzy logic-based tuning of the control gains.

The sliding-mode approach is an effective means of controlling an ROVM, largely due to its ability to tolerate imprecision in the dynamics model. However, one major drawback of the sliding-mode approach is the high-frequency control action (chattering). In classic land-based applications, chattering concerns include premature wear in actuators and the excitation of unmodelled high-frequency dynamics. For underwater vehicle-manipulator systems, the high-frequency system response can be compromised by the hydrodynamics, including added mass effects, and it is preferable to exert a sustained smooth control rather than infrequent discontinuous control. To eliminate or reduce chattering, various methods have been presented, such as the boundary layer method (Slotine and Shastry, 1983; Yoerger and Slotine, 1985) and the disturbance compensation-based chattering elimination method (Elmali and Olgac, 1992; Zeinali and Notash, 2004a). In the current study, the work of Zeinali and Notash (2004a) is adapted to the ROVM paradigm to eliminate the chattering problem. In Zeinali and Notash (2004a), the discontinuous term of a conventional sliding-mode controller is replaced with an adaptive estimate of the dynamic uncertainties. The same approach was used by Soylu et al. (2008) for the trajectory control of ROV.

REDUNDANCY RESOLUTION

For the redundancy resolution task, the work of Soylu et al. (2007) is followed in the current work.

Gradient Projection Method

The differential kinematic relationship between the task-space velocities $\dot{\boldsymbol{x}} \in \mathbb{R}^m$, which corresponds to the pilot command to the end-effector, and the generalized velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ is given by:

$$\dot{\boldsymbol{x}} = \mathbf{J}\dot{\boldsymbol{q}} \tag{1}$$

where $\mathbf{J} \in \mathbb{R}^{m \times n}$ is the Jacobian matrix, $\dot{\boldsymbol{q}} = [\dot{\boldsymbol{q}}_1^T \ \dot{\boldsymbol{q}}_2^T \ \dot{\boldsymbol{q}}_3^T]^T \in \mathbb{R}^n$ is the generalized coordinate vector, consisting of the liner velocity vector $\dot{\boldsymbol{q}}_1 = [\dot{x} \ \dot{y} \ \dot{z}]^T$, Euler rates (Roll-Pitch-Yaw) $\dot{\boldsymbol{q}}_2 = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ of the vehicle, and joint rates $\dot{\boldsymbol{q}}_3 = [\dot{q}_1 \ \dot{q}_2 \ \dots \ \dot{q}_{n-6}]^T$ of the manipulator. For a kinematically redundant manipulator, $(n > m)$, and there is an infinite set of joint rate solutions that can complete the desired end-effector motion, $\dot{\boldsymbol{x}}$.

The following solution to Eq. 1 (redundancy resolution) is proposed:

$$\begin{aligned}\dot{\boldsymbol{q}} &= \dot{\boldsymbol{q}}_p + \dot{\boldsymbol{q}}_h \\ &= \mathbf{J}_w^{\dagger}\dot{\boldsymbol{x}} + \lambda(\mathbf{I} - \mathbf{J}_w^{\dagger}\mathbf{J})\mathbf{W}^{-1}\left(\sum_{j=1}^{s} \alpha_j \nabla h_j / \|\nabla h_j\|_2\right).\end{aligned} \tag{2}$$

where $\dot{\boldsymbol{q}}_p = \mathbf{J}_w^{\dagger}\dot{\boldsymbol{x}}$ is the particular solution with:

$$\mathbf{J}_w^{\dagger} = \mathbf{W}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^T)^{-1} \tag{3}$$

being the right weighted Moore-Penrose pseudoinverse (Watkins, 2002). The weighting matrix is in the form of $\mathbf{W}^{-1} = \mathrm{diag}(w_1, w_2, \dots, w_n)$, and is a positive definite matrix of weight factors for each DOF in a robotic system. The particular solution corresponds to the weighted minimum-norm solution associated with Eq. 1 that minimizes $\dot{\boldsymbol{q}}^T\mathbf{W}\dot{\boldsymbol{q}}$ subject to $\dot{\boldsymbol{x}} = \mathbf{J}\dot{\boldsymbol{q}}$. The term $\dot{\boldsymbol{q}}_h$ is the homogeneous solution that satisfies $\mathbf{0} = \mathbf{J}\dot{\boldsymbol{q}}_h$ where s is the total number of kinematic criteria, and α_j is a weight that determines the extent to how much emphasis is placed on each criterion h_j with respect to the others. The limiting factors on the choice of α_j can be found in Euler et al. (1989). In order to prevent one single criterion from excessively dominating the solution, the normalized gradient is used in Eq. 2. Note that when the norm of the gradient is equal to zero, Eq. 2 can not be used. This situation occurs when the objective function is in a flat region, and it can not be further optimized. Thus, if the norm of the gradient is equal to zero, the corresponding objective can be eliminated from consideration. Desired position values can be found by $\boldsymbol{q} = \int \dot{\boldsymbol{q}}\, dt$.

In this scheme, the primary objective is to follow the desired end-effector motion $\dot{\boldsymbol{x}}$ that is commanded by a higher-level human pilot motion using a 6-DOF master arm. Regarding the secondary objectives h_j, they include, but are not limited to:

- avoiding manipulator joint limits;
- singularity avoidance;
- keeping the end-effector inside the camera view;
- minimizing the ROV drag force (otherwise known as weather-vaning);
- minimizing the ROV motion.

To pursue the joint limit avoidance and the minimal vehicle motion objectives, the weighting matrix $\mathbf{W}$ is used. In this implementation, $w_i, i = 1 \dots 6$ are reserved for the minimal vehicle motion objective, while $w_i, i = 7 \dots n$ are reserved for the joint limit avoidance. The singularity avoidance, the camera view, and the drag force minimization objectives are included through the GPM. The importance of each objective is determined by a low-level artificial pilot. To this end, the artificial pilot accepts index values associated with the secondary objectives, and maps them into a weighting factor $\alpha_j \in [0, 1]$ based on if-then type fuzzy rules that reflect an expert human pilot's knowledge. Greater α_j values indicate a higher demand, while smaller α_j values indicate a lower demand for that particular objective. This rule interpretation process by the artificial pilot is performed based on the Mamdani Fuzzy Interference method (Driankov et al., 1996). The resulting weighting schedule yields a self-motion (null-space motion) that emulates how a skilled operator would utilize the redundancy of the ROVM to achieve the secondary objectives. A more detailed explanation of this scheme can be found in Soylu et al. (2007).

UNIFIED DYNAMIC MODELLING OF ROVM SYSTEMS

The derivation of the equations of motion for an ROVM system is more convenient when they are defined in terms of state values expressed in the body-fixed frames since on-board sensors and actuators measure and create rates specific to the body fixed frame (Sarkar and Podder, 2001; Kim et al., 2003). To this end, a quasi-Lagrange approach (Meirovitch and Stemple, 1995) can be used. The quasi-Lagrange approach was originally applied to the ROVM modeling by Sarkar and Podder (2001) and Kim et al. (2003).

Quasi-Lagrange Equations

The quasi-Lagrange equations are an extension of Kirchoff's equations (Meirovitch, 1990) and can be defined as:

$$\begin{aligned}
&\frac{d}{dt}\left(\frac{\partial \bar{L}}{\partial \boldsymbol{v}_1}\right)+\boldsymbol{v}_2\times\frac{\partial \bar{L}}{\partial \boldsymbol{v}_1}-\mathbf{J}_1^T\frac{\partial \bar{L}}{\partial \boldsymbol{q}_1}=\boldsymbol{\rho}_1\\
&\frac{d}{dt}\left(\frac{\partial \bar{L}}{\partial \boldsymbol{v}_2}\right)+\boldsymbol{v}_2\times\frac{\partial \bar{L}}{\partial \boldsymbol{v}_2}+\boldsymbol{v}_1\times\frac{\partial \bar{L}}{\partial \boldsymbol{v}_1}-\mathbf{J}_2^T\frac{\partial \bar{L}}{\partial \boldsymbol{q}_2}=\boldsymbol{\rho}_2\\
&\frac{d}{dt}\left(\frac{\partial \bar{L}}{\partial \boldsymbol{v}_3}\right)-\frac{\partial \bar{L}}{\partial \boldsymbol{q}_3}=\boldsymbol{\rho}_3
\end{aligned} \tag{4}$$

where $\bar{L}=\overline{T}(\boldsymbol{q},\boldsymbol{v})$ is the kinetic energy of the system in terms of the state values expressed in the body-fixed reference frames. (This also corresponds to the quasi-Lagrangian.) The vectors:

$$\begin{aligned}
&\mathbf{v}=[\mathbf{v}_1^T\ \ \mathbf{v}_2^T\ \ \mathbf{v}_3^T]^T,\quad \mathbf{v}_1=[u\ \ v\ \ w]^T,\\
&\mathbf{v}_2=[p\ \ q\ \ r]^T,\quad \mathbf{v}_3=[\dot{q}_1\ \ \dot{q}_2\ \ \ldots\ \ \dot{q}_{n-6}]^T
\end{aligned} \tag{5}$$

are the body-fixed velocities; $\mathbf{v}_1\in\mathbb{R}^{3\times 1}$ and $\mathbf{v}_2\in\mathbb{R}^{3\times 1}$ are the body-fixed translational and rotational velocity vectors of the ROV, respectively; and $\mathbf{v}_3\in\mathbb{R}^{(n-6)\times 1}$ contains the manipulator joint rates.

The Kirchhoff equations given in Eq. 4 can be written in a matrix form as:

$$\frac{d}{dt}\left(\frac{\partial \bar{L}}{\partial \boldsymbol{v}}\right)+\mathbf{A}\frac{\partial \bar{L}}{\partial \boldsymbol{v}}-\mathbf{B}^T\frac{\partial \bar{L}}{\partial \boldsymbol{q}}=\boldsymbol{\tau} \tag{6}$$

where:

$$\mathbf{A}=\begin{bmatrix} S(\boldsymbol{v}_2) & 0 & 0\\ S(\boldsymbol{v}_1) & S(\boldsymbol{v}_2) & 0\\ 0 & 0 & 0\end{bmatrix},\qquad \mathbf{B}=\begin{bmatrix} \mathbf{R}(\boldsymbol{q}_2) & 0 & 0\\ 0 & \mathbf{T}(\boldsymbol{q}_2) & 0\\ 0 & 0 & \mathbf{I}\end{bmatrix},\qquad \boldsymbol{\tau}=\begin{bmatrix}\boldsymbol{\tau}_1\\ \boldsymbol{\tau}_2\\ \boldsymbol{\tau}_3\end{bmatrix} \tag{7}$$

and $\boldsymbol{\tau}=\mathbf{B}^T\boldsymbol{\rho}$ with $\boldsymbol{\rho}^T=[\boldsymbol{\rho}_1^T\ \ \boldsymbol{\rho}_2^T\ \ \boldsymbol{\rho}_3^T]^T$. The term $\mathbf{R}(\boldsymbol{q}_2)$ is the orthogonal rotation matrix from the vehicle's body-fixed frame to the inertial frame. Likewise, the term $\mathbf{T}(\boldsymbol{q}_2)$ is the nonlinear transformation matrix that maps the body-fixed values into the Euler rates. The term $S(\cdot)$ is the skew-symmetric operator.

The kinetic energy of the ROVM can be written in terms of body-fixed frames as:

$$\overline{T}=\frac{1}{2}\sum_{i=1}^{n}({v_{c,i}}^T\mathbf{M}_{RB_M}v_{c,i}+{v_{c,i}}^T\mathbf{M}_{AM_M}v_{c,i}) \tag{8}$$

where $\frac{1}{2}\sum_{i=1}^{n}{v_{c,i}}^T\mathbf{M}_{RB_M}v_{c,i}$ is the kinetic energy of the ROVM due to the rigid body, and $\frac{1}{2}\sum_{i=1}^{n}{v_{c,i}}^T\mathbf{M}_{AM_M}v_{c,i}$ is the kinetic energy of the ROVM due to the added mass; $v_{c,i}$ represents the velocity of the center of mass of the ith link starting from the vehicle ($i=0$) to the end effector ($i=n$). These velocity expressions should be defined in terms of the ROVM state vectors $\boldsymbol{q}$ and $\boldsymbol{v}$. To this end, the Newton-Euler recursive algorithm (Craig, 1989) can be used. The terms $\mathbf{M}_{RB,i}\in\mathbb{R}^{6\times 6}$ and $\mathbf{M}_{AM,i}\in\mathbb{R}^{6\times 6}$ denote the mass and added mass matrix of a relevant rigid body within the ROVM system. Eq. 8 can be written as:

$$\bar{L}=\overline{T}=\frac{1}{2}\boldsymbol{v}^T\mathbf{M}\boldsymbol{v} \tag{9}$$

where $\mathbf{M}\in\mathbb{R}^{m\times n}$ is the mass matrix of the ROVM system. Inserting Eq. 9 into Eq. 6 yields the closed-form equations of motion:

$$\mathbf{M}\dot{\boldsymbol{v}}+\mathbf{C}_s(\boldsymbol{q}\ \ \boldsymbol{v})=\boldsymbol{\tau} \tag{10}$$

where $\mathbf{C}_s\in\mathbb{R}^{n\times 1}$ is the vector containing all the velocity-dependent terms. To derive control law equations, it is desired to write the Coriolis vector term $\mathbf{C}_s(\boldsymbol{q}\ \ \boldsymbol{v})$ in the form of $\mathbf{C}(\boldsymbol{q}\ \ \boldsymbol{v})\boldsymbol{v}$ where $\mathbf{C}(\boldsymbol{q}\ \ \boldsymbol{v})\in\mathbb{R}^{n\times n}$ is the Coriolis matrix. However, there are many ways of factorizing the matrix $\mathbf{C}(\boldsymbol{q}\ \ \boldsymbol{v})$. To this end, the factorization of the $\mathbf{C}(\boldsymbol{q}\ \ \boldsymbol{v})$ matrix is chosen such that $\dot{\mathbf{M}}-2\mathbf{C}$ is a skew-symmetric matrix as suggested by Kim et al. (2003).[1] This specific parameterization affords simpler expressions in the derivation of control law methodologies.

The inertia mass matrix can be decomposed into the following submatrices:

$$\mathbf{M}=\begin{bmatrix}\mathbf{M}_{11} & \mathbf{M}_{12} & \mathbf{M}_{13}\\ \mathbf{M}_{21} & \mathbf{M}_{22} & \mathbf{M}_{23}\\ \mathbf{M}_{31} & \mathbf{M}_{32} & \mathbf{M}_{33}\end{bmatrix} \tag{11}$$

where $\mathbf{M}_{i,j}=\mathbf{M}_{j,i}\in\mathbb{R}^{3\times 3}$, $i=1,2$; $j=1,2$ are vehicle-related submatrices, $\mathbf{M}_{i,3}=\mathbf{M}_{3,i}^T\in\mathbb{R}^{3\times n-6}$, $i=1,2$ are reaction force-related submatrices due to the interaction between the manipulator and the vehicle, and $\mathbf{M}_{33}\in\mathbb{R}^{(n-6)\times(n-6)}$ is the manipulator inertia matrix. By parameterizing the Coriolis and centripetal matrix as:

$$\mathbf{C}=\begin{bmatrix}\mathbf{C}_{11} & \mathbf{C}_{12} & \mathbf{C}_{13}\\ \mathbf{C}_{21} & \mathbf{C}_{22} & \mathbf{C}_{23}\\ \mathbf{C}_{31} & \mathbf{C}_{32} & \mathbf{C}_{33}\end{bmatrix}=\begin{bmatrix}\boldsymbol{\Gamma}_1 & \boldsymbol{\Gamma}_2 & \boldsymbol{\Gamma}_3\end{bmatrix}, \tag{12}$$

$$\boldsymbol{\Gamma}_1=\begin{bmatrix}0\\ \frac{1}{2}\dot{\mathbf{M}}_{21}-S\left(\frac{\partial \bar{L}}{\partial \mathbf{v}_1}\right)\\ -\frac{1}{2}\left\lfloor\frac{\partial \mathbf{M}_{21}}{\partial \mathbf{q}}\mathbf{v}_2\right\rfloor\end{bmatrix},$$

$$\boldsymbol{\Gamma}_2=\begin{bmatrix}\frac{1}{2}\dot{\mathbf{M}}_{12}-S\left(\frac{\partial \bar{L}}{\partial \mathbf{v}_1}\right)\\ \frac{1}{2}\dot{\mathbf{M}}_{22}-S\left(\frac{\partial \bar{L}}{\partial \mathbf{v}_2}\right)\\ -\frac{1}{2}\left\lfloor\frac{\partial \mathbf{M}_{12}}{\partial \mathbf{q}}\mathbf{v}_1\right\rfloor-\frac{1}{2}\left\lfloor\frac{\partial \mathbf{M}_{22}}{\partial \mathbf{q}}\mathbf{v}_2\right\rfloor\end{bmatrix},$$

$$\boldsymbol{\Gamma}_3=\begin{bmatrix}\dot{\mathbf{M}}_{13}+\frac{1}{2}\left(\left\lfloor\frac{\partial \mathbf{M}_{21}}{\partial \mathbf{q}}\mathbf{v}_2\right\rfloor\right)^T\\ \dot{\mathbf{M}}_{23}+\frac{1}{2}\left(\left\lfloor\frac{\partial \mathbf{M}_{12}}{\partial \mathbf{q}}\mathbf{v}_1\right\rfloor\right)^T+\frac{1}{2}\left(\left\lfloor\frac{\partial \mathbf{M}_{22}}{\partial \mathbf{q}}\mathbf{v}_2\right\rfloor\right)^T\\ \frac{1}{2}\dot{\mathbf{M}}_{33}\end{bmatrix}$$

[1] There was an inaccuracy in the work of Kim et al. (2003) in which Eq. 12 is first proposed. The corrected version of the formula given here was provided by the authors of Kim et al. (2003) upon our request.

The term $\dot{\mathbf{M}} - 2\mathbf{C}$ can be guaranteed to be a skew symmetric matrix (or equivalently $\dot{\mathbf{M}} - (\mathbf{C} + \mathbf{C}^{\mathrm{T}}) = 0$). Note that the symbol, $\lfloor \frac{\partial \mathbf{A}}{\partial \mathbf{q}} \mathbf{a} \rfloor$ in Eq. 12, means:

$$\begin{bmatrix} \vdots \\ \mathbf{a}^T \dfrac{\partial \mathbf{A}}{\partial q_i} \\ \vdots \end{bmatrix}, \quad i = 1 \ldots n. \tag{13}$$

So far, the terms $\mathbf{M}$, and $\mathbf{C}$ are fully defined. In order to complete the closed-form expressions, it remains to define the dissipative drag matrix $\mathbf{D}(\boldsymbol{q}, \boldsymbol{v}) \in \mathbb{R}^{n \times n}$, buoyancy and gravitational force term $\mathbf{g} \in \mathbb{R}^{n \times 1}$. The definition of these forces can be found in Fossen (1994). Consequently, the dynamic equations of motion of ROVM systems in the body-fixed frame can be written as (Fossen, 1994):

$$\mathbf{M}\dot{\boldsymbol{v}} + \mathbf{C}(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} + \mathbf{D}(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} + \mathbf{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{14}$$

Equation 14 can also be represented in the inertial reference frame as (Fossen, 1994):

$$\boldsymbol{f} = \mathbf{M}_\eta(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{C}_\eta(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{D}_\eta(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_\eta(\boldsymbol{q}) + \boldsymbol{d} = J^{-\mathrm{T}}\boldsymbol{\tau} \tag{15}$$

where:

$$\mathbf{M}_\eta(\eta) = \mathbf{J}^{-\mathrm{T}}\mathbf{M}\mathbf{J}^{-1},$$
$$\mathbf{C}_\eta(\boldsymbol{q}, \boldsymbol{\eta}) = \mathbf{J}^{-\mathrm{T}}[\mathbf{C} - \mathbf{M}\mathbf{J}^{-1}\dot{\mathbf{J}}]\mathbf{J}^{-1},$$
$$\mathbf{D}_\eta(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathbf{J}^{-\mathrm{T}}\mathbf{D}\mathbf{J}^{-1}$$

and $\boldsymbol{g}_\eta(\boldsymbol{q}) = \mathbf{J}^{-\mathrm{T}}\boldsymbol{g}$, and $\boldsymbol{d} \in \mathbb{R}^{n \times 1}$ is the unknown disturbance forces acting on the system, such as tether disturbance forces, measurement noise, etc. Eq. 15 can be written in the more compact form of:

$$\boldsymbol{f} = \mathbf{M}_q\ddot{\boldsymbol{q}} + \boldsymbol{h}_q(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{16}$$

where:

$$\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathbf{C}_q(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{D}_q(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_q(\boldsymbol{q}) + \boldsymbol{d}.$$

Definition of Lumped Uncertainty

The ROVM dynamics are dominated by hydrodynamic loads, and it is difficult to accurately measure or estimate the hydrodynamic coefficients. As such, the system dynamics are not exactly known. The system dynamics given in Eq. 16 can then be written as the sum of estimated (known) dynamics $\hat{\boldsymbol{f}}$ and the unknown dynamics $\tilde{\boldsymbol{f}}$:

$$\boldsymbol{f} = \hat{\boldsymbol{f}} + \tilde{\boldsymbol{f}} \tag{17}$$

where the estimated dynamics vector is defined as:

$$\hat{\boldsymbol{f}} = \hat{\mathbf{M}}_q(\boldsymbol{q})\ddot{\boldsymbol{q}} + \hat{\boldsymbol{h}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{18}$$

with $\hat{\boldsymbol{h}}_q(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \hat{\mathbf{C}}_q(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\mathbf{D}}_q(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\mathbf{g}}_q(\boldsymbol{q})$, and the unknown dynamics vector is defined as:

$$\tilde{\boldsymbol{f}} = \tilde{\mathbf{M}}_q(\boldsymbol{q})\ddot{\boldsymbol{q}} + \tilde{\boldsymbol{h}}_q(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{d} \tag{19}$$

with:

$$\tilde{\boldsymbol{h}}_q(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \tilde{\mathbf{C}}_q(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \tilde{\mathbf{D}}_q(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \tilde{\boldsymbol{g}}_q(\boldsymbol{q}),$$

$\tilde{\mathbf{M}}_q = \mathbf{M}_q - \hat{\mathbf{M}}_q$, $\tilde{\mathbf{C}}_q = \mathbf{C}_q - \hat{\mathbf{C}}_q$, $\tilde{\mathbf{D}}_q = \mathbf{D}_q - \hat{\mathbf{D}}_q$, and $\tilde{\boldsymbol{g}}_q = \boldsymbol{g}_q - \hat{\mathbf{g}}_q$. The unknown dynamics vector is also called the lumped uncertainty vector and has been proposed for use in adaptive control by Lewis et al. (1995), Lin et al. (2002), Lin and Wai (2002) and Zeinali and Notash (2004a). In all of these prior works, it is assumed that the unknown lumped uncertainty vector is bounded.

ADAPTIVE SLIDING-MODE DESIGN

Conventional Adaptive Sliding-Mode Controller

The first step in the sliding-mode controller design procedure is to define the desired dynamics in the form of a vector of sliding manifolds $\boldsymbol{s} \in \mathbb{R}^{6 \times 1}$. The second step is to find a control law $\boldsymbol{\tau} \in \mathbb{R}^{6 \times 1}$ such that the system trajectories move towards the sliding manifold, and eventually hit the manifold. The control law $\boldsymbol{\tau}$ then keeps the state variables on the sliding manifold in the presence of system uncertainties and disturbances (Slotine and Li, 1991).

The vector of sliding manifolds can be defined as:

$$\boldsymbol{s} = \left(\frac{d}{dt} + \boldsymbol{\Lambda}\right)^2 \left(\int \boldsymbol{e}\, dt\right) = \dot{\boldsymbol{e}} + 2\boldsymbol{\Lambda}\boldsymbol{e} + \boldsymbol{\Lambda}^2 \int \boldsymbol{e}\, dt \tag{20}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{6 \times 6}$ is a constant, symmetric, positive definite and diagonal matrix that defines the bandwidth of the closed-loop system. The limiting factors on the choice of control bandwidth are given by Slotine and Li (1991). The tracking error between the measured state values and the desired state values is given by $\boldsymbol{e} = \boldsymbol{q} - \boldsymbol{q}_d$ with the subscript d denoting the desired position values produced by the redundancy resolver corresponding to the integral of Eq. 2. For notational simplicity, Eq. 20 can also be written as:

$$\boldsymbol{s} = \dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_r \tag{21}$$

where $\dot{\boldsymbol{q}}_r = \dot{\boldsymbol{q}}_d - 2\boldsymbol{\Lambda}\mathbf{e} - \boldsymbol{\Lambda}^2 \int \mathbf{e}\, dt$ and the subscript r refers to virtual reference trajectory (Slotine and Li, 1991).

The adaptive sliding-mode control law is in the form of:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{eq} + \boldsymbol{\tau}_{ad} \tag{22}$$

where $\boldsymbol{\tau}$ corresponds to a generalized force acting at the centre of the ROV mass, and $\boldsymbol{\tau}_{eq}$ and $\boldsymbol{\tau}_{ad}$ symbolize the equivalent control law and the adaptive control law, respectively.

The equivalent control law is continuous and reflects the model based component of the total control signal. This equivalent control signal alone can accomplish the desired dynamic when there are no uncertainties in the system dynamics. However, due to model uncertainties, discrepancy occurs between the desired dynamics and real dynamics. In order to offset this discrepancy, a continuous adaptive term $\boldsymbol{\tau}_{ad}$ is used in the current work. In the conventional sliding mode theory, a nonlinear discontinuous switching term is used for this purpose. However, this switching term is the source of high control activity, or chattering (Slotine and Li, 1991). The adaptive term does not cause the same complication since it acts on the system in a persistent continuous manner.

Equivalent Control Law

The model-based equivalent control law component of the sliding-mode control signal can be derived by assuming that the motion is constrained to the sliding manifold, i.e., $\dot{\boldsymbol{s}} = \mathbf{0}$. Taking the time derivative of Eq. 20 yields:

$$\dot{\boldsymbol{s}} = \ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_r \tag{23}$$

where $\ddot{\boldsymbol{q}}_r = \ddot{\boldsymbol{q}}_d - 2\boldsymbol{\Lambda}\dot{\boldsymbol{e}} - \boldsymbol{\Lambda}^2\boldsymbol{e}$. Multiplying both sides of Eq. 23 by the inertia matrix $\hat{\mathbf{M}}_\eta$ and substituting $\mathbf{M}_q\ddot{\boldsymbol{q}} = \mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau}_{eq} - (\mathbf{C}_q\dot{\boldsymbol{q}} + \mathbf{D}_q\dot{\boldsymbol{q}} + \boldsymbol{g}_q)$ from Eq. 15 into the resulting equation yields:

$$\hat{\mathbf{M}}_q\dot{\boldsymbol{s}} = \mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau}_{eq} - (\hat{\mathbf{M}}_q\dot{\boldsymbol{q}} + \hat{\mathbf{C}}_q\dot{\boldsymbol{q}} + \hat{\mathbf{D}}_q\dot{\boldsymbol{q}} + \hat{\mathbf{g}}_q) \tag{24}$$

Solving the resulting equation for $\boldsymbol{\tau}_{eq}$ with $\dot{\boldsymbol{s}} = \mathbf{0}$ yields the equivalent control:

$$\boldsymbol{\tau}_{eq} = \mathbf{J}^{\mathrm{T}}\hat{\boldsymbol{f}}_r(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_r) \tag{25}$$

where:

$$\hat{\boldsymbol{f}}_r(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_r) = \hat{\mathbf{M}}_q\ddot{\boldsymbol{q}}_r + \hat{\mathbf{C}}_q\dot{\boldsymbol{q}} + \hat{\mathbf{D}}_q\dot{\boldsymbol{q}} + \hat{\mathbf{g}}_q \tag{26}$$

Adaptive Control Law

The following continuous adaptive control law is used in place of the switching term for ROVM systems:

$$\boldsymbol{\tau}_{ad} = \mathbf{J}^{\mathrm{T}}(\tilde{\boldsymbol{f}}_{est} - (\mathbf{K} + \hat{\mathbf{C}}_q)\boldsymbol{s}) \tag{27}$$

where $\tilde{\boldsymbol{f}}_{est}$ is the estimation of the lumped uncertainty vector defined in Eq. 19, and $\mathbf{K} \in \mathbb{R}^{6\times6}$ is a diagonal positive definite constant matrix that is related to the convergence rate of the controller. The estimation of the lumped uncertainty vector is performed according to:

$$\dot{\tilde{\boldsymbol{f}}}_{est} = -\boldsymbol{\Gamma}\boldsymbol{s} \tag{28}$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{n\times n}$ is a positive definite diagonal constant design matrix that determines the rate of adaptation. This adaptive term relates the error metric $\boldsymbol{s}$ function to the dynamic uncertainties. Note that the assumption of $\tilde{\boldsymbol{f}}$ being bounded ensures that Eq. 28 is bounded.

Eq. 28 was first used by Lin et al. (2002) to estimate the lumped uncertainty vector in the control of linear inductance motors; it was also used in the 2007 work by the same authors. This adaptation law was also proposed by Zeinali and Notash (2004a) and was implemented to the trajectory control problem of parallel manipulators in Zeinali and Notash (2004b and 2007).

Note that unlike conventional adaptive control theory, no parameterization of a regressor matrix and unknown parameter vector needs to be found; this eliminates the need to simplify system dynamics, thus allowing the use of a more detailed dynamic model in the controller. In this regard, Eq. 28 blends the best traits of the sliding and adaptive control.

Finally, the total control input $\boldsymbol{\tau}$ based on Eq. 22 is defined as:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{eq} + \boldsymbol{\tau}_{ad} = \mathbf{J}^{\mathrm{T}}(\hat{\boldsymbol{f}}_r + \tilde{\boldsymbol{f}}_{est} - (\mathbf{K} + \hat{\mathbf{C}}_q)\boldsymbol{s}) \tag{29}$$

In order to verify that Eq. 29 provides asymptotic stability, the Lyapunov theorem is utilized. For this analysis, it is assumed that:

$$\boldsymbol{s}^T\mathbf{K}\boldsymbol{s} \geq |\,\dot{\tilde{\boldsymbol{f}}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w}\,| \quad \textit{only when} \quad \dot{\tilde{\boldsymbol{f}}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} < 0 \tag{30}$$

where $\boldsymbol{w} = \tilde{\boldsymbol{f}}_{est} - \tilde{\boldsymbol{f}}$ is the difference vector between the estimated and exact lumped uncertainty vectors. Note that when $\dot{\tilde{\boldsymbol{f}}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} \geq 0$, the system is stable in the Lyapunov sense, and thus no assumption is necessary. In the context of ROV trajectory control, a Lyapunov stability analysis of the controller given in Eqs. 28 and 29 under the assumption of Eq. 30 was presented by Soylu et al. (2008). The same work also presents the convergence of the estimated dynamics to the unknown dynamics (i.e. $\boldsymbol{w} \to 0$).

The condition given in Eq. 30 is distinct from many of the previous studies related to the adaptive sliding-mode controllers in that it does not assume that $\dot{\tilde{\boldsymbol{f}}}^{\mathrm{T}} = \mathbf{0}$. In the ROVM paradigm, there is a distinct possibility of sudden vehicle or arm motions being commanded by the human pilot. Such rapid changes in the commanded system state make the assumption of $\dot{\tilde{\boldsymbol{f}}}^{\mathrm{T}} = \mathbf{0}$ unrealistic.

In sliding-mode control theory, it is assumed that the dynamic parameter uncertainties are bounded, with their bounds available to the designers. However, these bounds are difficult to obtain due to the nonlinearity of the underwater environment. The controller given in Eq. 29 removes this constraint since the stability proof does not require the knowledge of upper bounds on the dynamic parameters of ROVM.

RESULTS AND DISCUSSION

The ROVM system considered in this work is the FALCON™ ROV equipped with a conceptual 4-DOF manipulator (Fig. 1). In the figure, the inertial frame is labelled E. As shown in Fig. 1, the body-fixed frame (X_v, Y_v, Z_v) is attached to the center of the ROV mass. The body-fixed frames are attached to a base joint of each successive link using the modified Denavit-Hartenberg convention (Craig, 2004) shown in Table 1.

Simulation case studies were performed to investigate the performance of the proposed method. In lieu of an actual pilot command, a series of waypoints was set for the end-effector, and a continuous set of end-effector values was generated using a 3rd-order polynomial function with zero initial and final velocities. The system starts from the initial configuration of $\mathbf{q} = [0\ 0\ 0;\ 0\ 0\ 0;\ 200\ \ -40\ \ 115\ \ 45]^T$ m; rad; deg; that corresponds to the end-effector position of $\mathbf{x} = [0.83\ \ 0.055\ \ 0.75]^T$ m. The final end-effector location of $\mathbf{x} = [1.62\ \ 0.82\ \ 0.04]^T$ m is attained in 15 s.

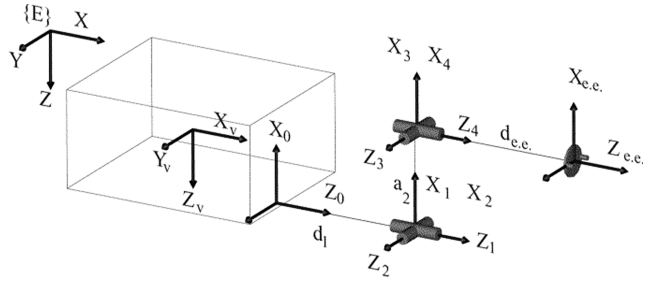


Fig. 1 FALCON ROVM with zero-displacement configuration

$i-1$	$\alpha-1$	a_{i-1}	d_i	θ_j	i
0	0	0	$d_1 = 0.15$m	q_1	1
1	$-\pi/2$	0	0	q_2	2
2	0	$a_2 = 0.22$m	0	q_3	3
3	$\pi/2$	0	$d_{ee} = 0.15$m	q_4	ee

Table 1 D&H parameters of manipulator

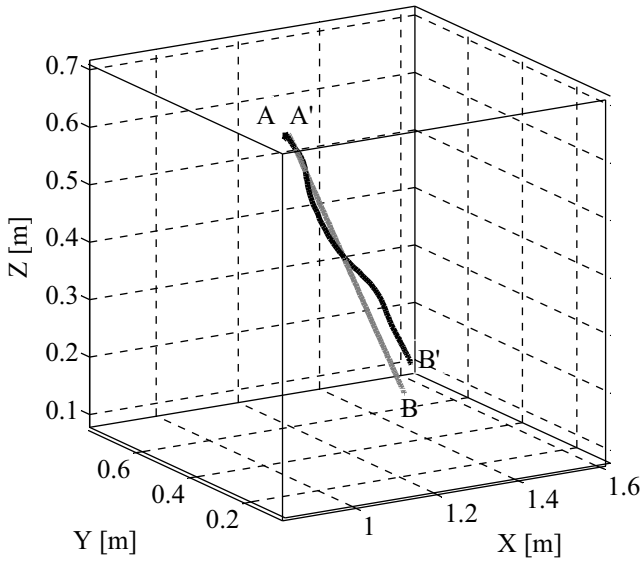


Fig. 2 End-effector commanded trajectory

The ROV is rectangular in shape with length, width and height of 1, 0.6 and 0.5 m, respectively. The mass of the vehicle is 50 kg. Each manipulator link is modeled as a cylinder; links 1, 2 and 3 are 0.15, 0.22 and 0.15 m long, respectively. The link masses are 2, 4, and 3 kg, respectively. The system is assumed to be neutrally buoyant. With regards to the system's thruster layout, the system is equipped with 5 electro-mechanical thrusters; 2 pairs for lateral motions, and one is for the ROV's heave motion. In the current simulation study, the thruster dynamic response is assumed to be much faster than that of the ROV itself, and thus

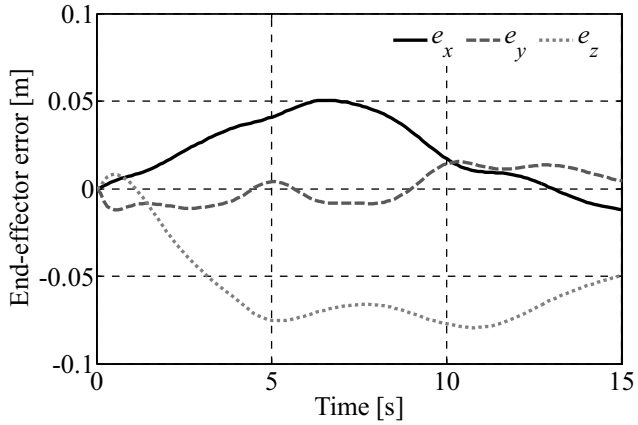


Fig. 3 End-effector error

is neglected. Regarding the environment, the surrounding fluid is assumed to be motionless. During the simulation, the ROV's roll and pitch motion are not controlled since the on-board thrusters allow only lateral and vertical motions. To reflect uncertainties that are present in the ROVM system dynamics, the exact added mass and drag coefficients in the controller's dynamic model parameters were changed by 40% to 60% randomly. The sampling frequency for the simulation was set to 10 Hz.

The primary task is to make the end-effector follow the pilot's command while fulfilling as much as possible the secondary tasks (joint limits, singularity, camera view and drag minimization via the yaw angle alignment with the ocean current, and minimal ROV motion tasks). The manipulator has the following joint limits with respect to the zero displacement ($q_i = 0$) configuration

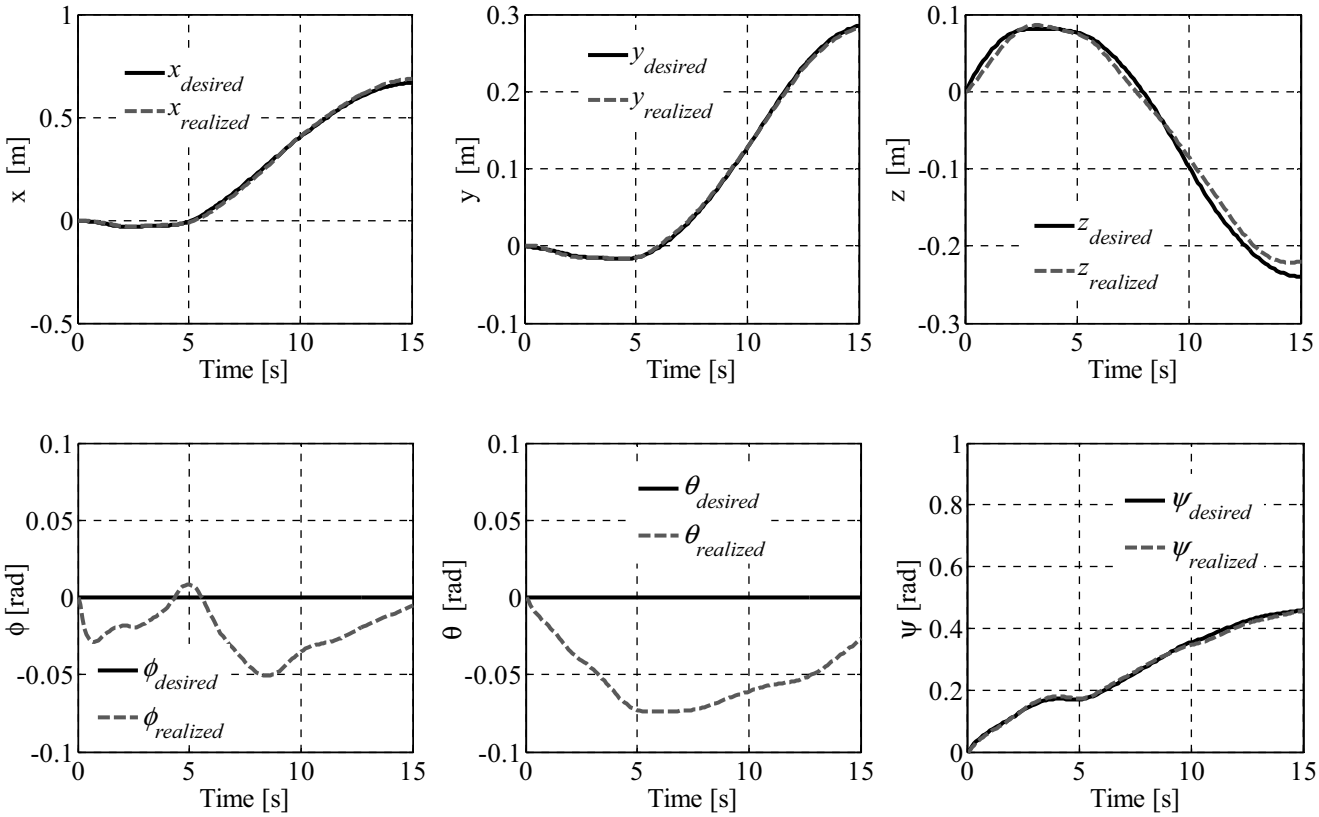


Fig. 4 Vehicle's desired and realized state values

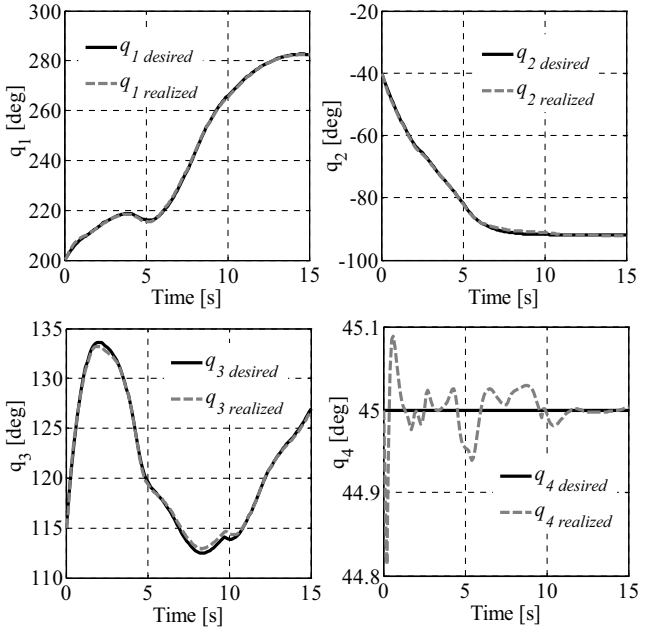


Fig. 5 Manipulator's desired and realized state values

shown in Fig. 1; $180 \leq q_1 \leq 290$, $-95 \leq q_2 \leq -5$, $90 \leq q_3 \leq 135$, $q_4 \leq 180$ degrees.

Fig. 2 demonstrates the desired and the achieved trajectories that are indicated by the grey and black lines, respectively. In Fig. 2, the points of A and B and A′ and B′, represent the initial and final end-effector locations for the desired and achieved trajectories, respectively. Fig. 3 reveals the simulation results for end-effector errors; as can be seen from the figure, these are

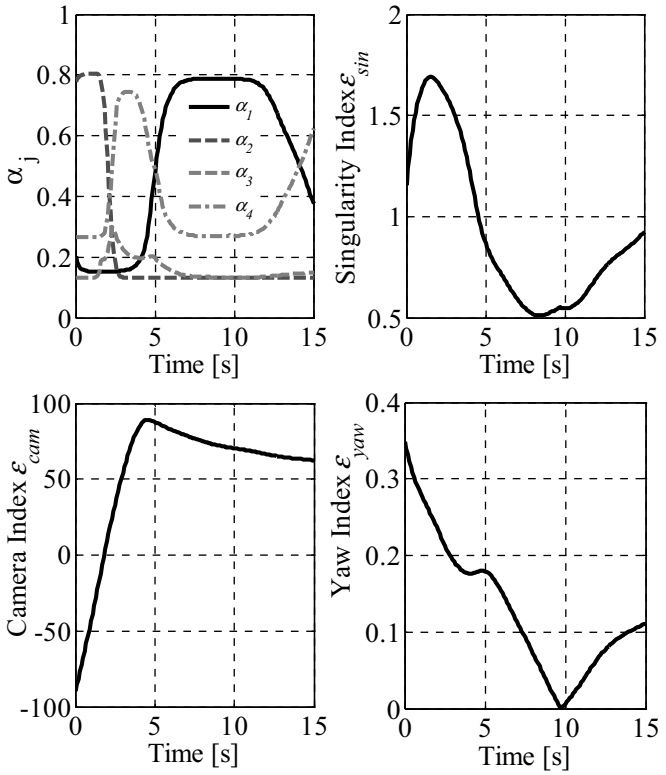


Fig. 6 Time history of fuzzy outputs α_j and corresponding index values

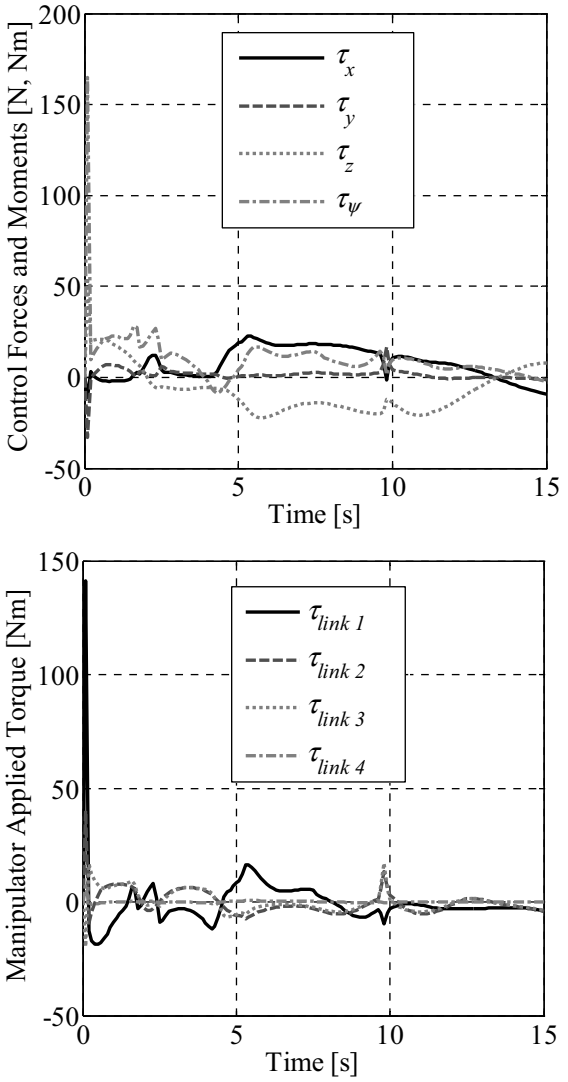


Fig. 7 Top: control forces and moments at ROV center of mass; bottom: manipulator control torques

within a small range. The desired and realized position history for the vehicle and the manipulator are illustrated in Figs. 4 and 5, respectively.

As can be seen from the desired state values, the redundancy resolution scheme produced ROVM positions that are realizable, as they show a smooth pattern with no jerk. As well, the desired joint positions were kept within their physical joint limits. It can also be seen in Fig. 4 that the discrepancy between the desired and realized values is small, demonstrating the ability of the proposed controller not only to realize the redundancy revolver's demand but also to produce kinematically and dynamically realizable ROV-manipulator state values. Note that by virtue of the existing thruster layout on the FALCON vehicle, the roll and pitch motion of the ROV can not be controlled, hence the error values are high for these ROV DOF.

Fig. 6 reports the outputs of the fuzzy logic, α_j. In this, α_j, $j = 1, 2, 3$, and 4 represents the decision variables for the singularity avoidance, camera view, yaw alignment and minimal vehicle motion objectives, respectively. This order also indicates the hierarchy from highest to lowest priority to reduce the number of fuzzy rules that are presented by Soylu et al. (2007).

Acceptable values for the index values were determined to be $\varepsilon_{sin} \geq 3$, $\varepsilon_{cam} \geq 0$, and $\varepsilon_{yaw} \geq 0.5$ (Soylu et al., 2007). As Fig. 6

shows, the redundancy resolver is successful in keeping these index values within their allowable range. Note that minimal ROV motion is realized only when the other objectives are not critical. This arrangement allows the ROV DOF to be used to improve higher-priority objectives.

Fig. 7 shows the demanded control forces and moments; it can be seen that the proposed controller yields a smooth control output without high-frequency controller demand. It is thus chattering-free. As mentioned above, the adaptive sliding-mode control removes the chattering action by means of the on-line adaptation law.

CONCLUSIONS

In this work, ROV and manipulator motions were coordinated using a consolidated dynamic model and control method to achieve a desired end-effector motion as defined by a single pilot command. The controller reference signals were provided by the redundancy resolution technique. The redundancy resolution uses the GPM method merged with a fuzzy logic-based weighting scheme to introduce several secondary objectives. It has been shown that this method provides an efficient means to translate the pilot's command into optimal ROVM motion. In the dynamic modeling and robust control stage, the quasi-Lagrange method was used to derive the equations of motion for the ROVM system, and the dynamic model was applied in a sliding-mode controller. To eliminate the chattering problem associated with conventional sliding-mode control, a continuous adaptive term was used. This adaptive term continuously compensates for the unknown ROVM system dynamics caused by poorly approximated nonlinear hydrodynamics. Numerical case studies have been developed based on the proposed scheme. Results have shown that complex spatial maneuvers can be realized with the proposed scheme.

ACKNOWLEDGEMENTS

The authors would like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for providing financial support for this work. Also, the authors wish to extend their thanks to Suboceanic Sciences Canada Ltd. in Duncan, British Columbia, Canada, for their participation in and support of the research.

REFERENCES

Antonelli, G, and Chiaverini, S (1998). "Adaptive Tracking Control of Underwater Vehicle-Manipulator Systems," *Proc Conf Control Applications*, Trieste, Italy, pp 1089–1093.

Antonelli, G (2003). *Underwater Robots: Motion and Force Control of Vehicle-Manipulator Systems.* Springer-Verlag, Berlin, Germany.

Antonelli, G, and Chiaverini, S (2003). "Fuzzy Redundancy Resolution and Motion Coordination for Underwater Vehicle-Manipulator Systems," *IEEE Trans Fuzzy Systems*, Vol 11, No 1, pp 109–120.

Canudas de Wit, C, Olguin Diaz, E, and Perrier, M (2000). "Nonlinear Control of an Underwater Vehicle/Manipulator System with Composite Dynamics," *IEEE Trans Control System Tech,* Vol 8, pp 948–960.

Chiaverini, S (1997). "Singularity-robust Task-Priority Redundancy Resolution for Real-time Kinematic Control of Robot Manipulators," *IEEE Trans Robotics Automation*, Vol 13, pp 398–410.

Craig, JJ (2004). *Introduction to Robotics: Mechanics and Control*, 3rd Ed, Prentice-Hall, NJ, USA.

Driankov, D, Hellendoorn, H, and Reinfrank M (1996). *An Introduction to Fuzzy Control*, 2nd Ed, Springer-Verlag, New York, USA.

Elmali, H, and Olgac, N (1992). "Theory and Implementation of Sliding-mode Control with Perturbation Estimation," *Proc IEEE Int Conf Robotics and Automation*, Vol 3, Nice, France, pp 2114–2119.

Euler, JA, Dubey, RV, and Babcock, SM (1989). "Self-motion Determination Based on Joint Velocity Bounds for Redundant Robots," *J Robotic System*, Vol 4, pp 417–425.

Fossen, T (1994). *Guidance and Control of Ocean Vehicles*, John Wiley & Sons, New York, USA.

Ioi, K, and Itoh, K (1990). "Modeling and Simulation of an Underwater Manipulator," *Advanced Robotics,* Vol 4, No 4, pp 303–317.

Kim, J, Chung, WK, and Yuh, J (2003). "Dynamic Analysis and Two-time Scale Control for Underwater Vehicle-Manipulator Systems," *Proc 2003 IEEE/RSJ Int Conf Intelligent Robots and Systems,* Vol 1, pp 577–582.

Lewis, FL, Liu, K, and Yesildirek, A (1995). "Neural Net Controller with Guaranteed Tracking Performance," *IEEE Trans Neural Networks*, Vol 6, No 3, pp 703–715.

Lin, FJ, Shen, PH, and Hsu, SP (2002). "Adaptive Backstepping Sliding-mode Control for Linear Induction Motor Drive," *IEE Proc Electric Power Applications*, Vol 149, No 3, pp 184–194.

Lin, FJ, and Wai, RJ (2002). "Robust Control Using Neural Network Uncertainty Observer for Linear Induction Motor Servo Drive," *IEEE Trans on Power Electronics,* Vol 17, No 2, pp 241–254.

Lin, FJ, Chang, CK, and Huang, PK (2007). "FPGA-Based Adaptive Backstepping Sliding-Mode Control for Linear Induction Motor Drive," *IEEE Trans on Power Electronics*, Vol 22, No 4, pp. 1222–1231.

McMillan, S, Orin, DE, and McGhee, RB (1995). "Efficient Dynamic Simulation of an Underwater Vehicle with a Robotic Manipulator," *IEEE Trans Systems, Man, and Cybernetics*, Vol 25, pp 1194–1206.

Meirovitch, L (1990). *Dynamics and Control of Structures*, Wiley Interscience, New York, USA.

Meirovitch, L, and Stemple, T (1995). "Hybrid Equation of Motion for Flexible Multibody Systems Using Quasicoordinates," *J of Guidance, Control and Dynamics*, Vol 18, No 4, pp 678–688.

Sagatun, S, and Fossen T (1991). "Lagrangian Formulation of Underwater Vehicles' Dynamics," *Proc IEEE Int Conf on Systems, Man and Cybernetics,* Charlottesville, Virginia, USA, pp 1029–1034.

Sarkar, N, and Podder, TK (2001). "Coordinated Motion Planning and Control of Underwater Vehicle-Manipulator Systems Subject to Drag Optimization," *IEEE J Oceanic Eng,* Vol 26, No 2, pp 228–239.

Slotine, JJ, and Shastry, S (1983). "Tracking Control of Nonlinear System Using Sliding Surfaces," *Int J Control*, Vol 38, No 2, pp 465–492.

Slotine, JJ, and Li, W (1991). *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, NJ, USA.

Soylu, S, Buckham, BJ, and Podhorodeski, RP (2008). "A Chattering-free Sliding-mode Controller for Underwater Vehicles with Fault-tolerant Infinity-norm Thrust Allocation," *Ocean Eng*, Vol 35, No 16, pp 1647–1659.

Soylu, S, Buckham, BJ, and Podhorodeski, RP (2007). "A Fault-tolerant Fuzzy Logic-based Redundancy Resolution Method for Mobile Underwater Manipulators," *Proc Oceans 2007 MTS/IEEE*, Vancouver, pp 1–9.

Tarn, TJ, Shoults, GA, and Yang, SP (1996). "A Dynamic Model of an Underwater Vehicle with a Manipulator Using Kane's Method," *J Autonomous Robots*, Vol 3, No 2, pp 195–209.

Watkins, DS (2002). *Fundamentals of Matrix Computations,* 2nd Ed, John Wiley & Sons, New York, USA.

Xu, B, Pandian, SR, and Petry, F (2005). "A Sliding-mode Fuzzy Controller for Underwater Vehicle-Manipulator Systems," *Proc 2005 Annual Mtg North Amer Fuzzy Info Processing Soc (NAFIPS' 2005)*, Ann Arbor, Michigan, USA, pp 181–186.

Yoerger, D, and Slotine, JJ (1985). "Robust Trajectory Control of Underwater Vehicles," *IEEE J Oceanic Eng,* Vol OE-10, pp 462–470.

Zeinali, M, and Notash, L (2004a). "New Approach to Design: A Nonlinear Robust Adaptive Controller for Manipulators," *Proc CSME Forum*, London, Canada, 10 pp.

Zeinali, M, and Notash, L (2004b). "Robust Adaptive Neural Fuzzy Controller with Model Uncertainty Estimator for Manipulators," *CSME Trans*, Vol 28, No 2A, pp 197–219.

Zeinali, M and Notash, L (2006). "Chattering-Free Sliding Mode Control Design Using Fuzzy Modelling and Implementation on Robot Manipulators," Proc of the CSME Forum, Calgary, Canada, 13 pp.

Zeinali, M, and Notash, L (2007). "Fuzzy Logic-based Adaptive Robust Control for Parallel Manipulators," *Proc 12th World Cong Mechanism and Machine Sci,* Besançon, France, 8 pp.

In Memoriam

Cupertino, California (March 1, 2009). It is with a deep sense of loss that we announce the passing on January 30, 2009 of our dear colleague **Dr. Narasimha Rao V. Bangaru** at age of 60. Rao received his PhD degree in Materials Science from University of California at Berkeley in 1978. Subsequently, he joined General Motors Research Laboratories in 1977 and, in 1982, joined the Corporate Research Laboratory of ExxonMobil Research and Engineering Company where he was recently promoted to the position of a Distinguished Research Associate. Dr. Bangaru was an internationally renowned expert in physical metallurgy of steels and led major technology programs, notably the development of high strength pipeline steels for high pressure gas transmission.

Rao was a member of the editorial board of the *International Journal of Offshore and Polar Engineering* (IJOPE) and a charter member of ISOPE. Rao authored numerous reports, patents, and journal publications.

Editors, International Journal of Offshore and Polar Engineering (IJOPE)

Appendix H. New Operation Scheme for Underwater Mobile Manipulator

(Manuscript in Preparation)

NEW OPERATION SCHEME FOR UNDERWATER MOBILE MANIPULATORS

Serdar Soylu*, Flavio Firmani, Bradley J. Buckham, and Ron P. Podhorodeski
Department of Mechanical Engineering,
University of Victoria, P.O. Box 3055, Victoria, B.C., Canada, V8W 3P6
{serdar, ffirmani, bbuckham, podhoro}@me.uvic.ca
*Corresponding author: Fax: (250) 721-6035

Abstract

This work presents a novel operation scheme for the coordinated control of small remotely operated vehicle–manipulator systems (**ROVMs**). In small commercial ROVMs, the motion of the vehicle and the manipulator are usually commanded separately. In the proposed scheme, a human pilot commands only the manipulator's end-effector motion using a six-degree-of-freedom (**6-DOF**) joystick. The generated reference motion is then converted into a set of desired vehicle and manipulator joint motions by means of a fault-tolerant redundancy resolution scheme that utilizes the ROVMs' redundant DOF to accomplish secondary objectives. The redundancy resolver generates reference state values that are fed into a novel sliding-mode based adaptive controller. The controller contains two layers of adaptivity: the first layer adjusts PID gains, and the second layer estimates the upper bound on a lumped uncertainty vector. The proposed adaptation laws are robust to the parameter drift problem that would otherwise lead to instability in the presence of the poor system input signals that are expected in the submerged mobile-robot paradigm. In addition, a new human-machine interface (**HMI**) is proposed that facilitates effective use of the proposed scheme. The new HMI provides a set of ROVM operation modes to the human pilot for an effective use of the proposed

redundancy resolution algorithm to perform complex subsea tasks involving 6-DOF tasks. Hardware-in-the-loop simulation studies illustrate that detailed subsea tasks that require full 6-DOF motion of the end effector can be completed with a small, low-cost ROVM with an otherwise underactuated manipulator using the proposed operation scheme.

1 INTRODUCTION

Whether motivated by science or harvest of natural resources, the extent and complexity of human interaction with the submerged world is increasing. To support a growing population of investigators and industries working in littoral regions, there is a need for small, low cost and easily deployed underwater remotely operated vehicle-manipulator (**ROVM**) systems that can efficiently interact with seabed ecosystems and submerged infrastructure. However, there are roadblocks associated with the move to small commercial ROVM systems. In standard ROVM operations, the motions of the remotely operated vehicle (**ROV**) and the manipulator are guided independently by human pilots with the ROV typically held stationary to ease the burden on the human operators. This practice eliminates the redundancy inherent in the complete ROVM system and, specific to smaller ROVM systems with simple (underactuated) manipulators, compresses the end-effector workspace beyond useful limits. In order to drive the dexterity of small ROVM systems above the threshold of utility, one must complement the manipulator degrees of freedom (**DOF**) with coordinated commands to the active ROV DOF. This unification of the ROVMs collective DOF transforms the small ROVM into a mobile redundant manipulator. Such unified ROVM operation requires a human-machine interface through which the human pilot provides a single 6-

DOF end effector input. In this work, a parallel-architectured 6-DOF joystick is used to command the ROVM's end-effector motion in real-time. The parallel joystick architecture uses a redundant sensing methodology to provide very accurate interpretation of the pilot command and also affords unencumbered input motions during operation. In addition, the joystick's parallel-architecture is very compact compared to a standard serial master arm manipulator interface.

In this work, a comprehensive ROVM operation scheme is presented that merges the joystick and previously reported redundancy resolution and thrust allocation techniques with a new robust non-linear control strategy. The scheme is demonstrated in hardware-in-the-loop simulations of a small ROVM. The ROVM considered is a small Saab-Seaeye Falcon™ ROV with a modified Hydrolek™ 43000 manipulator that is assumed capable of simultaneous and variable rate joint motions. The consideration of actual pilot behaviour via the new joystick is distinct from other simulation based studies, and allows evaluation of the redundancy resolver's and control system's sensitivity and robustness, respectively, to uncertainty in the commanded end effector motion.

As discussed in [1], the ROVM's intrinsic redundancy can be automatically exploited to achieve additional, or secondary, objectives that should improve the efficacy of the pilot-ROVM interface in the coming moments. If properly chosen, the secondary objective(s) create internal motions that reflect the behaviour of a very skilled human operator. A handful of works have considered redundancy resolution for ROVMs, with each work separated by the algorithm employed and the choice of secondary objective applied. Antonelli and Chiaverini used a task priority approach with a secondary

objective of minimizing the ROV motion to mimic standard ROVM pilot behaviour [2]. In [3], Sarkar and Podder use a gradient projection method (**GPM**) to attempt to minimize the system drag while completing a requested end-effector motion. In [4], Podder and Sarkar use the same GPM to calculate dynamically admissible ROV and manipulator motions for a preordained end-effector trajectory. Yet another choice of secondary task, minimizing the ROV restoring moments, was implemented by Han and Chung [5]. However, given the wide range of circumstances encountered in ROVM deployments, it is likely that the optimal secondary task, one that does ease the pilot's burden, is a dynamic entity – it will change throughout a single maneuver. Antonelli and Chiaverini showed how a fuzzy hierarchy of multiple secondary objectives can be accommodated in the secondary task priority algorithm [6].

In this work, a fault tolerant redundancy resolution algorithm originally presented by Soylu *et al*. [1] is used to distribute a desired end-effector trajectory over the vehicle and manipulator joint rates. This distribution is made by coupling the **GPM** [7] with a Mamdani type fuzzy interference machine [8]. As opposed to the task priority approach, this GPM method does not require specification of the secondary tasks in terms of real world trajectories nor a Jacobian that translates these trajectories into the joint space. Rather, scalar objective functions, and their gradients, are defined in terms of the joint rates a priori, and then referred to on-the-fly to determine the current direction in the joint space that should improve adherence to the desired objective. The method is believed to be well suited to ROVM operations for which the pilot intent is always changing. In many ROVM redundancy resolution works, the end effector task is preordained: the

desired end effector trajectory is known with time (see [6] and [4] for example). The redundancy resolution functions as a low level artificial pilot that oversees a robust controller. This artificial pilot monitors the desired end-effector motion from the spatial joystick, prioritizes the secondary objectives based on the current pose and determines the subsequent pose of the ROVM system. The secondary objectives include, but are not limited to: avoiding manipulator joint limits, singularity and high joint velocity avoidance, keeping the end-effector inside the camera view, aligning the vehicle with the ocean current (drag minimization), and minimization of the ROV motion.

The coordinated ROV and manipulator states requested by the redundancy resolver should be realized by a consolidated controller for the ROVM. The few cases where the manipulator and ROV dynamics are unified include: Antonelli and Chiaverini's application of an adaptive unified controller for an ROVM [9], [10] a singular perturbation method demonstrated by Canuscas *et al*. [11] a computed-torque method implemented by Sarkar and Podder [3], a model based sliding-mode controller presented by Antonelli [12], and a sliding-mode strategy with a fuzzy logic based tuning of the control gains presented by Xu *et al*. [13]. It is the authors' opinions that the sliding-mode strategy is well suited to the ROVM paradigm: the technique is very robust with respects to parameter uncertainty in the system dynamics model as well as unknown external disturbances. For ROVMs, both of these difficulties are assured – given changes in the payloads and configuration of an ROV sled, the dynamic parameters are often in error and unknown water currents and tether disturbances produce unpredicted forces on the ROVM. However, to avoid the conventional switching term from demanding high

frequency control actions, a chattering free adaptive-sliding-mode controller is preferred. Such a technique was implemented by the authors in [14] for automatic ROV positioning and later applied to ROVM end-effector control in [15]. In [15], the continuous adaptive control law is based on estimation of a lumped uncertainty vector (the difference between the known and the unknown dynamics) that was presented in [16]. However, the stability criterion of the adaptive sliding-mode controller presented in that work proves difficult to satisfy in the stop-and-go type maneuvering that is typical of ROVM operations: when the objective changes suddenly, the stability criterion is very difficult to satisfy. In addition, typical ROVM operation (that is to say "stop-and-go") introduces moments where the control input is not persistently exciting the vehicle dynamics (an ROV station-keeping in still water for instance). In the authors' experiences, this has been found to lead to significant and rapid, parameter drift in the adaptive control.

In the current work, the sliding-mode controller presented in [15] for ROVMs is enhanced and applied to pursue the redundancy resolver's demands. The control law is three layered and includes: an equivalent controller, a new adaptive controller, and a new adaptive PID control signal. The equivalent control signal is a model based component that reflects the knowledge of the system dynamics available to the controller designer. Since a model-based controller is implemented, a closed form dynamic equation of the ROVM is required, which is determined using the quasi-Lagrange approach [17], [18]. The quasi-Lagrange approach differs from the standard Lagrange formulation in that it generates the equations of motion in the body-fixed frame. This is convenient for ROVM systems since the motions, forces and torques measured by on-board sensors are in terms

of a body-fixed frame. The adaptive control signal constantly estimates the upper bound on the lumped uncertainty vector – an incremental change from the work of [14] and [15], which aims to improve the controller stability in the stop and go circumstances expected with the inclusion of a human operator in the control loop (via the new spatial joystick interface). Finally, the adaptive PID control layer is used to tune the overall system performance by eliminating overshoot and any steady state errors that penetrate the equivalent and adaptive layers. The two adaptation laws herein contain leakage term based on a variant of the $\sigma-$modification technique [19] to avoid the "parameter drift" problem. The generalized force calculated through this three layered strategy is transformed into a set of thruster commands using the infinity norm thrust allocation method described in [14].

Execution of any of the strategies relies on an improved human-machine interface (**HMI**). To this end, a new HMI is proposed that integrates RSI 6-DOF Joystick, an OpenGL-based 3-D graphical display, and a new graphical user interface (**GUI**) to facilitate complex ROVM missions. While the Joystick generates reference end-effector motion, the 3-D graphical display emulates the camera feedback that would exist in actual ROVM operation. As for the GUI, it consists of a set of control knobs and monitoring interfaces to facilitate the interaction between the human pilot and the ROVM operation-modes of the HMI. Furthermore, the new GUI also allows the pilot to operate the ROVM system using the conventional one-joint-at-a-time with the optional station-keeping mode.

To the authors' best knowledge, an integrated ROVM operation strategy, a multi-objective fault tolerant redundancy resolver that processes those inputs, a non-linear controller, a fault-tolerant thrust allocator accounting for real limits in ROV manoeuvrability and an advanced HMI has not been presented before. Furthermore, none of these works include a hardware to generate end-effector desired velocities reflecting the tendencies of a human pilot – a factor that is very important in the evaluation of any ROVM operating strategy for a broader sense.

The paper is organized as follows: Section 2 Trajectory Generation presents the joystick structure and the method for processing the human inputs to the ROVM. Section 3 ROVM Kinematics reviews the methods for describing the collective ROVM motions (ROV and manipulator) as well as a brief overview of the fuzzy multi-objective GPM that forms the redundancy resolver. Section 4 Control presents the development of the three-layered control strategy with emphasis on the new adaptive terms and the methods employed to prevent parameter drift. In Sections 5, the overall system software architecture and the components of the new HMI are presented with emphasis on the GUI component. In Section 6, the results of the simulation studies are provided. A comparison study shows the performance of the proposed controller compared to a fundamental computed-torque method for a synthesized ROVM maneuver in which the end-effector follows a continuous and smooth trajectory from point to point. To demonstrate the efficacy of the new HMI consisting of the new operation scheme and controller, hardware-in-the-loop simulations are completed using the new parallel

architecture joystick. The results illustrate that detailed subsea tasks can be completed with a small, low-cost ROVM system only using the proposed operation scheme.

2 TRAJECTORY GENERATION

In the current implementation, the human pilot sets only the desired ROVM end-effector position and orientation using the parallel-architectured joystick. The displacement of the handle's joystick is determined by sensing the displacement of the joystick's joints; then through a computer interface, the displacement of the handle is transferred to the actuators of the remote system. The joystick employed in this project is the RSI joystick [20] shown in Figure 1.

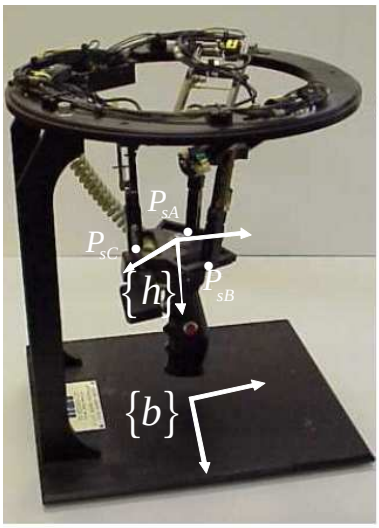


Figure 1: RSI 6-DOF joystick.

A high level block diagram of the proposed ROVM operation scheme is presented in Figure 2. A human pilot controls the motion of the end-effector by manipulating the 6-DOF parallel joystick.

The pilot uses a 3D computer graphics rendition of the ROVM system. The pose of the joystick contains six quantities: three for location and three for orientation (Euler angles). Depending on the operation mode, these quantities are transformed into either a desired displacement vector or a desired velocity vector of the end-effector. The desired end-effector motion is then fed into the redundancy resolution block that determines desired state values based on the secondary objectives using redundant DOFs. The generated desired state values correspond to the optimal posture for a given set of secondary objectives. The desired state values are then input into the controller. The controller accepts displacement and velocity feedback and outputs a generalized force control vector that includes manipulator joint torques for the manipulator and generalized force and moment vector for the vehicle. Finally, the control force/moment vectors are distributed over the on-board thrusters in such a way that the redundant thrust configuration is utilized.

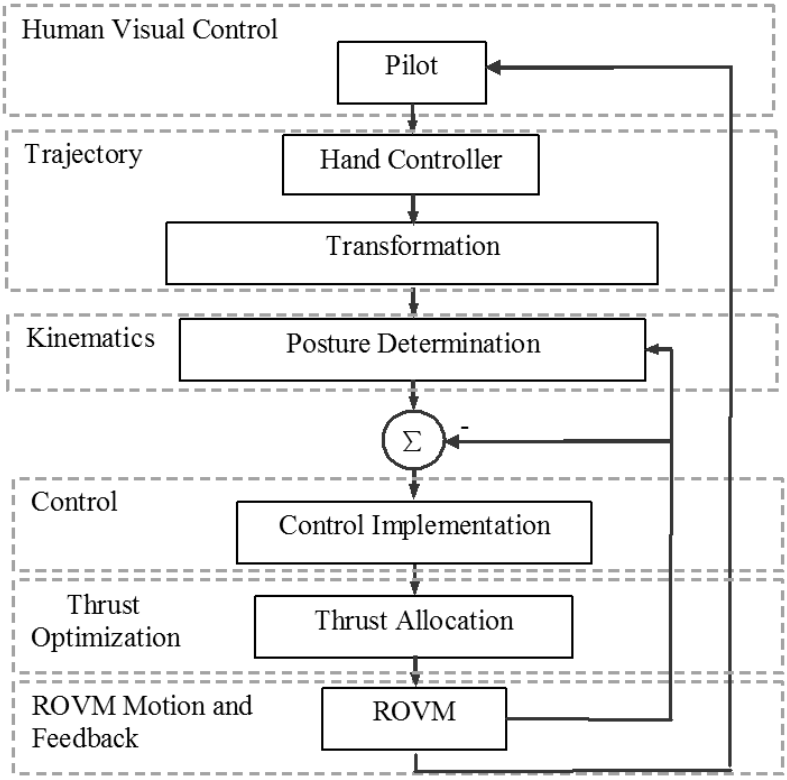


Figure 2: High level block diagram of the ROVM implementation.

The RSI joystick has a parallel architecture that is composed of two platforms (the base and the handle), which are connected with three symmetric branches. Each branch contains five links and six DOF. Platforms and links are sequentially connected with joints, six in total as shown in Figure 1. The minimum number of transducers that the RSI hand controller requires is six. Nevertheless, there would be up to 16 solutions to the forward displacement problem. To eliminate this uncertainty, Notash and Podhorodeski [21] analyzed the forward displacement problem of the RSI joystick and concluded that with three additional sensors, there is a unique solution to the forward displacement problem. As a result, the joystick is instrumented with nine transducers, namely rotary Midori™ CP-2FB potentiometers. These potentiometers are symmetrically arranged and

their function is to measure the angular displacement of the first three joints of each branch. To digitalize the output voltages of the potentiometers, the DAQ card is employed.

2.1 Kinematic Analysis of the Joystick

As can be seem from Figure 1, the arrangement of the spherical group of joints on the handle form an equilateral triangle, and hence the position of the centre of the handle, ${}^{b}\boldsymbol{p}$, with respect to frame {b} can be taken as the average of the ${}^{b}\boldsymbol{p}_{si}$ where $i = A, B, C$ denotes one of the three spherical joints as shown in Figure 1.

$$ {}^{b}\boldsymbol{p} = ({}^{b}\boldsymbol{p}_{sA} + {}^{b}\boldsymbol{p}_{sB} + {}^{b}\boldsymbol{p}_{sC})/3 \tag{1} $$

where ${}^{b}\boldsymbol{p}_{si}, i = A, B, C$ can be obtained from the forward kinematics solution for each branch [21]. The orientation of the handle with respect to frame {b} requires the identification of the unit vectors describing the axes of the frame attached to the handle {h} with respect to frame {b}, i.e., ${}^{b}_{h}\mathbf{R} = [{}^{b}\hat{x} \quad {}^{b}\hat{y} \quad {}^{b}\hat{z}]$. The ${}^{b}\hat{x}$ and ${}^{b}\hat{y}$ axes of frame {h} lie on the plane that contains points ${}^{b}\boldsymbol{p}_{sA}$, ${}^{b}\boldsymbol{p}_{sB}$, and ${}^{b}\boldsymbol{p}_{sC}$.

Thus, ${}^{b}\hat{x} \in \Re^{3\times1}$ is determined as follows:

$$ {}^{b}\hat{\boldsymbol{x}} = \left({}^{b}\boldsymbol{p}_{sC} - {}^{b}\boldsymbol{p}_{sB}\right) \Big/ \left\| {}^{b}\boldsymbol{p}_{sC} - {}^{b}\boldsymbol{p}_{sB} \right\| \tag{2} $$

The ${}^{b}\hat{z} \in \Re^{3\times1}$ axis, which is normal to the handle plane, can be determined taking the cross-product, i.e.

$$ {}^{b}\hat{\boldsymbol{z}} = \left({}^{b}\boldsymbol{p}_{sA} - {}^{b}\boldsymbol{p}_{sC}\right) \times \left({}^{b}\boldsymbol{p}_{sB} - {}^{b}\boldsymbol{p}_{sC}\right) \Big/ \left\| \left({}^{b}\boldsymbol{p}_{sA} - {}^{b}\boldsymbol{p}_{sC}\right) \times \left({}^{b}\boldsymbol{p}_{sB} - {}^{b}\boldsymbol{p}_{sC}\right) \right\| \tag{3} $$

Finally, ${}^{b}\hat{y} \in \Re^{3\times 1}$, which is normal to ${}^{b}\hat{x}$ and ${}^{b}\hat{z}$, is determined as ${}^{b}\hat{y} = {}^{b}\hat{z} \times {}^{b}\hat{x}$. Finally, the rotation matrix that describes frame {h} with respect to frame {b} is assembled, i.e. ${}^{b}_{h}\mathbf{R} = [{}^{b}\hat{x} \quad {}^{b}\hat{y} \quad {}^{b}\hat{z}]$.

2.2 Calibration

The voltage gathered from the transducers has to be converted into angular displacement through the following linear transfer function:

$$\theta_{j_i} = \gamma_{j_i}\left(\mathrm{V}_{j_i}\right) + \delta_{j_i} \tag{4}$$

where V_{j_i} is the digitalized voltage of the j^{th} potentiometer of the i^{th} branch, γ_{j_i} is the gain of the transducer, and δ_{j_i} is a potentiometer specific bias. The correct angular displacement depends on the accuracy of the gains and the offsets.

Given that all nine transducers define the position of the group of spherical joints, i.e., ${}^{b}\boldsymbol{p}_{\mathrm{sA}}$, ${}^{b}\boldsymbol{p}_{\mathrm{sB}}$, and ${}^{b}\boldsymbol{p}_{\mathrm{sC}}$, the distance between two of these points should be equal to the edge-length of the handle, $l_{edge} = 66$ mm. Therefore, the following objective function can be established:

$$\begin{aligned}&\left(\left\|{}^{b}\boldsymbol{p}_{\mathrm{sA}} - {}^{b}\boldsymbol{p}_{\mathrm{sB}}\right\| - l_{edge}\right)^2 + \left(\left\|{}^{b}\boldsymbol{p}_{\mathrm{sA}} - {}^{b}\boldsymbol{p}_{\mathrm{sC}}\right\| - l_{edge}\right)^2 + \\ &\left(\left\|{}^{b}\boldsymbol{p}_{\mathrm{sB}} - {}^{b}\boldsymbol{p}_{\mathrm{sC}}\right\| - l_{edge}\right)^2 = err^2\end{aligned} \tag{5}$$

By squaring every term, it is ensured that the difference among edge–lengths will not cancel out in case one term is positive and the other one negative. The value *err* should be as small as possible. For improved calibration, a large number (approx. 500) of digitalized voltages from the potentiometers are collected and The MATLAB™ function

"*fminunc*" was employed to minimize $Err = \|err_1 \quad err_2 \quad \cdots \quad err_n\|$ where *n* is the number of sampled handle poses. The result is the set of γ_{j_i} and δ_{j_i} values [22].

2.3 Determination of Velocity Commands

The task space velocities with respect to the inertial frame are denoted as $\dot{\boldsymbol{x}}_{ee} = [\dot{\boldsymbol{x}}_p; \quad \dot{\boldsymbol{x}}_o]^T = [\dot{x}_{ee} \quad \dot{y}_{ee} \quad \dot{z}_{ee}; \quad \dot{\phi}_{ee} \quad \dot{\theta}_{ee} \quad \dot{\psi}_{ee}]^T$, where $\dot{\boldsymbol{x}}_{\boldsymbol{p}}$ is the desired translational velocity vector and $\dot{\boldsymbol{x}}_o$ is the desired Euler rates (Roll-Pitch-Yaw) vector of the end-effector commanded by the pilot using the parallel joystick.

2.3.1 Translational Motion

The desired end-effector translational velocities can be related to the position of the parallel joystick according to:

$$\dot{\boldsymbol{x}}_{j,p} = \mathbf{k}_p{}^b \boldsymbol{p} \tag{6}$$

where $\mathbf{k}_p \in \Re^3$ is a positive definite diagonal matrix with the diagonal elements corresponding the sensitivity of the desired velocities to the joystick motion. When $\dot{\boldsymbol{x}}_{j,p}$ is given in terms of the inertial reference frame $\dot{\boldsymbol{x}}_p = \dot{\boldsymbol{x}}_{j,p}$

2.3.2 Rotational Motion

The desired end-effector Euler rates follow:

$$\dot{\boldsymbol{x}}_o = k_o \mathbf{R}_e \tag{7}$$

where $\mathbf{k}_o \in \Re^3$ is a positive definite diagonal matrix with the diagonal elements correspond to the sensitivity of the end effector rates to the joystick motion and $\mathbf{R}_e$ is the approximation for the orientation error defined as [23]:

$$\mathbf{R}_e = 1/2[{}^I_{ee}\mathbf{Rn}_d \times {}^I_{ee}\mathbf{Rn}, {}^I_{ee}\mathbf{Ra}_d \times {}^I_{ee}\mathbf{Ra}, {}^I_{ee}\mathbf{Ro}_d \times {}^I_{ee}\mathbf{Rn}] \tag{8}$$

In Eq. (8), the rotation matrix from the end-effector reference frame to the Inertial reference frame is defined as ${}^I_{ee}\mathbf{R} = \left[{}^I_{ee}\mathbf{Rn} \quad {}^I_{ee}\mathbf{Ra} \quad {}^I_{ee}\mathbf{Ro} \right]$ and the desired orientation is defined as ${}^I_{ee}\mathbf{R}_d = \left[{}^I_{ee}\mathbf{Rn}_d \quad {}^I_{ee}\mathbf{Ra}_d \quad {}^I_{ee}\mathbf{Ro}_d \right]$. While the rotation matrix ${}^I_{ee}\mathbf{R}$ can be obtained from the forward kinematics solution of the ROVM system [24], the desired rotation ${}^I_{ee}\mathbf{R}_d$ is obtained from the orientation of the handle with respect to the base of the Joystick, i.e., ${}^I_{ee}\mathbf{R}_d = {}^b_h\mathbf{R}$.

3 ROVM KINEMATICS

3.1 Fuzzy Redundancy Resolution

The differential kinematic relationship between the task-space velocities $\dot{\boldsymbol{x}}_{ee}$, which corresponds to the pilot command to the end-effector, and the generalized velocities $\dot{\boldsymbol{q}} \in \Re^n$ is given by:

$$\dot{\boldsymbol{q}} = \mathbf{J}_m^{\dagger} \dot{\boldsymbol{x}}_{ee} \tag{9}$$

where $\mathbf{J}_m \in \Re^{m \times n}$ is the Jacobian matrix, $\dot{\boldsymbol{q}} = [\dot{\boldsymbol{q}}_1^T \quad \dot{\boldsymbol{q}}_2^T \quad \boldsymbol{q}_3^T]^T \in \mathrm{R}^n$ is the generalized coordinate vector, consisting of the liner velocity vector $\dot{\boldsymbol{q}}_1 = [\dot{x} \quad \dot{y} \quad \dot{z}]^T$, Euler rates (Roll-Pitch-Yaw) $\dot{\boldsymbol{q}}_2 = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T$ of the vehicle, and joint rates

$\dot{\boldsymbol{q}}_3 = [\dot{q}_1 \quad \dot{q}_2 \quad \ldots \quad \dot{q}_{n-6}]^T$ of the manipulator. Since $n > 6$, there are an infinite set of joint rate solutions that can complete the desired end-effector motion, $\dot{\boldsymbol{x}}$.

The following solution to (9) is implemented:

$$\begin{aligned} \dot{\boldsymbol{q}} &= \dot{\boldsymbol{q}}_p + \dot{\boldsymbol{q}}_h \\ \dot{\boldsymbol{q}} &= \mathbf{J}_{m,w}^{\dagger}\dot{\boldsymbol{x}}_{ee} + \lambda\left(\mathbf{I} - \mathbf{J}_{m,w}^{\dagger}\mathbf{J}\right)\mathbf{W}^{-1}\left(\sum_{j=1}^{s}\alpha_j \nabla h_j \Big/ \left\|\nabla h_j\right\|_2\right) \end{aligned} \tag{10}$$

where $\dot{\boldsymbol{q}}_p = \mathbf{J}_{m,w}^{\dagger}\dot{\boldsymbol{x}}_{ee}$ is the particular solution with $\mathbf{J}_{m,w}^{\dagger} = \mathbf{W}^{-1}\mathbf{J}_m^T\left(\mathbf{J}_m^T\mathbf{W}^{-1}\mathbf{J}_m^T\right)^{-1}$ being the right weighted Moore-Penrose pseudoinverse [25]. The weighting matrix is of the form $\mathbf{W}^{-1} = diag(w_1, \quad w_2, \quad \ldots \quad w_n)$ and is a positive definite matrix of weight factors for each degree of freedom in a robotic system. The particular solution corresponds to the weighted minimum-norm solution associated with (9) that minimizes $\dot{\boldsymbol{q}}^T\mathbf{W}\dot{\boldsymbol{q}}$ subject to $\dot{\boldsymbol{x}}_{ee} = \mathbf{J}_m\dot{\boldsymbol{q}}$. The term $\dot{\boldsymbol{q}}_h$ is the homogeneous solution that satisfies $\mathbf{0} = \mathbf{J}_m\dot{\boldsymbol{q}}_h$. In (10), *s* is the total number of secondary objectives, ∇h_j is the gradient of j^{th} objective function h_j, and α_j is a fuzzy weighting factor defining the relative importance of h_j. The homogeneous solution does not produce any end-effector motion and corresponds to "internal" motions that can be exploited to achieve additional or secondary objectives.

3.2 A Fuzzy hierarchy of Secondary Objectives

In the redundancy resolution scheme, the primary objective is to follow the desired end-effector motion $\dot{\boldsymbol{x}}$ that is commanded by a higher level human pilot motion using a 6-DOF master arm. Regarding the secondary objectives h_j, they include, but are not limited to: avoiding manipulator joint limits, singularity and high joint velocity

avoidance, keeping the end-effector inside the camera view, aligning the vehicle with the ocean current (drag minimization), minimization of the ROV motion. For the joint limit avoidance and the minimal vehicle motion objectives, the weighting matrix **W** is used. In this implementation, $w_i, i = 1 \ldots 6$ are reserved for the minimal vehicle motion objective, whereas $w_i, i = 7 \ldots n$ are reserved for the joint limit avoidance.

The singularity avoidance, the camera view, and the drag force minimization objectives are included through the GPM. The order of priority from the highest to the lowest is established to be *singularity avoidance → camera angle → yaw angle → joint displacement →vehicle motion*. Each objective is defined within the framework of the GPM. The importance of each objective is determined by the artificial pilot. To this end, the artificial pilot assigns a weighting factor $\alpha_j \in [0, 1],\ j = 1...3, v$ for each criterion based on if-then type fuzzy rules that reflect an expert human pilot's knowledge. These rules are defined for the current set of secondary objectives as:

- If the *manipulator* is *singular* then α_1 is *high*;
- If the *manipulator* is *not singular* then α_1 is *low*;
- If the *manipulator* is *not singular* and the *camera angle* is *bad* then α_2 is *high*;
- If the *manipulator* is *singular* or the *camera angle* is *good* then α_2 is *low*;
- If the *manipulator* is *not singular* and the *camera angle* is *good* and the *yaw angle* is *not aligned* then α_3 is *high*;
- If the *manipulator* is *singular* or the *camera angle* is *bad* or the *yaw angle* is *aligned* then α_3 is *low*;
- If the *manipulator* is *not singular* and the *camera angle* is *good* and the *yaw angle* is *aligned* and *joint displacement* is *not close* then α_v is *high*;

- If the *manipulator* is *singular* or the *camera angle* is *bad* or the *yaw angle* is *not aligned* or *joint displacement* is *close* then α_v is *low*.

Greater α values indicate a higher demand, whereas smaller values indicate a lower demand for that particular objective. Note that the fuzzy logic design herein differs from that of [1] in that the input linguistic rules and the corresponding input fuzzy sets of *joint displacement = {close, not close}* were added to the last two rules in the set. This change was made since it was observed that without this addition, the joint limit avoidance may not be realized effectively since when the vehicle motion is heavily penalized, the fuzzy system demands motions from the joints to compensate for the lack of vehicle motion even if the joint displacements are close to their limits. This can potentially drive joint position values beyond their displacement limits.

This rule interpretation process by the artificial pilot is established using a Mamdani fuzzy inference method. The Mamdani Fuzzy Interference method outlines a decision-making process by mimicking the human pilot's terms. The decisions are made based on index values that determine the importance of each particular objective at a given time. In the current work, the monitored index values are the mobility index ($\varepsilon_{\sin}$), the camera angle index (ε_{cam}), and the yaw angle index (ε_{yaw}). These index values are first normalized based on their expected maximum values $\varepsilon_{\max}$, which are obtained from the MATLAB™ function "*fmincon*", and then are fed into the fuzzy block. The resulting weight schedule yields a self-motion (null-space motion) that emulates how a skilled operator would utilize the redundancy of the ROVM to achieve the secondary objectives.

The redundancy resolution method employed in this work has a fault tolerant property. The fault tolerance method prevents the redundancy resolver from requesting intolerable joint motions in the event of joint failure. According to the employed redundancy resolution method, joint failure refers to three incidents; first, it occurs when a joint breaks down, second, it happens when a joint has partly lost its driving capacity, and third, it takes place when the redundancy resolution scheme produces joint velocities that are beyond the capacity of the corresponding joints. A more detailed explanation of this scheme along with its fault-tolerant version can be found in Soylu *et al*. [1]

4 CONTROL

4.1 Dynamic Modelling

The derivation of the equations of motion for a ROVM system are more convenient when defined in terms of state values expressed in the body-fixed frames since on-board sensors (actuators) measure (create) rates (forces) specific to the axes of body fixed frames. The quasi-Lagrange approach presented by Meirovitch [17] is well suited to this end. The quasi-Lagrange approach was applied to ROVM modeling by [16], Kim *et al*. [26], Soylu *et al*. [15]. In the current work, the work of Soylu *et al*. [15] is followed.

The ROVM dynamics, in terms of body-fixed frame quantities, follow:

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{q}, \boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{q}, \boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{11}$$

The body fixed velocity vector $\boldsymbol{\nu} \in \Re^n$ is defined as:

$$\begin{aligned} \boldsymbol{\nu} &= \begin{bmatrix} \boldsymbol{\nu}_1^T & \boldsymbol{\nu}_2^T & \boldsymbol{\nu}_3^T \end{bmatrix}^T, \ \boldsymbol{\nu}_1 = \begin{bmatrix} u & v & w \end{bmatrix}^T, \\ \boldsymbol{\nu}_2 &= \begin{bmatrix} p & q & r \end{bmatrix}^T, \quad \boldsymbol{\nu}_3 = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \dots & \dot{q}_{n-6} \end{bmatrix}^T \end{aligned} \tag{12}$$

with $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$ being body-fixed translational and rotational velocity vectors of the ROV, and the manipulator joint rates, respectively. The term $\mathbf{M}(\boldsymbol{q}) \in \Re^{n \times n}$ is the inertia matrix including the added mass effects, $\mathbf{C}(\boldsymbol{q}) \in \Re^{n \times n}$ is the matrix of centrifugal and Coriolis terms, $\mathbf{D}(\boldsymbol{q}) \in \Re^{n \times n}$ is the drag matrix, $\mathbf{g}(\boldsymbol{q}) \in \Re^{n}$ is the vector of gravity and buoyancy forces and moments, and finally $\boldsymbol{\tau} \in \Re^{n}$ is the control forces and moments acting on the ROV centre of mass.

Equation (11) can also be represented with respect to the inertial reference frame, as discussed by Fossen [27]:

$$\boldsymbol{f} = \mathbf{M}_q(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{C}_q(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{D}_q(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_q(\boldsymbol{q}) + \boldsymbol{d} = \mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau} \tag{13}$$

where $\mathbf{M}_q(\boldsymbol{q}) = \mathbf{P}^{-\mathrm{T}}\mathbf{M}\mathbf{P}^{-1}$, $\mathbf{C}_q(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathbf{P}^{-\mathrm{T}}\left[\mathbf{C} - \mathbf{M}\mathbf{P}^{-1}\dot{\mathbf{P}}\right]\mathbf{P}^{-1}$, $\mathbf{D}_q(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathbf{P}^{-\mathrm{T}}\mathbf{D}\mathbf{P}^{-1}$, and $\boldsymbol{g}_q(\boldsymbol{q}) = \mathbf{P}^{-\mathrm{T}}\boldsymbol{g}$ with $\mathbf{P} \in \Re^{6\times 6}$ the spatial transformation matrix between the inertial frame and the ROV's body fixed frame that can be defined through the Euler angle transformation [27]. The term $\boldsymbol{d} \in \Re^{n}$ is the unknown disturbance acting on the system which could comprise errors in the estimation of the hydrodynamics or a tether disturbance.

Equation (13) can be rewritten in a more compact form of as:

$$\boldsymbol{f} = \mathbf{M}_q\ddot{\boldsymbol{q}} + \boldsymbol{h}_q(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{14}$$

where $\boldsymbol{h}_q(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathbf{C}_q(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{D}_q(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_q(\boldsymbol{q}) + \boldsymbol{d}$.

The ROVM dynamics are assumed to have the following structural properties [27]:

Property 1: The inertia matrix $\mathbf{M}_q$ is symmetric and positive definite, i.e., $\mathbf{M}_q^{\mathrm{T}} = \mathbf{M}_q$;

Property 2: Matrix $\dot{\mathbf{M}}_q - 2\mathbf{C}_q$ is skew symmetric, i.e., for any vector $\boldsymbol{\zeta}$, $\boldsymbol{\zeta}^{\mathrm{T}} (\dot{\mathbf{M}}_q - 2\mathbf{C}_q) \boldsymbol{\zeta} = 0$.

4.2 Definition of Lumped Uncertainty

The ROVM dynamics are dominated by hydrodynamic loads, and it is difficult to accurately measure or estimate the hydrodynamic coefficients that are valid for all ROV operating conditions and instrument or tether configurations. As such, the system dynamics are not exactly known. Therefore, the true system dynamics given in Eq. (13) can be written as the sum of estimated dynamics $\hat{\boldsymbol{f}}$ and the unknown dynamics $\tilde{\boldsymbol{f}}$:

$$\boldsymbol{f} = \hat{\boldsymbol{f}} + \tilde{\boldsymbol{f}} \tag{15}$$

where the estimated dynamics vector is defined as

$$\hat{\boldsymbol{f}} = \hat{\mathbf{M}}_q \ddot{\boldsymbol{q}} + \hat{\boldsymbol{h}}_q (\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{16}$$

with $\hat{\boldsymbol{h}}_q (\boldsymbol{q}, \dot{\boldsymbol{q}}) = \hat{\mathbf{C}}_q (\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} + \hat{\mathbf{D}}_q (\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} + \hat{\boldsymbol{g}}_q (\boldsymbol{q}) + \boldsymbol{d}.$ and the unknown dynamics vector defined as:

$$\tilde{\boldsymbol{f}} = \tilde{\mathbf{M}}_\eta (\boldsymbol{q}) \ddot{\boldsymbol{q}} + \tilde{\boldsymbol{h}}_\eta (\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{17}$$

with $\tilde{\boldsymbol{h}}_q (\dot{\boldsymbol{q}}, \boldsymbol{\eta}) = \tilde{\mathbf{C}}_q (\dot{\boldsymbol{q}}, \boldsymbol{\eta}) \dot{\boldsymbol{\eta}} + \tilde{\mathbf{D}}_q (\dot{\boldsymbol{q}}, \boldsymbol{\eta}) \dot{\boldsymbol{\eta}} + \tilde{\boldsymbol{g}}_q (\boldsymbol{\eta})$, where

$$\tilde{\mathbf{M}}_\eta = \mathbf{M}_\eta - \hat{\mathbf{M}}_\eta,\ \tilde{\mathbf{C}}_\eta = \mathbf{C}_\eta - \hat{\mathbf{C}}_\eta,$$
$$\tilde{\mathbf{D}}_\eta = \mathbf{D}_\eta - \hat{\mathbf{D}}_\eta,\quad \tilde{\mathbf{g}}_\eta = \mathbf{g}_\eta - \hat{\mathbf{g}}_\eta \tag{18}$$

The unknown dynamics vector is also called the lumped uncertainty vector and has been proposed for use in adaptive control by Lewis *et al*. [16], Lin *et al*. [28], and Lin and Wai [29].

Assumption 1: The two norm of the lumped uncertainty vector $\tilde{\boldsymbol{f}} = \begin{bmatrix} \tilde{f}_1 & \tilde{f}_2 & \dots & \tilde{f}_n \end{bmatrix}^T$ given in (17) is bounded by ρ, i.e., $\left\| \tilde{\boldsymbol{f}} \right\|_2 \le \rho$.

Definition 1: The error in the estimation of ρ is defined as $\tilde{\rho} = \rho - \hat{\rho}$ with $\hat{\rho}$ being the estimate of ρ .

4.3 Control Law

The first step in the current controller design procedure is to define an error function $\boldsymbol{s} \in \Re^{n\times 1}$, commonly referred to as a sliding manifold [30]. The second step is to find a control law $\boldsymbol{\tau} \in \Re^{n\times 1}$ such that the error function converges to zero in the presence of system uncertainties and disturbances [30].

The error function can be defined as:

$$\boldsymbol{s} = \left(\frac{d}{dt} + \boldsymbol{\Lambda} \right)^2 \left(\int \boldsymbol{e} dt \right) = \dot{\boldsymbol{e}} + 2\boldsymbol{\Lambda}\boldsymbol{e} + \boldsymbol{\Lambda}^2 \int \boldsymbol{e} dt \tag{19}$$

where $\boldsymbol{\Lambda} \in \Re^{n\times n}$, is a constant, symmetric, positive definite and diagonal matrix The tracking error between the measured state values and the desired state values is given by $\boldsymbol{e} = \boldsymbol{q} - \boldsymbol{q}_d$ with the subscript *d* denoting the desired position values produced by the

redundancy resolver corresponding to the integral of (10), i.e., $\boldsymbol{q} = \int \dot{\boldsymbol{q}} dt$. The integral term in Eq. (19) ensures zero offset error. For notational simplicity, Eq. (19) can also be written as:

$$\boldsymbol{s} = \dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_r \tag{20}$$

where $\dot{\boldsymbol{q}}_r = \dot{\boldsymbol{q}}_d - 2\boldsymbol{\Lambda}\,\mathbf{e} - \boldsymbol{\Lambda}^2 \int \mathbf{e}\, dt,$ and the subscript *r* refers to virtual reference trajectory [30].

The proposed control law is in the form:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{eq} - \boldsymbol{\tau}_{PID} - \boldsymbol{\tau}_{robust} \tag{21}$$

where $\boldsymbol{\tau}$ corresponds to a generalized force acting at the centre of mass of the ROV, and $\boldsymbol{\tau}_{eq}$ is the equivalent control law, $\boldsymbol{\tau}_{PID}$ is the PID control law and finally $\boldsymbol{\tau}_{robust}$ is the robust control signal.

4.3.1 Equivalent Control

The equivalent control law is continuous and model based. In the absence of uncertainties in the system dynamics, this equivalent control alone suffices to realize the desired dynamics.

The model based equivalent control law component of the sliding-mode control signal can be derived by assuming that the motion is constrained to the sliding manifold; i.e., $\dot{\boldsymbol{s}} = \mathbf{0}$. Taking the time derivative of Eq. (20) yields:

$$\dot{\boldsymbol{s}} = \ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_r \tag{22}$$

where $\ddot{\boldsymbol{q}}_r = \ddot{\boldsymbol{q}}_d - 2\boldsymbol{\Lambda}\dot{\mathbf{e}} - \boldsymbol{\Lambda}^2\mathbf{e}\,dt$. Multiplying both sides of Eq. (22) by the inertia matrix $\hat{\mathbf{M}}_q$, and substituting $\mathbf{M}_q\ddot{\boldsymbol{q}} = \mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau}_{eq} - \left(\mathbf{C}_q\dot{\boldsymbol{q}} + \mathbf{D}_q\dot{\boldsymbol{q}} + \boldsymbol{g}_q\right)$ from Eq. (13) into the resulting equation yields:

$$\hat{\mathbf{M}}_q\dot{\boldsymbol{s}} = \mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau}_{eq} - \left(\hat{\mathbf{M}}_q\ddot{\boldsymbol{q}}_r + \hat{\mathbf{C}}_q\dot{\boldsymbol{q}} + \hat{\mathbf{D}}_q\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}}_q\right). \tag{23}$$

Letting $\dot{\boldsymbol{s}} = \mathbf{0}$ and solving the resulting equation for $\boldsymbol{\tau}_{eq}$ yield the equivalent control as:

$$\boldsymbol{\tau}_{eq} = \mathbf{J}^{\mathrm{T}}\hat{\boldsymbol{f}}_r(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_r) \tag{24}$$

where

$$\hat{\boldsymbol{f}}_r(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_r) = \hat{\mathbf{M}}_q\ddot{\boldsymbol{q}}_r + \hat{\mathbf{C}}_q\dot{\boldsymbol{q}} + \hat{\mathbf{D}}_q\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}}_q \tag{25}$$

As mentioned before, in the absence of uncertainties in the system dynamics, this equivalent control alone can keep the state variables on the sliding surface.

4.3.2 Adaptive PID Control Design

In this section, an adaptive PID control signal with a new adaptation law is derived. The PID control signal can be defined as:

$$\boldsymbol{\tau}_{PID} = \mathbf{J}^{\mathrm{T}}\left(\mathbf{K}_{\mathbf{D}}\dot{e} + \mathbf{K}_{\mathbf{P}}e + \mathbf{K}_{\mathbf{I}}\int e\right) \tag{26}$$

where $\mathbf{K}_{\mathrm{D}} \in \Re^{\mathrm{n}\times\mathrm{n}}$, $\mathbf{K}_{\mathrm{P}} \in \Re^{\mathrm{n}\times\mathrm{n}}$, $\mathbf{K}_{\mathrm{I}} \in \Re^{\mathrm{n}\times\mathrm{n}}$ are diagonal derivative, proportional and integral gain matrices, respectively. Equation (26) can be written as:

$$\boldsymbol{\tau}_{PID} = \mathbf{J}^{\mathrm{T}}\begin{bmatrix}\mathbf{K}_{\mathbf{D}} & \mathbf{K}_{\mathbf{P}} & \mathbf{K}_{\mathbf{I}}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{e}}^T & \boldsymbol{e}^T & \int\boldsymbol{e}^T\end{bmatrix}^T = \mathbf{J}^{\mathrm{T}}\mathbf{K}\boldsymbol{r} \tag{27}$$

where $\mathbf{K} = [\mathbf{K_D} \quad \mathbf{K_P} \quad \mathbf{K_I}] \in \Re^{n\times 3n}$ is the gain matrix and $\boldsymbol{r} = [\dot{\boldsymbol{e}}^{\mathrm{T}} \quad \boldsymbol{e}^{\mathrm{T}} \quad \int \boldsymbol{e}^{\mathrm{T}}]^{T} \in \Re^{3n\times 1}$ is the error vector. The following adaptation law is proposed for the PID gains:

$$\dot{\mathbf{K}}^{\mathrm{T}} = \boldsymbol{\beta}\left(\boldsymbol{r}\boldsymbol{s}^{\mathrm{T}} + \kappa_1\left(\mathbf{K}_0 - \mathbf{K}\right)^{\mathrm{T}}\right) \tag{28}$$

where the term $\boldsymbol{\beta} \in \Re^{3n\times 3n}$ is a positive definite diagonal matrix and $\mathbf{K}_0$ is the user-defined constant diagonal gain matrix. Note that the direct implementation of Eq. (28) causes the off diagonal elements of $\mathbf{K_D}$, $\mathbf{K_P}$, and $\mathbf{K_I}$ to become nonzero; leading to potential instability due non-zero, off-diagonal terms. In addition, it creates an unnecessary computational burden. In order to eliminate these complications, the following modification is made to (28):

$$\dot{\mathbf{K}}^{\mathrm{T}} = \boldsymbol{\beta}\left(\Delta\left(\boldsymbol{r}\boldsymbol{s}^{\mathrm{T}}\right) + \kappa_1\left(\mathbf{K}_0 - \mathbf{K}\right)^{\mathrm{T}}\right) \tag{29}$$

where $\Delta(\cdot)$ is an operator that zeros the off-diagonal elements of the $n\times n$ sub-matrices of the $3n\times n$ matrix of $\boldsymbol{r}\boldsymbol{s}^{\mathrm{T}}$. For example for $n = 2$, the $\Delta(\cdot)$ operator yields:

$$\Delta\left(\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} m_{31} & m_{32} \\ m_{41} & m_{42} \end{bmatrix} \begin{bmatrix} m_{51} & m_{52} \\ m_{61} & m_{62} \end{bmatrix}\right) = \begin{bmatrix} \begin{bmatrix} m_{11} & 0 \\ 0 & m_{22} \end{bmatrix} \begin{bmatrix} m_{31} & 0 \\ 0 & m_{42} \end{bmatrix} \begin{bmatrix} m_{51} & 0 \\ 0 & m_{62} \end{bmatrix}\end{bmatrix}$$

With this modification, the adaptive term $\dot{\mathbf{K}}$ is of the form:

$$\dot{\mathbf{K}} = \begin{bmatrix} \begin{bmatrix} \dot{k}_{D,1} & 0 & \dots & 0 \\ 0 & \dot{k}_{D,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dot{k}_{D,n} \end{bmatrix}_{n\times n} & \begin{bmatrix} \dot{k}_{P,1} & 0 & \dots & 0 \\ 0 & \dot{k}_{P,1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dot{k}_{P,1} \end{bmatrix}_{n\times n} & \begin{bmatrix} \dot{k}_{I,1} & 0 & \dots & 0 \\ 0 & \dot{k}_{I,1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dot{k}_{I,1} \end{bmatrix}_{n\times n} \end{bmatrix}_{n\times 3n} \tag{30}$$

and hence, does not yield nonzero off-diagonal terms in the sub gain matrices of $\mathbf{K_D}$, $\mathbf{K_P}$, and $\mathbf{K_I}$ provided that the initial off-diagonal terms of each sub-gain matrix of $\mathbf{K}$ and $\mathbf{K}_0$ are set to be zero. In addition, with the devised operator, it can be readily shown that $tr(\mathbf{K}\boldsymbol{r}\boldsymbol{s}^T) = tr(\mathbf{K}\Delta(\boldsymbol{r}\boldsymbol{s}^T))$ with $tr(\cdot)$ denoting the trace of a matrix. This equality is required for the stability analysis of the proposed controller as shown in Section 4.3.5.

4.3.3 Upper Bound Estimation Algorithm

The following robust control signal is defined:

$$\boldsymbol{\tau}_{rob} = \mathbf{J}^{\mathrm{T}}\left(\mathrm{sgn}(\boldsymbol{s})\hat{\rho} + \hat{\mathrm{C}}_q\boldsymbol{s}\right) \tag{31}$$

with the following bound estimation law:

$$\dot{\hat{\rho}} = \upsilon\left(\|\boldsymbol{s}\|_2 + \kappa_2\left(\rho_0 - \hat{\rho}\right)\right) \tag{32}$$

where κ, λ and ρ_0 are strictly positive, user defined scalars. Therefore, the total control signal $\boldsymbol{\tau} = \boldsymbol{\tau}_{eq} - \boldsymbol{\tau}_{PID} - \boldsymbol{\tau}_{robust}$ is defined as:

$$\boldsymbol{\tau} = \mathbf{J}^{\mathrm{T}}\left(\hat{\boldsymbol{f}}_r(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_r) - \mathbf{K}\boldsymbol{r} - \hat{\rho}\,\mathrm{sgn}(\boldsymbol{s}) - \hat{\mathbf{C}}_q\boldsymbol{s}\right) \tag{33}$$

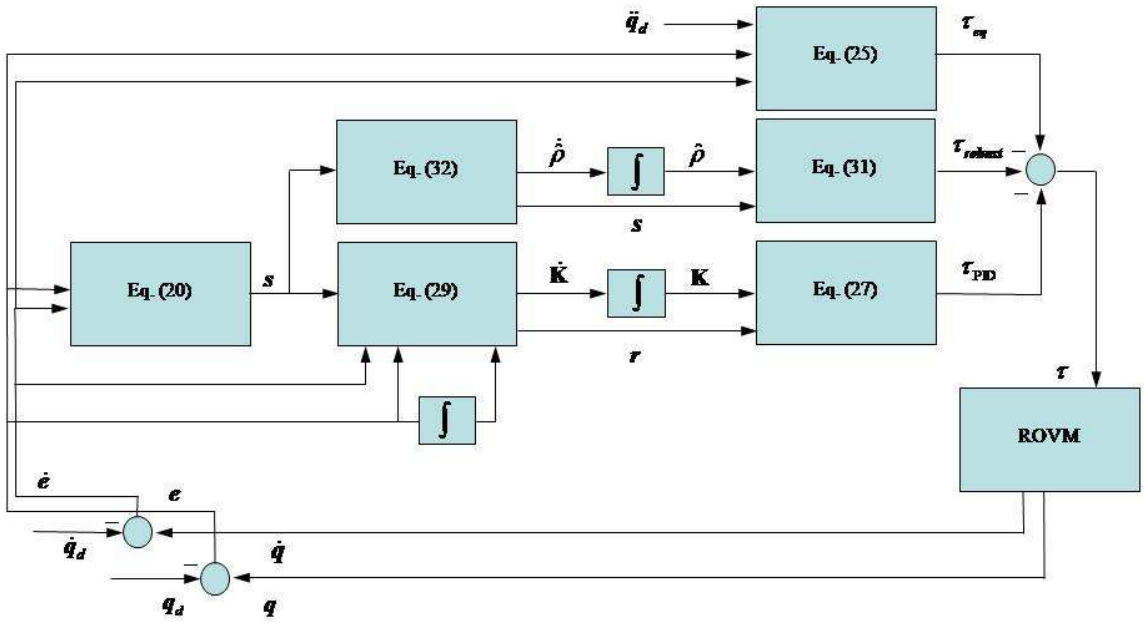


Figure 3: Block diagram of the proposed controller.

Note that the upper bound adaptation algorithm given in Eq. (32) is different than the one presented in [31] in that it allows the upper bound to decrease. In [31], the upper bound adaptation law constantly leads to the increase in $\hat{\rho}$ when $\boldsymbol{s}$ is outside the above a threshold value, and keeps it at the previous value when below the threshold value. This may lead to an excessively large upper bound estimate and, subsequently, aggressive controller behaviour. The proposed algorithm eliminates this problem by letting the upper bound estimation $\hat{\rho}$ increase and decrease depending on the value of the error metric $\boldsymbol{s}$ at a given time. Similar adaptation law, which works based on the first-norm of the error metric $\boldsymbol{s}$ is presented in [32].

In adaptation laws of Eq. (29) and Eq. (32), the parameters $\mathbf{K}_0$ and ρ_0 can be considered as initial estimation of $\mathbf{K}$ and ρ. A better controller performance can be obtained by selecting $\mathbf{K}_0$ and ρ_0 values that are close to their actual values.

4.3.4 Persistency of Excitation

A vector $\boldsymbol{x}(t)$ is said to be persistently exciting if there exists α_1, α_2 and $\Delta t > 0$ for any $t > 0$ such that

$$\alpha_1 \mathbf{I} \le \int_{t_o}^{t_o+\Delta t} \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}} d\tau \le \alpha_2 \mathbf{I} \tag{34}$$

where $\mathbf{I}$ is a unit matrix. The persistency of excitation indicates that the content of input signal is rich enough to excite all parameters in the system so that the adaptation algorithm can converge to the true values of the estimated parameters. The unboundness of an estimated signal when the persistency of excitation condition fails to hold is referred to as “parameter drift” problem [30].

In the current study, this problem is circumvented by adding a leakage term [32], i.e., $\kappa_1 \left(\mathbf{K}_0 - \mathbf{K}\right)^{\mathrm{T}}$ and $\kappa_2 \left(\rho_0 - \hat{\rho}\right)$, based on a version of the $\sigma-$modification technique [19]. With the addition of the leakage terms, no persistency of excitation assumption is necessary on the input signal for accurate estimations.

The user-defined parameters κ_1 and κ_2 determine the degree of trade-off between the tracking performance and the robustness of the control signal to parameter drift. Perfect tracking can be obtained provided that the persistency of excitation condition is satisfied

and κ_1 and κ_2 are set to zero. However, since the persistency of excitation condition is difficult to satisfy in practice all the time, κ_1 and κ_2 are set as $\kappa_1 > 0$ and $\kappa_2 > 0$.

4.3.5 Stability Analysis

Definition: A nonlinear system such as $\dot{x} = f(x,u), y = h(x)$ where *x* is a state vector and *u* is the input vector, is said to be uniformly ultimately bounded (**UUB**), if there is a compact set $S \subset \Re^n$ so that for all $s(t_o) = s_o \in S$ there exists a bound $\varepsilon > 0$ and a time $T(\varepsilon, s_o)$ such that $\|\boldsymbol{s}\| < \varepsilon$ for all $t \geq t_o + T$ [33].

Theorem: Consider the nonlinear dynamical system described by Eq. (13) with Assumption 1 and Definition 1. If the control law is expressed as Eq. (33) with the adaptation laws of Eq. (29) and Eq. (32), then the tracking error metric ***s*** is UUB.

Lemma: Consider the following Lyapunov function candidate:

$$V = \frac{1}{2}\left(\boldsymbol{s}^{\mathrm{T}}\mathbf{M}_q\mathbf{s} + tr(\mathbf{K}\boldsymbol{\beta}^{-1}\mathbf{K}^{\mathrm{T}}) + \upsilon^{-1}\tilde{\rho}^2\right) \tag{35}$$

Defining the vector $\boldsymbol{n} = \begin{bmatrix}\boldsymbol{s}^{\mathrm{T}} & \boldsymbol{k}_1 & \ldots & \boldsymbol{k}_{3n} & \tilde{\rho}\end{bmatrix}$ with $\mathbf{K} = [\boldsymbol{k}_1 \ \ldots \ \boldsymbol{k}_{3n}]$, the Lyapunov function given in Eq. (35) satisfies $\frac{1}{2}P_m\boldsymbol{n}^{\mathrm{T}}\boldsymbol{n} \leq V \leq \frac{1}{2}P_M\boldsymbol{n}^{\mathrm{T}}\boldsymbol{n}$ where $P_m = \min\left(\lambda_{\mathbf{M},\min}, \lambda_{\boldsymbol{\beta}^{-1},\min}, \upsilon^{-1}\right)$ and $P_M = \max\left(\lambda_{\mathbf{M},\max}, \lambda_{\boldsymbol{\beta}^{-1},\max}, \upsilon^{-1}\right)$ with $\lambda_{\mathbf{M},\min}$ and $\lambda_{\mathbf{M},\max}$ being the minimum and maximum eigenvalues of the mass matrix $\mathbf{M}_q$, respectively, and $\lambda_{\boldsymbol{\beta}^{-1},\min}$ and $\lambda_{\boldsymbol{\beta}^{-1},\max}$ being the minimum and maximum eigenvalues of the $\boldsymbol{\beta}$, respectively. This result is due to the following inequalities:

$$\begin{aligned}\lambda_{\mathbf{M},\min}\|\boldsymbol{s}\|_2^2 \le \boldsymbol{s}^{\mathrm{T}}\mathbf{M}_q\boldsymbol{s} \le \lambda_{\mathbf{M},\max}\|\boldsymbol{s}\|_2^2 \\ \lambda_{\boldsymbol{\beta}^{-1},\min}\|\mathbf{K}\|_F^2 \le tr(\mathbf{K}\boldsymbol{\beta}^{-1}\mathbf{K}^{\mathrm{T}}) \le \lambda_{\boldsymbol{\beta}^{-1},\max}\|\mathbf{K}\|_F^2\end{aligned} \tag{36}$$

with $\|\mathbf{K}\|_F$ being the Frobenius form defined as $\|\mathbf{K}\|_F^2 = tr\left(\mathbf{K}\mathbf{K}^{\mathrm{T}}\right)$.

Therefore, it can be concluded that:

$$\frac{V}{P_M} \le \frac{1}{2}\boldsymbol{n}^{\mathrm{T}}\boldsymbol{n} \tag{37}$$

Proof: Differentiating the Lyapunov function, given in (35), with respect to time yields;

$$\begin{aligned}\dot{V} &= \frac{1}{2}\left(\boldsymbol{s}^{\mathrm{T}}\dot{\mathbf{M}}_q\boldsymbol{s} + \dot{\boldsymbol{s}}^{\mathrm{T}}\mathbf{M}_q\boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}\mathbf{M}_q\dot{\boldsymbol{s}}\right) + tr(\mathbf{K}\boldsymbol{\beta}^{-1}\dot{\mathbf{K}}^{\mathrm{T}}) + \upsilon^{-1}\tilde{\rho}\dot{\tilde{\rho}} \\ &= \frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\left(\dot{\mathbf{M}}_q - 2\mathbf{C}_q\right)\boldsymbol{s} + \boldsymbol{s}^{T}\left[\mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau} - \left(\mathbf{M}_q\ddot{\boldsymbol{q}}_r + \mathbf{C}_\eta\dot{\boldsymbol{q}}_r + \mathbf{D}_\eta\dot{\boldsymbol{q}} + \boldsymbol{g}_q\right)\right] \\ &\quad + tr(\mathbf{K}\boldsymbol{\beta}^{-1}\dot{\mathbf{K}}^{\mathrm{T}}) + \upsilon^{-1}\tilde{\rho}\dot{\tilde{\rho}}\end{aligned} \tag{38}$$

Since $\dot{\mathbf{M}}_q - 2\mathbf{C}_q$ is a skew symmetric matrix, then $\boldsymbol{s}^{\mathrm{T}}\left(\dot{\mathbf{M}}_q - 2\mathbf{C}_q\right)\boldsymbol{s} = \mathbf{0}$, and Eq. (38) can be simplified to:

$$\dot{V} = \boldsymbol{s}^{T}\left(\mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau} - \boldsymbol{f}_{\underline{r}}\right) + tr(\mathbf{K}\boldsymbol{\beta}^{-1}\dot{\mathbf{K}}^{\mathrm{T}}) + \upsilon^{-1}\tilde{\rho}\dot{\tilde{\rho}} \tag{39}$$

where $\boldsymbol{f}_{\underline{r}} = \mathbf{M}_q\ddot{\boldsymbol{q}}_r + \mathbf{C}_q\dot{\boldsymbol{q}}_{\mathrm{r}} + \mathbf{D}_q\dot{\boldsymbol{q}} + \boldsymbol{g}_q$. Substituting Eq. (33) into Eq. (39), and using $\dot{\boldsymbol{q}}_r = \dot{\boldsymbol{q}} - \boldsymbol{s}$ and $\tilde{\boldsymbol{f}} = \boldsymbol{f} - \hat{\boldsymbol{f}}$ yields:

$$\dot{V} = \boldsymbol{s}^{\mathrm{T}}\left(-\tilde{\boldsymbol{f}} - \mathbf{K}\boldsymbol{r} - sign(\boldsymbol{s})\hat{\rho}\right) + tr(\mathbf{K}\boldsymbol{\beta}^{-1}\dot{\mathbf{K}}^{\mathrm{T}}) + \upsilon^{-1}\tilde{\rho}\dot{\tilde{\rho}} \tag{40}$$

Using $\boldsymbol{s}^{\mathrm{T}}\mathbf{K}\boldsymbol{r} = tr(\mathbf{K}\boldsymbol{r}\boldsymbol{s}^{\mathrm{T}}) = tr(\mathbf{K}\Delta(\boldsymbol{r}\boldsymbol{s}^{\mathrm{T}}))$, $\dot{\tilde{\rho}} = -\dot{\hat{\rho}}$ and $\boldsymbol{s}^{\mathrm{T}}\hat{\rho}sign(\boldsymbol{s}) = \hat{\rho}\|\boldsymbol{s}\|_1$ with $\|\boldsymbol{s}\|_1 = \sum_{i=1}^{n}|s_i|$ yields;

$$\begin{aligned}\dot{V} &= -\boldsymbol{s}^{\mathrm{T}}\,\tilde{\boldsymbol{f}} - tr\left(\mathbf{K}\Delta\left(\boldsymbol{r}\boldsymbol{s}^{\mathrm{T}}\right)\right) - \hat{\rho}\|\boldsymbol{s}\|_1 + tr\left(\mathbf{K}\boldsymbol{\beta}^{-1}\dot{\mathbf{K}}^{\mathrm{T}}\right) - \upsilon^{-1}\tilde{\rho}\dot{\hat{\rho}} \\ &= -\boldsymbol{s}^{\mathrm{T}}\,\tilde{\boldsymbol{f}} + tr\left(\mathbf{K}\boldsymbol{\beta}^{-1}\dot{\mathbf{K}}^{\mathrm{T}} - \mathbf{K}\Delta\left(\boldsymbol{r}\boldsymbol{s}^{T}\right)\right) - \hat{\rho}\|\boldsymbol{s}\|_1 - \upsilon^{-1}\tilde{\rho}\dot{\hat{\rho}}\end{aligned} \tag{41}$$

With the adaptation laws of Eq. (29) and (32) and considering the fact that $-\boldsymbol{s}^{\mathrm{T}}\,\tilde{\boldsymbol{f}} \le \|\boldsymbol{s}\|_2 \left\|\tilde{\boldsymbol{f}}\right\|_2$ $\|\boldsymbol{s}\|_1 \ge \|\boldsymbol{s}\|_2$ and $tr\left(\mathbf{K}\mathbf{K}_0^{\mathrm{T}}\right) \le \|\mathbf{K}_0\|_F \|\mathbf{K}\|_F$;

$$\dot{V} \le \|\boldsymbol{s}\|_2 \left\|\tilde{\boldsymbol{f}}\right\|_2 - \kappa_1\|\mathbf{K}\|_F^2 + \kappa_1\|\mathbf{K}_0\|_F \|\mathbf{K}\|_F - \hat{\rho}\|\boldsymbol{s}\|_2 - \tilde{\rho}\|\boldsymbol{s}\|_2 - \tilde{\rho}\kappa_2\left(\rho_0 - \hat{\rho}\right) \tag{42}$$

Using $-\tilde{\rho}\kappa_2\left(\rho_0 - \hat{\rho}\right) = -1/2\,\kappa_2\tilde{\rho}^2 - 1/2\,\kappa_2\left(\rho_0 - \hat{\rho}\right)^2 + 1/2\,\kappa_2\left(\rho_0 - \rho\right)$ yields;

$$\begin{aligned}\dot{V} &\le \|\boldsymbol{s}\|_2\left(\left\|\tilde{\boldsymbol{f}}\right\|_2 - \hat{\rho} - \tilde{\rho}\right) - \kappa_1\|\mathbf{K}\|_F^2 + \kappa_1\|\mathbf{K}_0\|_F \|\mathbf{K}\|_F \\ &\quad -1/2\,\kappa_2\tilde{\rho}^2 - 1/2\,\kappa_2\left(\rho_0 - \hat{\rho}\right)^2 + 1/2\,\kappa_2\left(\rho_0 - \rho\right)^2 \\ &\le \|\boldsymbol{s}\|_2\left(\left\|\tilde{\boldsymbol{f}}\right\|_2 - \rho\right) - \kappa_1\|\mathbf{K}\|_F^2 + \kappa_1\|\mathbf{K}_0\|_F \|\mathbf{K}\|_F - 1/2\,\kappa_2\tilde{\rho}^2 + 1/2\,\kappa_2\left(\rho_0 - \rho\right)^2 \\ &= \|\boldsymbol{s}\|_2^2\left(\left\|\tilde{\boldsymbol{f}}\right\|_2 - \rho\right)/\|\boldsymbol{s}\|_2 - \kappa_1\|\mathbf{K}\|_F^2 + \kappa_1\|\mathbf{K}_0\|_F \|\mathbf{K}\|_F - 1/2\,\kappa_2\tilde{\rho}^2 + 1/2\,\kappa_2\left(\rho_0 - \rho\right)^2\end{aligned} \tag{43}$$

using Assumption 1, i.e., $\left\|\tilde{\boldsymbol{f}}\right\|_2 \le \rho$ leads to:

$$\begin{aligned}\dot{V} &\le -\|\boldsymbol{s}\|_2^2\left(\rho - \left\|\tilde{\boldsymbol{f}}\right\|_2\right)/\|\boldsymbol{s}\|_2 - 1/2\,\kappa_2\tilde{\rho}^2 + 1/2\,\kappa_2\left(\rho_0 - \rho\right)^2 + \kappa_1\|\mathbf{K}_0\|_F \|\mathbf{K}\|_F - \kappa_1\|\mathbf{K}\|_F^2 \\ &= -\sigma + \eta\end{aligned} \tag{44}$$

where $\eta = 1/2\,\kappa_2\left(\rho_0 - \rho\right)^2 + \kappa_1\|\mathbf{K}_0\|_F \|\mathbf{K}\|_F$, $\sigma = \|\boldsymbol{s}\|_2^2\left(\rho - \left\|\tilde{\boldsymbol{f}}\right\|_2\right)/\|\boldsymbol{s}\|_2 + \kappa_1\|\mathbf{K}\|_F^2 + 1/2\,\kappa_2\tilde{\rho}^2$.

Defining $\mu = \frac{1}{2}\min\left(2\left(\rho - \left\|\tilde{\boldsymbol{f}}\right\|_2\right)/\|\boldsymbol{s}\|_2,\ 2\kappa_1,\ \kappa_2\right)$, it can be shown that

$$\frac{1}{2}\mu\boldsymbol{n}^{\mathrm{T}}\boldsymbol{n} \le \sigma \tag{45}$$

Using Eq. (37) of Lemma 1, i.e., $\frac{V}{P_M} \leq \frac{1}{2}\boldsymbol{n}^{\mathrm{T}}\boldsymbol{n}$, and Eq. (45), the following inequality can be written:

$$\sigma \geq \frac{\mu V}{P_M} \tag{46}$$

using Eq. (44) and Eq. (46), it can be concluded that:

$$\dot{V} \leq -\sigma + \eta \leq -\frac{\mu V}{P_M} + \eta \tag{47}$$

By defining $\varsigma = \mu / P_M$, multiplying both sides of Eq. (47) by the integrating factor $e^{\int \varsigma dt} = e^{\varsigma t}$, taking the integral of the resulting equation and considering the fact that the Lyapunov function is positive definite, the following inequality can be written:

$$0 \leq V(t) \leq \frac{\eta}{\varsigma} + \left[V(0) - \frac{\eta}{\varsigma} \right] e^{-\varsigma t} \tag{48}$$

Eq. (48) implies that all the signals in the closed loop system including the tracking error metrics $\boldsymbol{s}$ is **UUB** and $\|\boldsymbol{s}\|_2$ will eventually enter into the residual set with the size of $O(\eta/\varsigma)$. This concludes the proof.

4.4 THRUSTER ALLOCATION

The control input $\boldsymbol{\tau} \in \Re^{n\times 1}$ given in (33) is distributed over several conventional thrusters arranged on the ROV chassis. To ensure manoeuvrability, the thruster arrangement is redundant; there are more thrusters than there are active vehicle degrees of freedom. Due to the excess number of thrusters, there is an infinite number of ways to

allocate the pilot's commanded thrust over the existing thrusters. As such, an optimal thrust allocation must be selected by applying criteria to distinguish the various options. In the current work, the infinity-norm based thrust allocation approach presented by Soylu *et al*. [14] was used as a solution criterion.

4.4.1 Infinity norm Thrust Allocation

The control input $\boldsymbol{\tau} \in \Re^{n\times 1}$ given in (33) is distributed over the on-board thrusters according to:

$$\boldsymbol{\tau} = \mathbf{E}\boldsymbol{p} \tag{49}$$

where $\mathbf{E} \in \Re^{6\times n}$ is the thrusters configuration matrix that captures the geometry of the thruster layout, and $\boldsymbol{p} \in \Re^{n}$ is the collection of individual thrusts. For a given $\boldsymbol{\tau}$, there is an infinite number of $\boldsymbol{p}$. To choose one particular solution, the following l_∞ norm optimization formulation is used:

$$\begin{aligned} &\text{minimize} \quad \|\boldsymbol{p}\|_\infty \\ &\text{subject to} \quad \boldsymbol{\tau} = \mathbf{E}\boldsymbol{p},\ p_{i,l} \le p_i \le p_{i,u},\ i = 1\ldots n \end{aligned} \tag{50}$$

where $\|\boldsymbol{p}\|_\infty = \max\{|p_1| \quad |p_2| \quad \ldots \quad |p_n|\}$, and $p_{i,l}$ and $p_{i,u}$ are the lower and upper thruster saturation limits, respectively. The optimization problem of Eq. (50) can be written in the compact matrix form of:

$$\begin{aligned} &\text{minimize} \quad \boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} \\ &\text{subject to} \quad \mathbf{A}_1\boldsymbol{y} \ge \boldsymbol{b}_1,\ \mathbf{A}_2\boldsymbol{y} = \boldsymbol{b}_2 \end{aligned} \tag{51}$$

where

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{-I} & \mathbf{I}_n \\ \mathbf{I} & \mathbf{I}_n \\ \mathbf{-I} & \mathbf{0}_n \\ \mathbf{I} & \mathbf{0}_n \end{bmatrix} \in \Re^{4n\times(n+1)}, \quad \boldsymbol{b}_1 = \begin{bmatrix} \mathbf{0}_n \\ \mathbf{0}_n \\ \mathbf{-}\boldsymbol{p}_u \\ \boldsymbol{p}_l \end{bmatrix} \in \Re^{4n},$$

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{E} & \mathbf{0}_m \end{bmatrix} \in \Re^{m\times(n+1)}, \quad \boldsymbol{b}_2 = \boldsymbol{\tau} \in \Re^m,$$

$$\boldsymbol{c} = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \end{bmatrix}^{\mathrm{T}} \in \Re^{n+1}, \ \boldsymbol{y} = \begin{bmatrix} \boldsymbol{p} & l \end{bmatrix}^{\mathrm{T}} \in \Re^{n+1}$$

with $\mathbf{I}_n = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^{\mathrm{T}} \in \Re^n$ and $\mathbf{I} \in \Re^{n\times n}$ denoting the identity matrix. The term $\boldsymbol{p}_u = \begin{bmatrix} p_{1,u} & p_{2,u} & \dots & p_{n,u} \end{bmatrix}^{\mathrm{T}} \in \Re^n$ is the upper saturation vector, and the term $\boldsymbol{p}_l = \begin{bmatrix} p_{1,l} & p_{2,l} & \dots & p_{n,l} \end{bmatrix}^{\mathrm{T}} \in \Re^n$ is the lower saturation vector with $\mathbf{0}_n \in \Re^n$ and $\mathbf{0}_m \in \Re^m$ being the zero column vectors. The term l is the l_∞ norm of the thrust manifold, i.e., $l = \|\boldsymbol{p}\|_\infty$. The transition from Eq. (50) to Eq. (51) and a fault-tolerant version of Eq. (51) are given by Soylu *et al*. [14].

5 HMI SOFTWARE ARCHITECTURE

A new HMI is proposed that integrates RSI 6-DOF Joystick, an OpenGL-based, 3-D graphical display, and a new GUI to facilitate complex ROVM missions. In Section 2, the 6-DOF input device, RSI joystick, was presented. In the current section, the remaining components of the HMI alongside with its software structure are presented.

5.1 HMI System Architecture

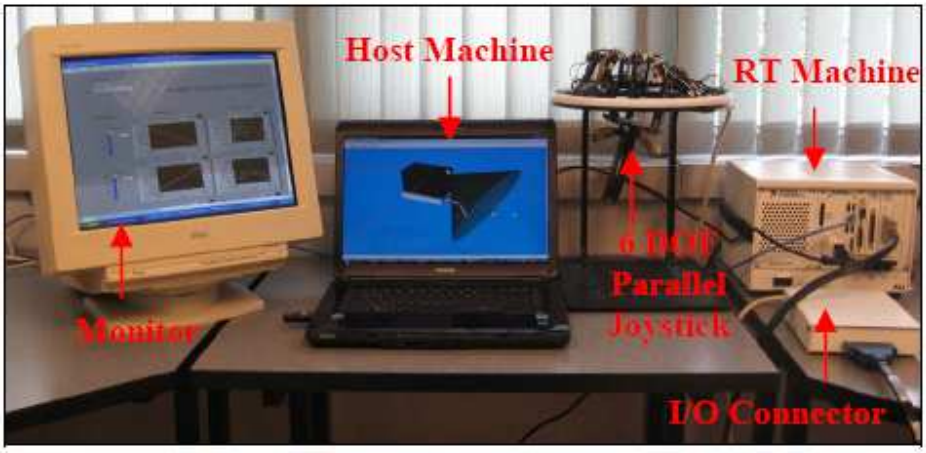


Figure 4: The console of real-time HMI.

The console of the HMI is shown in Figure 4. The console consists of the following hardware components: a Microsoft WINDOWS™ based host computer, a real-time (RT) computer (National Instrument™ PXI 1031) with data acquisition card (National Instrument™ PXI 6251), I/O connector (National Instrument™ SCB 68) a monitor for 3D graphical display of the ROVM, and a parallel architectured 6-DOF joystick produced by RSI Research Ltd.

5.2 HMI Real-Time Prototyping

The real-time implementation of the proposed scheme requires the completion of an operation within a specified time interval between events. To achieve the real time execution of the scheme given in Figure 2, MATLAB™/SIMULINK™ [34], Real-Time Workshop™, LabVIEW™ [35], and LabVIEW Simulation Interface Toolkit (SIT) are used in the current work. The host computer has LabVIEW™, MATLAB™\SIMULINK™ and Real-Time Workshop™ installed on it.

MATLAB™/SIMULINK™ is used to create the customizable blocks as well as to specify the data flow between the blocks involved in the high-level diagram.

To generate a C code from the matlab/simulink model, Real-Time Workshop™ is used. The generated C code is then compiled into dynamic link libraries (DLLs) that are used for the real-time execution of the code. LabVIEW™ is used to create a user interface to the simulink model that allows the human pilot to assess the performance of the system as well as to specify user defined control parameters based on the system performance. As well, the LabVIEW™ controls and indicators provide access to the SIMULINK™ parameters. This way, the pilot can change control parameters on the fly and immediately view the results of changes. This greatly facilities control prototyping and hardware-in-the-loop testing.

The generated labview and matlab/simulink codes were linked together through the SIT that uses TCP/IP communication protocol to exchange data between the two programs. As well, SIT allows the easy deployment of the generated DLLs to the RT engine through the Ethernet connection between the host and RT machine. The RT engine runs on a real-time operating system (RTOS) and ensures that the execution of the scheme adheres to real-time operation requirements. The real time data acquisition is performed the DAQ card that is connected to the RT machine. Finally, the real time visualization block, which was developed in C ++ using Irrlicht libraries [36], receives the output of the simulation using TCP/IP communication protocol and renders the ROV and relevant images in the host machine. The schematic of this hardware/software structure is given in Figure 5.

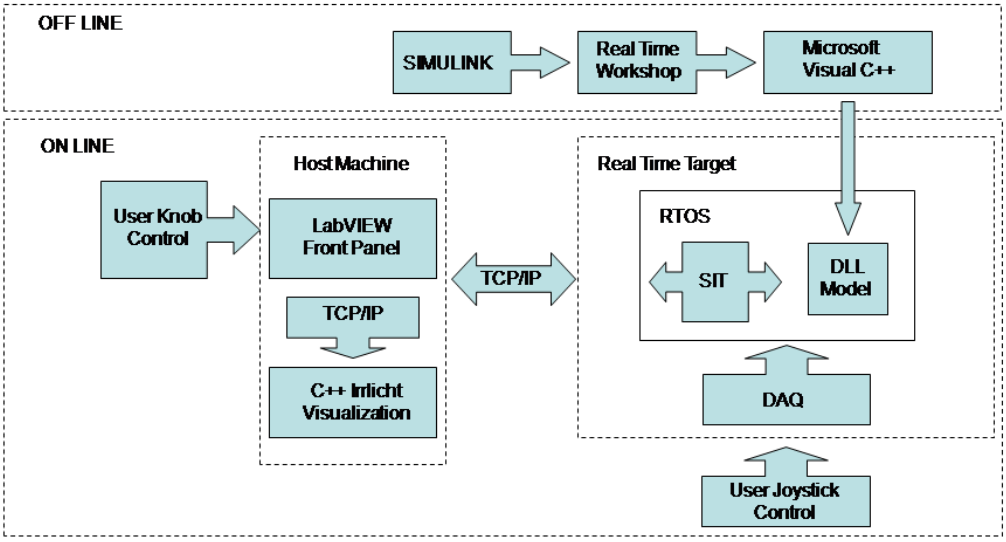


Figure 5: Hardware/Software structure.

5.3 GUI

The GUI component of the HMI allows the user to set the control mode of operation with their respective settings and to monitor the system response to given motion commands. A snapshot of the proposed GUI is shown Figure 6. According to the GUI, the user first selects the mode of operation by clicking the appropriate button on the GUI. The modes of operations available are *Joystick Control*, *Manuel Control* and *Automatic Trajectory Control*.

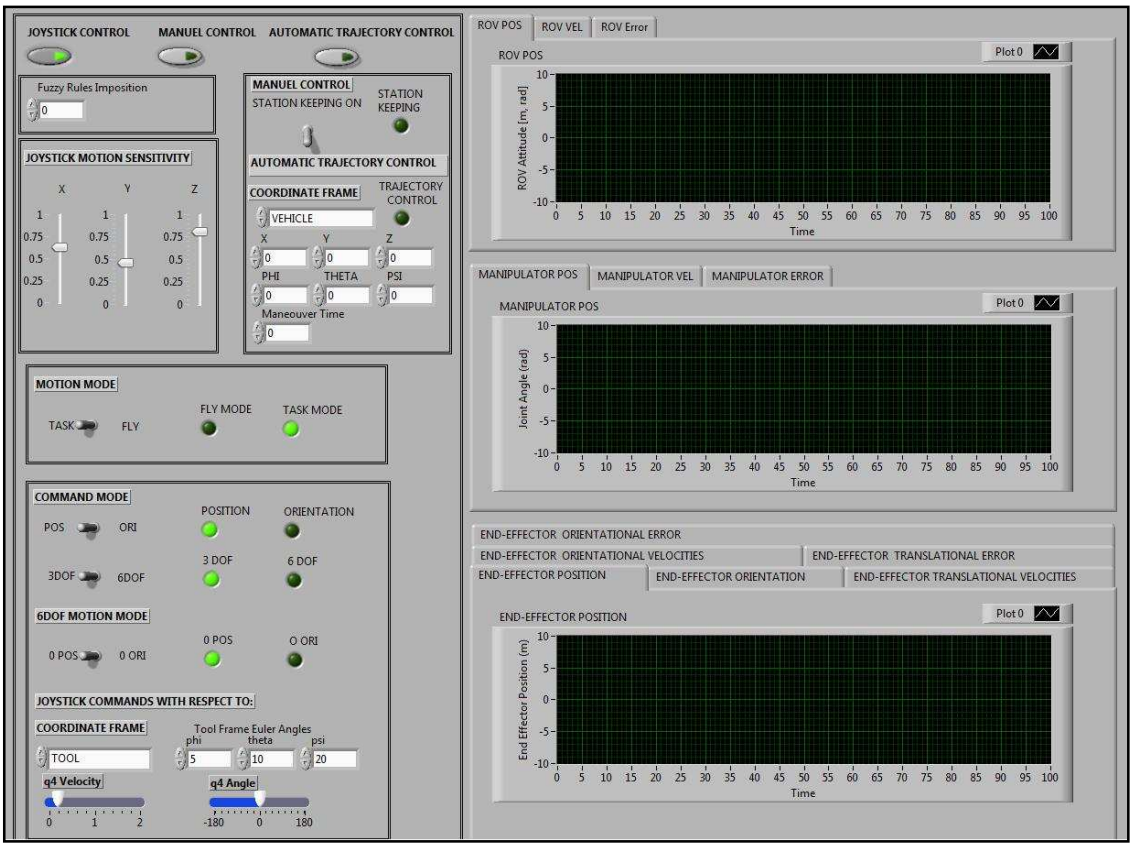


Figure 6: Snapshot of the GUI.

5.3.1 Joystick Control

In the *Joystick mode*, the scheme defined in Figure 2 is employed; that is; the user drives the commercial joystick described in Section 2 to generate desired end-effector velocity values with the visual clues provided by the Open-GL based visualization tool demonstrated in Figure 7. In response to the pilot's input, the redundancy resolution algorithm described in Section 3 distributes the motion over the vehicle and the manipulator and generates reference values for the controller. The controller defined in Section 4.3 generates the forces and moments to be applied to the vehicle and the manipulator to accomplish the reference state values, and finally, the thrust allocation

scheme defined in Section 4.4.1 determines the individual thrust to be demanded from the on-board thrusters.

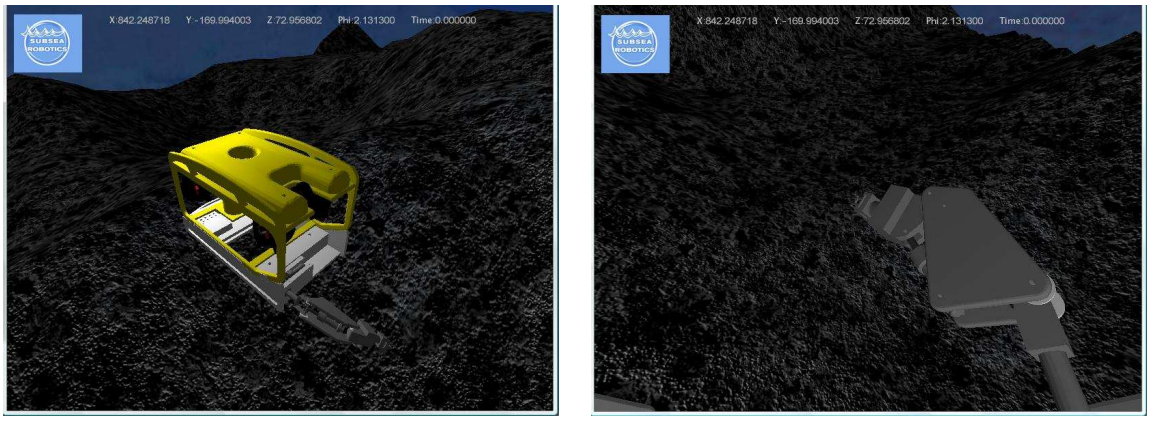


Figure 7: Snapshots from the Visualization tool. The left image is a third party view and the right image is the camera view.

In the *Joystick Mode*, the *Fuzzy Rules Imposition* controls λ in Eq. (10) and allows the pilot to determine how aggressively the fuzzy rules are employed on the fly. Similarly, the *Joystick Motion Sensitivity* controls the parameter $\boldsymbol{k}_i$ in Eq. (6) and hence allows the pilot to specify the sensitivity of the joystick motion in generating the desired end-effector velocities along *X*, *Y* and *Z* axis.

Fly Mode: In the *Joystick Mode*, under *Motion Mode* label on the GUI, there are two operation modes, namely, *Fly Mode* and *Task Mode*. The *Fly Mode* is designed to help the pilot place the end-effector in an intended position and orientation in an effective manner using the proposed redundancy resolution scheme with the options outlined in Table 1. According to this mode, the pilot can control either the end-effector position or orientation alone by switching the toggle to *3DOF* and choosing the type of the motion under the *Command Mode* or the end-effector position and orientation together by

switching the toggle to *6DOF*. When *6DOF* selected, the pilot can keep the end-effector position/orientation stationary while the end-effector orientation/position is commanded by switching the toggle to *POS*/*ORI* under *6DOF Motion Mode*. Depending on the motion mode selected, the definition of Jacobian matrix in the redundancy resolution algorithm of Section 3 changes according to Jacobian row in Table 1.

Under the *Coordinate Frame* label on the GUI, the pilot can set the coordinate frame through which reference velocity values are defined with respect to a particular coordinate frame. The available coordinate frames include *Inertia, Tool, Vehicle*, *Camera* and *End-Effector* coordinate frames for translational motion and *Tool* and *Inertia* coordinate frames for rotational motion with the coordinated frames being defined in Figure 8.

For the translational motion, the definitions of the desired translational end-effector velocities, $\dot{\boldsymbol{x}}_{\boldsymbol{ee}}$, based on the given joystick command, $\dot{\boldsymbol{x}}_{\boldsymbol{p}}$, are given in "EE Command Definition with Respect to" row in Table 1. In this row, the terms ${}^{I}_{T}\mathbf{R}$, ${}^{I}_{V}\mathbf{R}$, ${}^{I}_{C}\mathbf{R}$ and ${}^{I}_{EE}\mathbf{R}$ represent the rotation matrices from the Tool, Vehicle, Camera and End-Effector frames to the inertial reference frame, respectively and are used to map the desired end-effector velocities into the inertial reference frame for use in the redundancy resolution algorithm. For instance, when the camera frame is selected, the algorithm assumes that the pilot is demanding end-effector velocities $\dot{\boldsymbol{x}}_{\boldsymbol{p}}$ with respect to the camera reference frame and multiplies this velocity with ${}^{I}_{C}\mathbf{R}$ to obtain $\dot{\boldsymbol{x}}_{\boldsymbol{ee}}$ and send it to the redundancy resolution algorithm.

For the rotational motion, the pilot can give commands in terms of the *Tool* and *Inertial* reference frame. When *Tool* reference frame is selected, the pilot needs to provide the Euler angles of the rotation matrix between the Tool reference frame and the inertial reference frame, ${}^{I}_{T}\mathbf{R}$, so that the desired rotation matrix can be formed based on ${}^{I}_{ee}\mathbf{R}_{d} = {}^{I}_{T}\mathbf{R}\,{}^{T}_{ee}\mathbf{R}$ where ${}^{T}_{ee}\mathbf{R} = {}^{b}_{h}\mathbf{R}$ that is set by the pilot through the joystick.

Task Mode: The task mode is designed to help the pilot perform a manipulation task with the end-effector with the options outlined in Table 2. In this mode, while the end-effector motion is commanded by the pilot through the GUI, the redundancy resolution algorithm is performed up to the *n-1* joint with the Jacobian definitions given in Jacobian row in Table 2; allowing the automatic reconfiguration to take place in the presence of external underwater disturbances while the n^{th} joint performs the manipulation task. To generate commands for the end-effector, the pilot uses either q_n velocity or q_n angle slide bars. While q_n velocity slide bar sets the desired rotational rate for the end-effector, q_n angle slide bar sets the desired angle values for the end-effector. One example of the usage of this mode would be when the pilot is required to drill a hole into an underwater structure. In this case, the pilot can set the desired rotational rate using the GUI while demanding constant translational velocities along the drill direction using the Joystick with optimal system configuration thanks to the redundancy resolution algorithm.

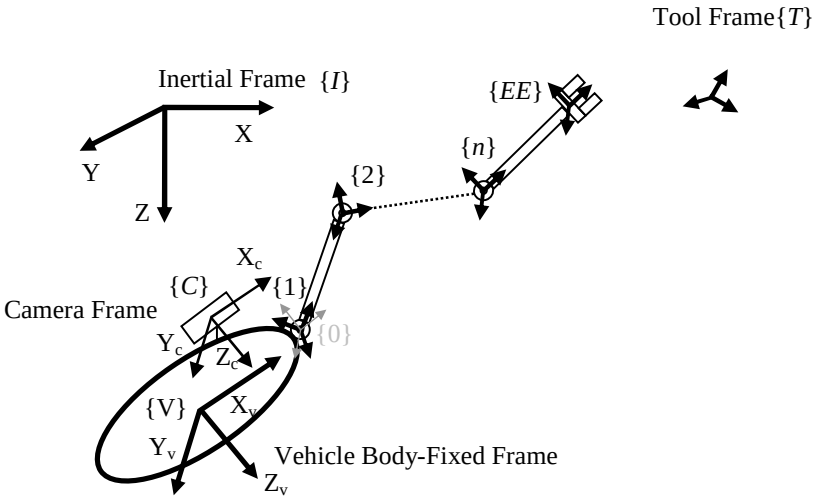


Figure 8: Coordinate frames for a ROVM system for the HMI

Table 1: Fly Mode.

FLY MODE					
MODE		POSITION		ORIENTATION	
		3DOF	6DOF $\dot{\boldsymbol{x}}_o = 0$	3DOF	6DOF $\dot{\boldsymbol{x}}_{ee} = 0$
JACOBIAN		$\mathbf{J}_m = \mathbf{J}(1:3,:)$	$\mathbf{J}_m = \mathbf{J}$	$\mathbf{J}_m = \mathbf{J}(4:6,:)$	$\mathbf{J}_m = \mathbf{J}$
EE Command Definition With Respect To:	Inertial	$\dot{\boldsymbol{x}}_p = 1_{3\times3}\,\dot{\boldsymbol{x}}_{j,p}$		$\dot{\boldsymbol{x}}_o = k\,\mathbf{R}\mathbf{n}_e,\ {}^{I}_{ee}\mathbf{R}_d = {}^{T}_{ee}\mathbf{R}$	
	Tool	$\dot{\boldsymbol{x}}_p = {}^{I}_{T}\mathbf{R}\,\dot{\boldsymbol{x}}_{j,p}$		$\dot{\boldsymbol{x}}_o = k\,\mathbf{R}\mathbf{n}_e,\ {}^{I}_{ee}\mathbf{R}_d = {}^{I}_{T}\mathbf{R}\,{}^{T}_{ee}\mathbf{R}$	
	Vehicle	$\dot{\boldsymbol{x}}_p = {}^{I}_{V}\mathbf{R}\,\dot{\boldsymbol{x}}_{j,p}$			
	Camera	$\dot{\boldsymbol{x}}_p = {}^{I}_{C}\mathbf{R}\,\dot{\boldsymbol{x}}_{j,p}$			
	End-Effector	$\dot{\boldsymbol{x}}_p = {}^{I}_{ee}\mathbf{R}\,\dot{\boldsymbol{x}}_{j,p}$			

Table 2: Task Mode.

TASK MODE

Redundancy Resolution up to the $n-1^{th}$ Joint

MODE	POSITION		ORIENTATION	
	3DOF	6DOF $\dot{x}_O = 0$	3DOF	6DOF $\dot{x}_{ee} = 0$
JACOBIAN	$\mathbf{J}_m = \mathbf{J}(1:3,1:n-1)$	$\mathbf{J}_m = \mathbf{J}(4:6,1:n-1)$	$\mathbf{J}_m = \mathbf{J}(:,1:n-1)$	$\mathbf{J}_m = \mathbf{J}(:,1:n-1)$
EE Command Definition With Respect To:	Same as Table 1			

***n*th Joint Manipulation**

n^{th} Joint	**Manipulation Mode**: q_n is set by the user through the GUI
	Screw Mode: $\dot{q}_n$ is set by the user through the GUI

5.3.2 Automatic Trajectory Control:

In this mode, the pilot first selects the coordinate frame among *Inertia, Tool, Vehicle, Camera* and *End-Effector* coordinate frame options using the pull-down menu under *Automatic Trajectory Control* label and enters the desired positions and orientations into the respective windows on the GUI. The algorithm then generates a continuous set of end-effector values using a third order polynomial function with zero initial and final velocities for the end-effector [24]. These desired values are then fed into the redundancy resolution scheme.

5.3.3 Manual Mode:

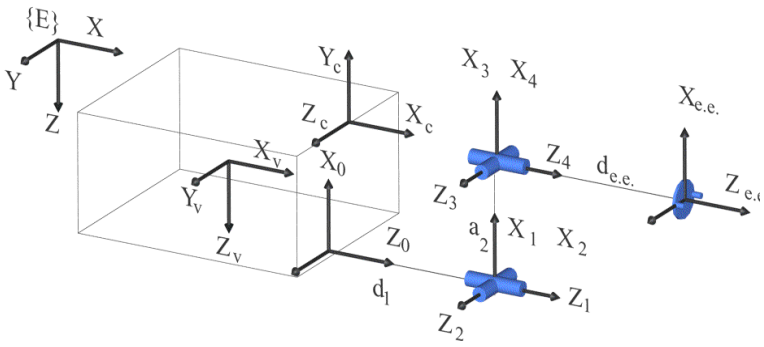
In this mode, the pilot can command the vehicle and manipulator separately using their respective hand control units (**HCU**). In cases where the vehicle is required to stay stationary while the manipulator is performing a task, the pilot can switch on the station keeping mode. This mode keeps the vehicle stationary by mitigating the disturbances such as the dynamic coupling effect caused by the manipulator motion and ocean currents etc. To this end, *Station-Keeping Mode* implements a series of Single-Input-Single Output (**SISO**) sliding-mode controllers as described in [37] to generate desired thrust schedules for the on-board thrusters.

6 SIMULATIONS

6.1 Numerical Simulation Studies

The ROVM system considered in this work is a Saab Seaeye Falcon™ ROV equipped with a 4-DOF Hydrolek™ manipulator. As illustrated in Figure 9a, the inertial frame is labelled as "E", and the body-fixed frame (X_v, Y_v, Z_v) is attached to the center of mass of the ROV. The dynamic parameters of the Falcon™ ROV were taken from [38]. Regarding the manipulator, the body-fixed frames are attached to a base joint of each successive link using the modified Denavit-Hartenberg convention [31] shown in Figure 9b. Each manipulator link is modeled as cylinders with the length of link 1, 2, and 3 being 0.15, 0.22, and 0.15m., respectively. The inertia parameters of the manipulator were obtained from the CAD models and the added mass properties were obtained from the strip theory [27].

a)



b)

i-1	*α-1*	a_{i-1}	d_i	θ_j	*i*
0	0	0	0	q_1	1
1	$-\pi/2$	0	d_1	q_2	2
2	0	d_2	0	q_3	3
3	$\pi/2$	0	d_3	q_4	ee

Figure 9: Zero displacement of the Falcon™- Hydrolek™ ROVM system

The system is assumed to be neutrally buoyant. In the current simulation study, the thruster dynamic response is assumed to be much faster than that of the ROV itself, and thus was neglected. Regarding the environment, the surrounding fluid is assumed to be motionless. During the simulation, roll and pitch motion of the ROV are not controlled since the on-board thrusters allow only lateral and vertical motions. The sampling frequency for the simulation was set to 100 Hz.

To investigate the performance of the proposed controller a series of waypoints were set using the “automatic trajectory command” mode of the proposed HMI. Again this command generates a continuous set of end-effector values using a third order polynomial function with zero initial and final velocities The system was commanded to move from the initial configuration of $\mathbf{q} = [0 \quad 0 \quad 0; \quad 0 \quad 0 \quad 0; \quad 270 \quad -10 \quad 100 \quad 0]^T$

m; rad; deg. that corresponds to the end-effector position of $\mathbf{x} = [0.8 \quad -0.152 \quad 0.315]^T$ m. to the final configuration of $\mathbf{q} = [5 \quad 1 \quad 2; \quad 0 \quad 0 \quad 0; \quad 220 \quad -70 \quad 120 \quad 0]^T$ m; rad; deg that corresponds to the end-effector location of $\mathbf{x} = [6.11 \quad 1.16 \quad 2.35]^T$ m. The manoeuvre time was set to be 60sec. To reflect uncertainties that are present in the underwater vehicle paradigm, up to 40% modelling inaccuracies were incorporated into the Falcon™ dynamic model used in the controller.

The snapshots of the ROVM manoeuvre were demonstrated in Figure 10. Time history of the vehicle position and orientation and the manipulator joint positions are shown in Figure 11-Figure 13, respectively. In light of Figure 14-Figure 16, it can be concluded that the proposed controller provides excellent performance in the trajectory following task. In regards to the end-effector position error, it is shown in Figure 17. As with the position error, the proposed controller provides high performance in following the desired end-effector trajectory.

The obtained results were also compared to those of the computed-torque method of [24] in which the control signal is generated as $\boldsymbol{\tau} = \hat{\mathbf{M}}_q \left(\ddot{\boldsymbol{q}}_d + \mathbf{K}_v \left(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}} \right) + \mathbf{K}_c \left(\boldsymbol{q}_d - \boldsymbol{q} \right) \right) + \hat{\boldsymbol{h}}_q \left(\boldsymbol{q}, \dot{\boldsymbol{q}} \right)$ with $\mathbf{K}_v$ and $\mathbf{K}_c$ being the positive gain matrices with the same dynamic discrepancies. Both controller's parameters were tuned using MATLAB™ Optimization Tool for step input with the desired controller performance of Rise *Time = 1sec, Settling Time =2.5sec, Overshoot =10%, Undershoot =1%*. The results of the comparison test are illustrated in Figure 14-Figure 17. These

results indicate that the proposed controller outperforms the computed-torque control in the trajectory following performance.

a) b)

c) d)

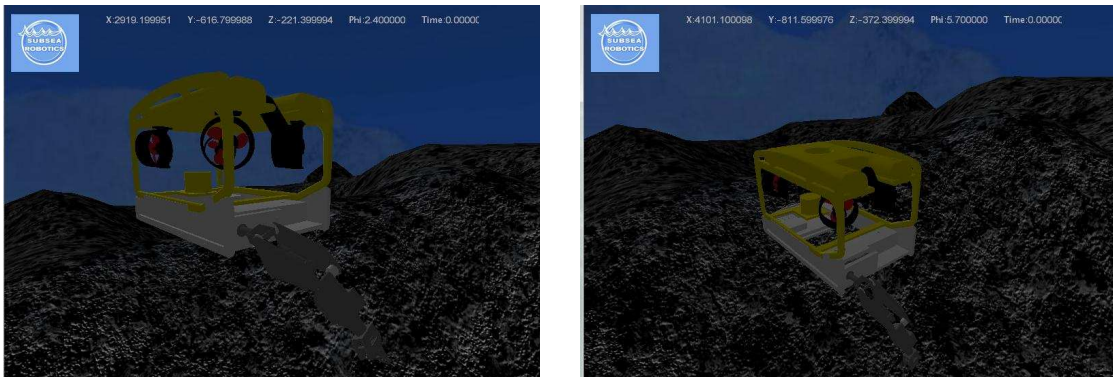


Figure 10: Snapshots of the ROVM maneouvre a) t = 0s, b) t = 20sec, c) t = 40sec, d) t = 60sec.

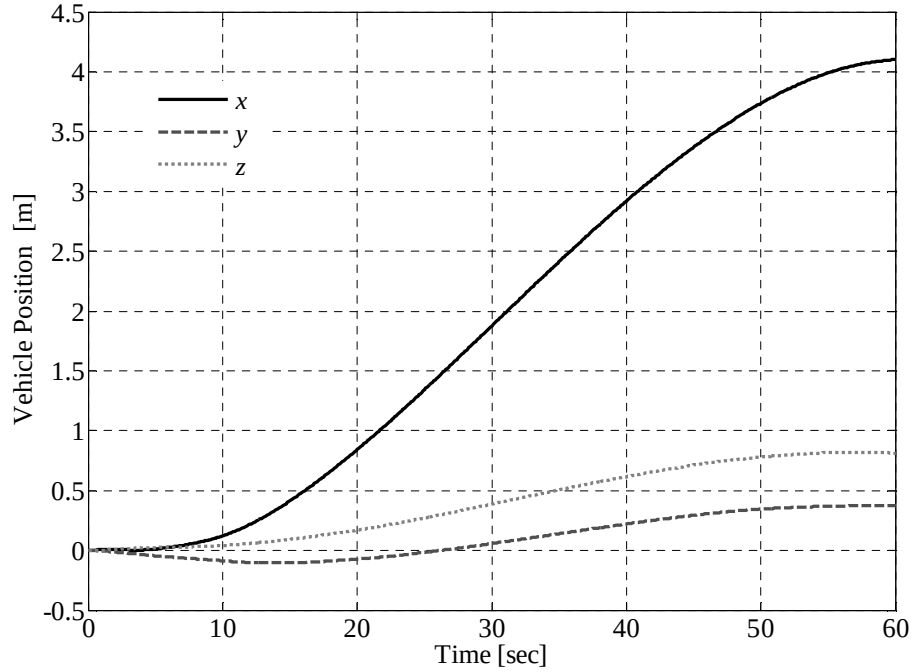


Figure 11: Time history of vehicle translational motion.

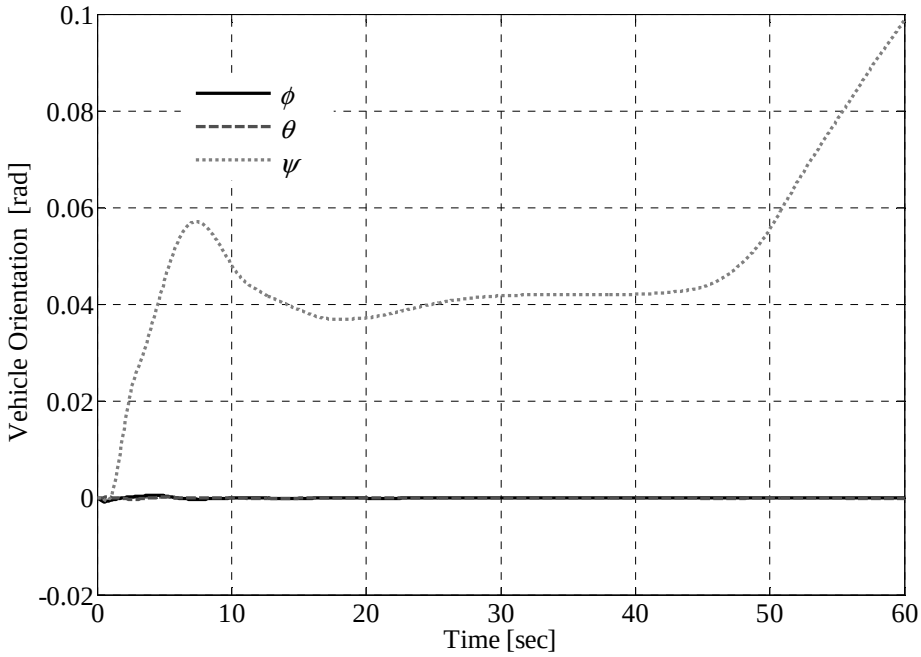


Figure 12: Time history of vehicle orientational motion.

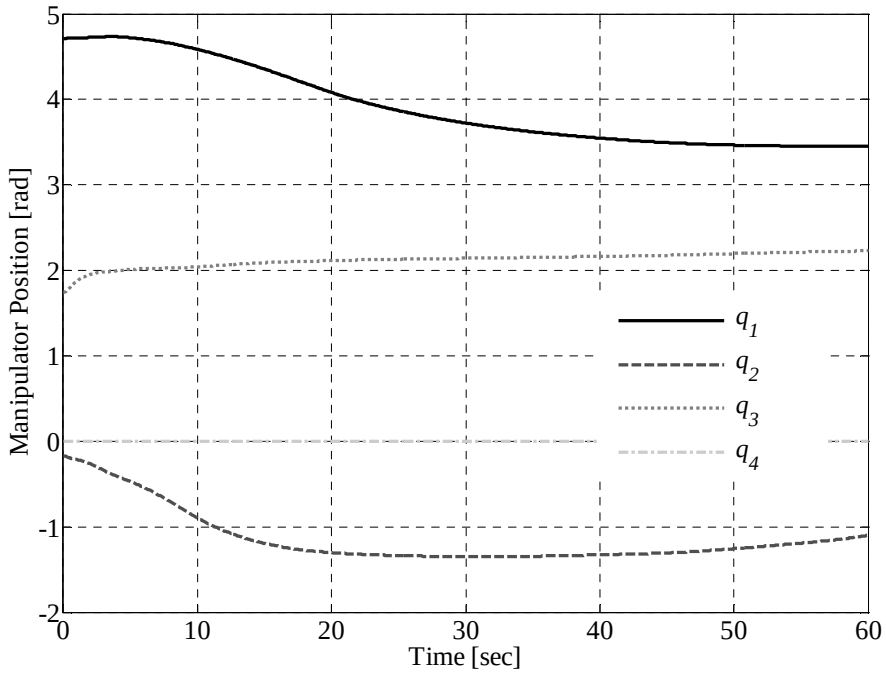


Figure 13: Time history of manipulator joint motion.

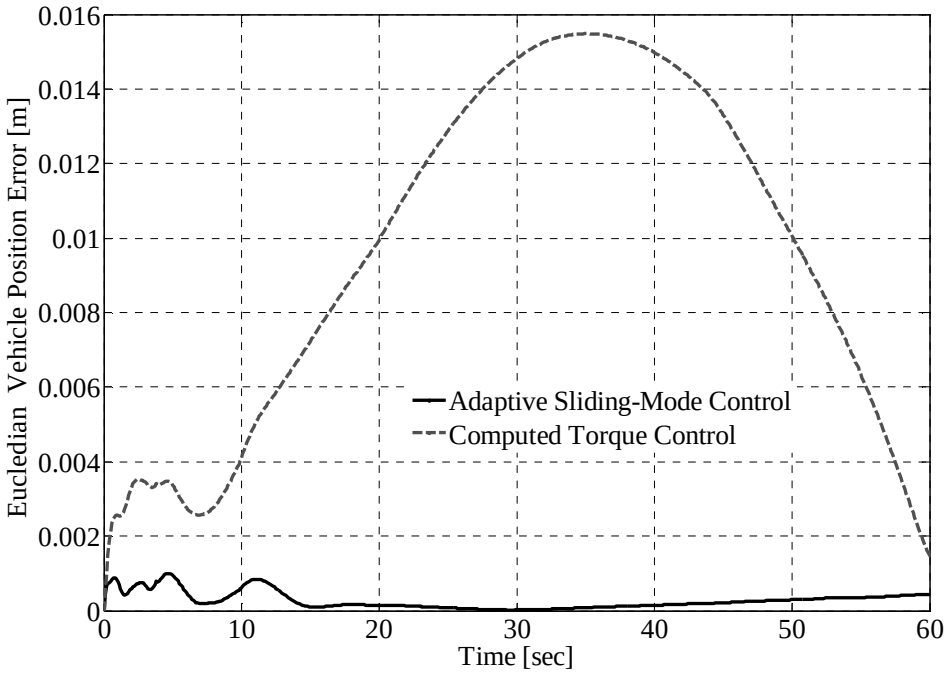


Figure 14: Time history of Euclidian vehicle position error.

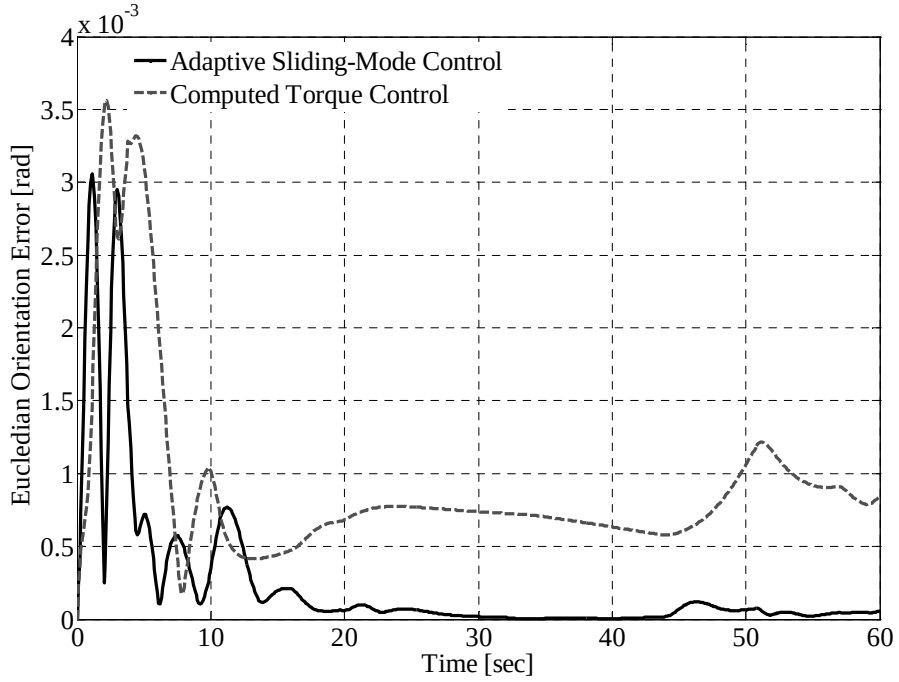


Figure 15: Time history of Euclidian vehicle orientation error.

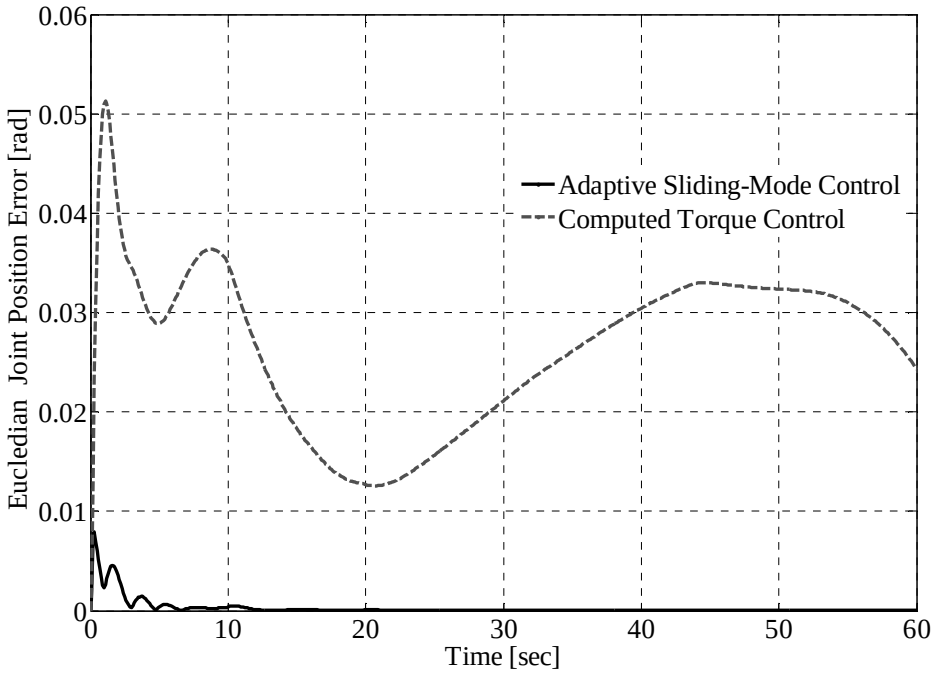


Figure 16: Time history of Euclidian manipulator position error.

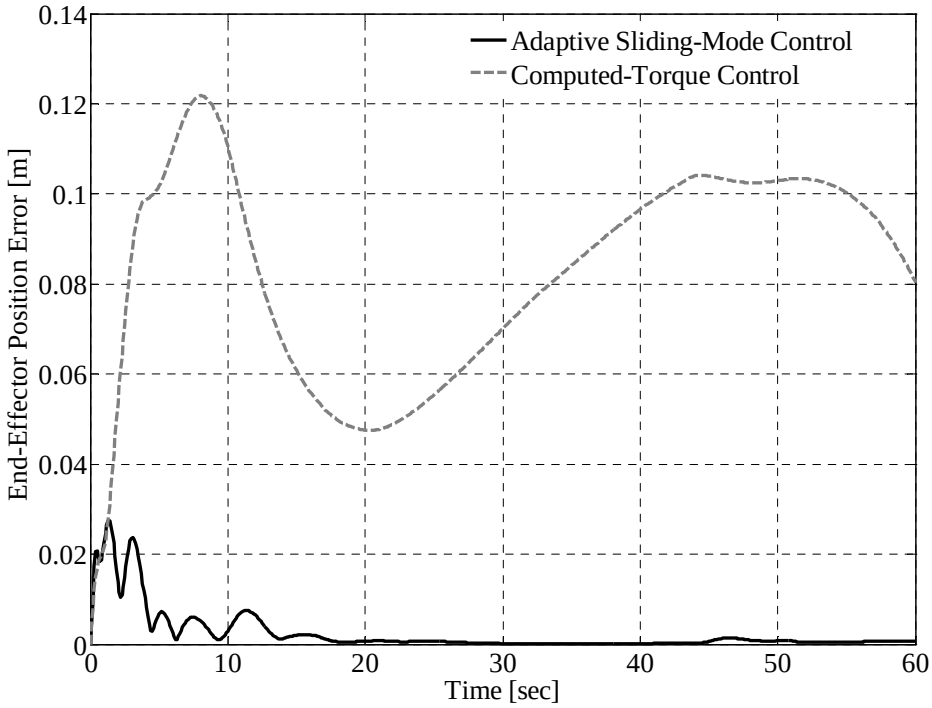


Figure 17: Time history of end-effector position error.

The time history of the fuzzy weighting factors, $\alpha_j \; j=1,2,3,v$, which defines the relative importance of each secondary objective with respect to the others, are demonstrated in Figure 18. In the very beginning of the manoeuvre, the system is close to being in singular configuration, and hence α_1 is high. As the system moves out of the singular configuration, α_1 decreases, and the priority gradually shifts from the singularity avoidance to yaw alignment. Note that during this reorganization of the secondary objective hierarchy, the camera angle objective is not prioritized despite its higher status in the hierarchy with respect to the camera angle objective. This is due to the fact the coordinated camera motion alone managed to keep the end-effector inside camera view, leading to the direct shift to the yaw alignment objective. Between 10 seconds and 50 seconds, the highest priority is placed on the singularity, as the arm approaches to

singular configurations. As the manipulator moves out of the close-singular configurations, the priority shits to the yaw alignment. Finally towards the end of the manoeuvre, the higher priority secondary objectives are achieved with success, and hence the lowest-level minimal vehicle motion index becomes low, also indication high joint-limit avoidance performance.

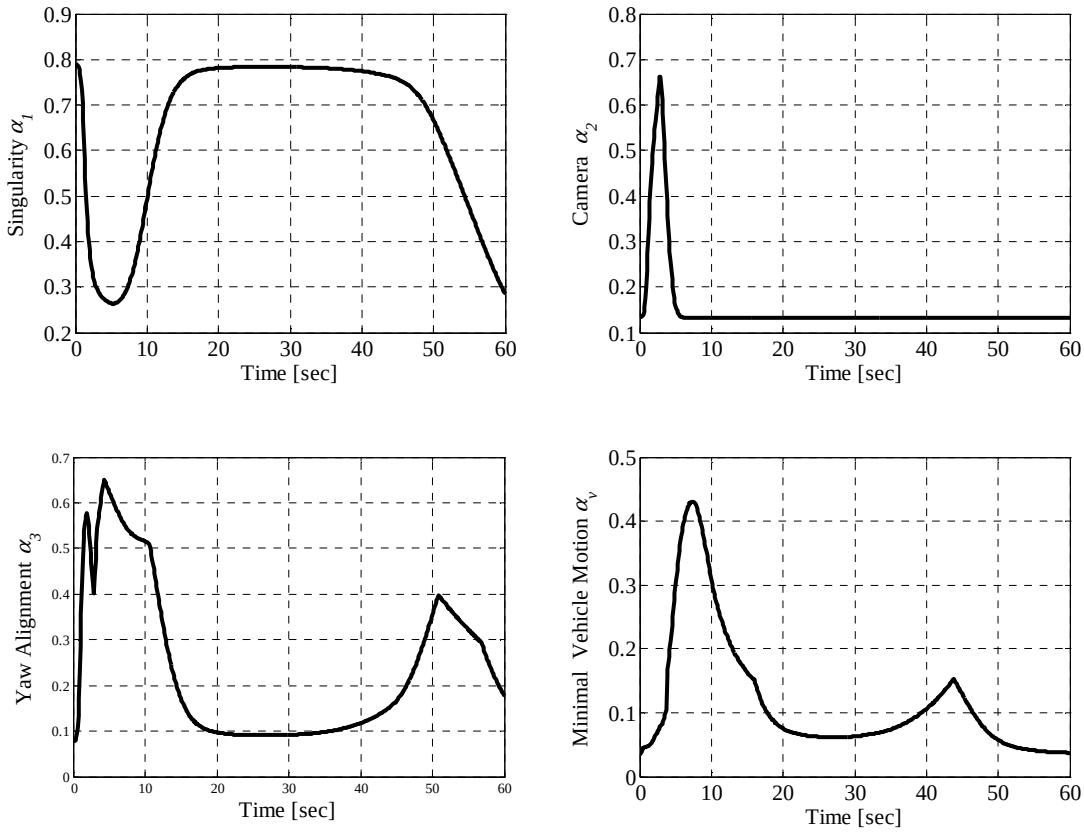


Figure 18: Time history of fuzzy parameters.

The time history of the adaptive upper bound term is shown in Figure 19. As can be seen in the figure, the adaptive upper bound term changes smoothly as the degree of mismatch between the known and the unknown dynamics varies; allowing *Assumption 1* of Section 4.2, i.e., $\hat{\rho} \geq \left\| \tilde{f} \right\|_2$, for the Lyapunov stability to hold. In regards to the

adaptive PID terms, the on-line adaptation law of Eq. (29) yields smooth PID parameters as demonstrated in Figure 20. Note that the PID parameters reported herein are only related to the vehicle's surge motion and similar smooth patterns were obtained for the remaining DOFs.

Figure 21 demonstrates the l_∞ norm thruster allocation based on the optimization problem of Eq. (50). The l2 norm thrust allocation was also performed, which is based on the pseudo-inverse of the thruster configuration matrix. The result shows that the largest thrust value obtained from the l_2 norm minimization and l_∞ norm minimization are 121 and 105 N, respectively. Therefore, l_∞ norm minimization reduces the largest thrust by 12.5% in comparison to the one obtained from the l_2 norm minimization. For this example, the average improvement over the whole manoeuvre is found to be 3.8%. The results are given in Table 3. Note that higher improvement will be achieved when the vehicle is commanded to transverse and rotate at the same time with higher rates, as when the vehicle is commanded to move or rotate along/around one direction at a time, the l_2 and l_∞ norm solutions are identical.

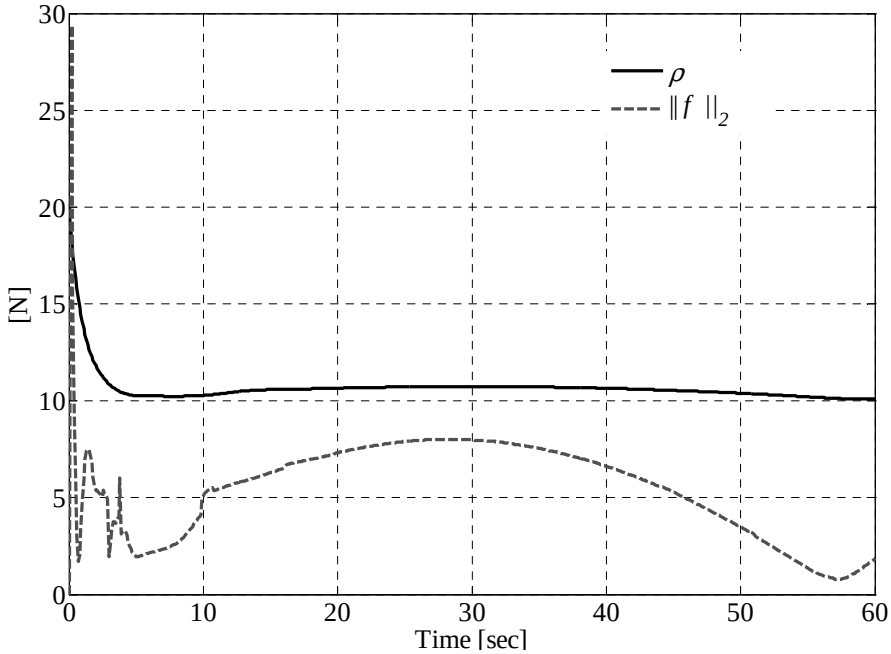


Figure 19: Time history of $\hat{\rho}$ and $\left\| \tilde{f} \right\|_2$

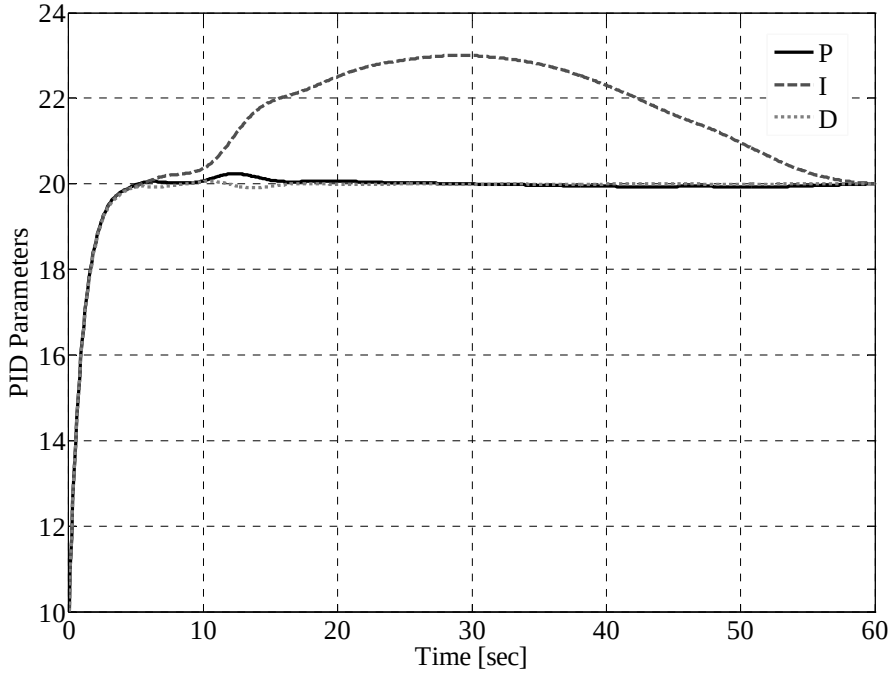


Figure 20: Time history of PID parameters for the x motion of the vehicle.

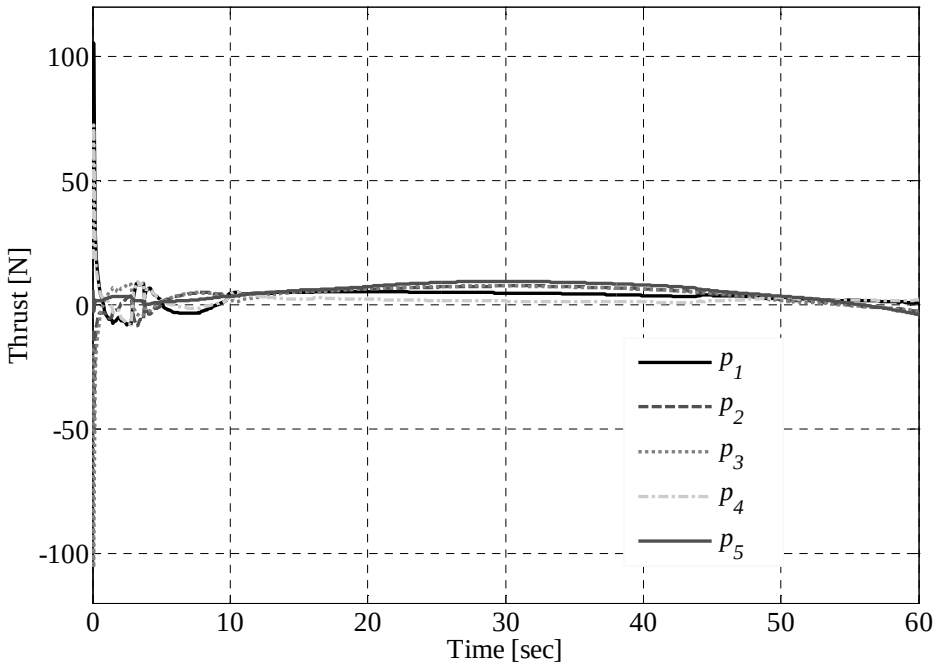


Figure 21: Time history of l_∞ norm thrust values.

Table 3: Comparison of the norms.

Max l_2 Norm Thrust	Max l_∞ Norm Thrust	Reduction	Average Reduction
121N	105N	12.5%	3.8%

6.2 Hardware-in-the-loop Simulations

In this section, the RSI 6DOF joystick was interfaced with the real-time simulation code and a hardware-in-the-loop simulation was performed to demonstrate the use of the proposed human-machine interface.

In the simulation, the HMI of Section 5 was utilized to accomplish a task. In this task, the pilot sets the orientation of the end-effector with respect to the tool-frame by guiding

the RSI joystick while keeping the end-effector position constant during the first 36 seconds of the manoeuvre, requiring 6-DOF end-effector motion control. To this end, the pilot turns on the *Joystick Mode* on the GUI and switches the relevant toggles to *Fly Mode* under *Motion Mode* and *0-POS* under the *6DOF Motion Mode* on the GUI, respectively.

For the remaining section of the manoeuvre, the pilot stops controlling the orientation of the end-effector and starts to command the end-effector translational motion while keeping the orientation constant, causing a stop-and-go type sudden motion. To this end, the pilot switches the toggle to *0-ROT* under the *6DOF Motion Mode* and then drives the joystick such that the end-effector moves along the x and y axis of the tool-frame (taken to be the same as the inertial reference frame), respectively.

The commanded translational end-effector velocities and the end-effector orientations via the RSI joystick during the manoeuvre are demonstrated in Figure 21 and Figure 22, respectively. The corresponding end-effector position and orientation error values in response to the pilot`s commands are illustrated in Figure 23 and Figure 24, respectively. Similarly, the associated state error values are shown in Figure 25. Small error values indicate that the complex ROVM task that requires 6-FOD end-effector motion was successfully completed with the proposed HMI with stable controller behaviour as the results suggest.

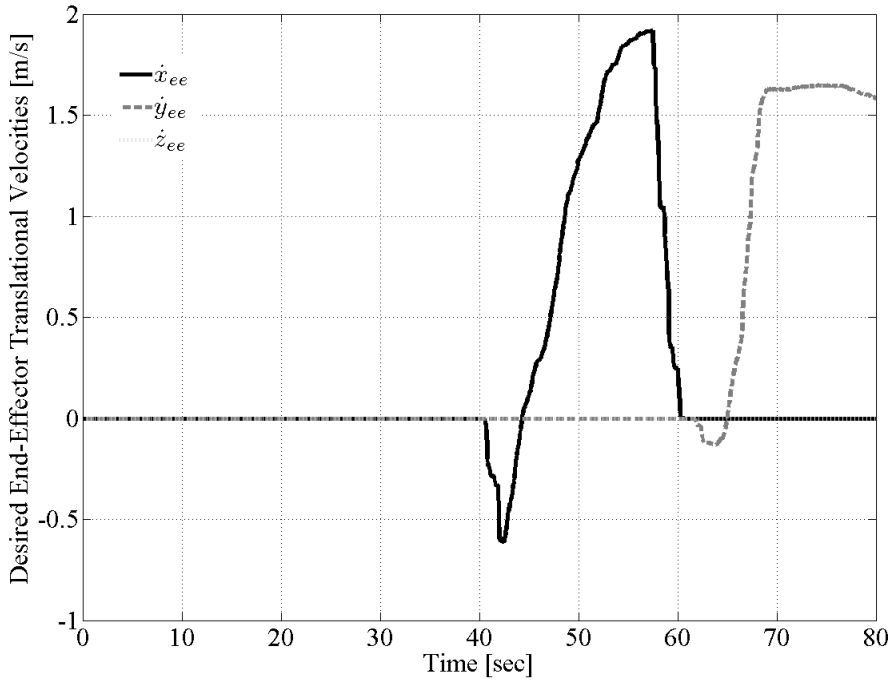


Figure 22: Commanded translational end-effector velocities by the human pilot via the RSI joystick.

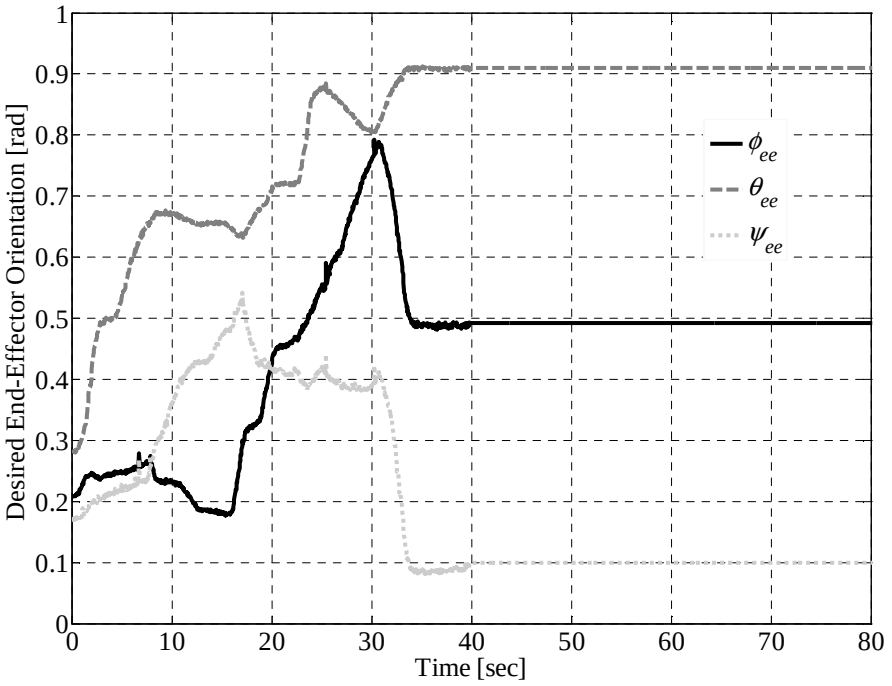


Figure 23: Commanded end-effector orientations by the human pilot via the RSI joystick.

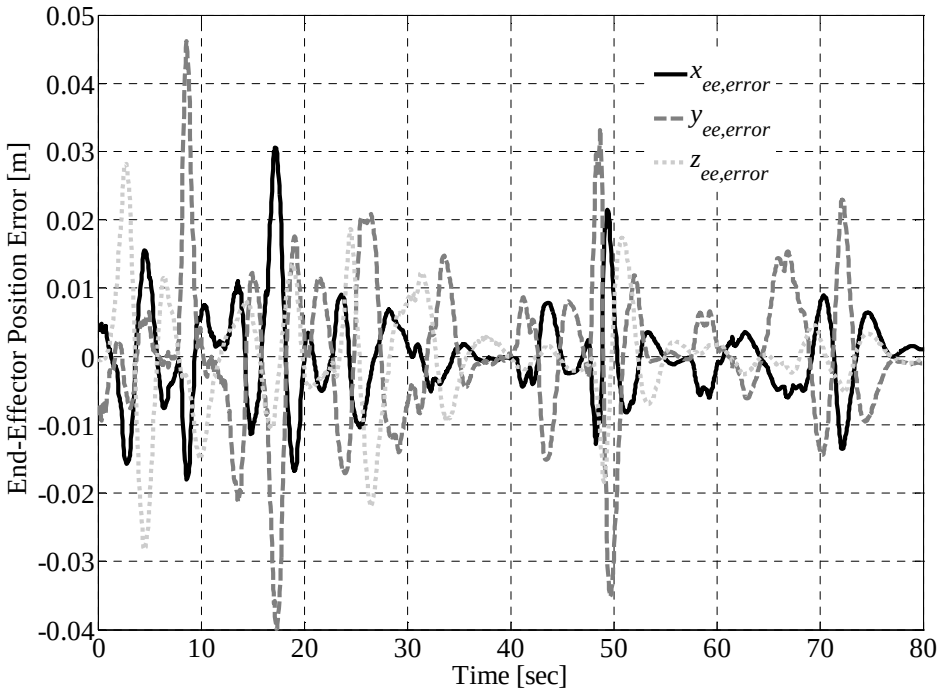


Figure 24: End-Effector translational error values with respect to the inertial reference frame.

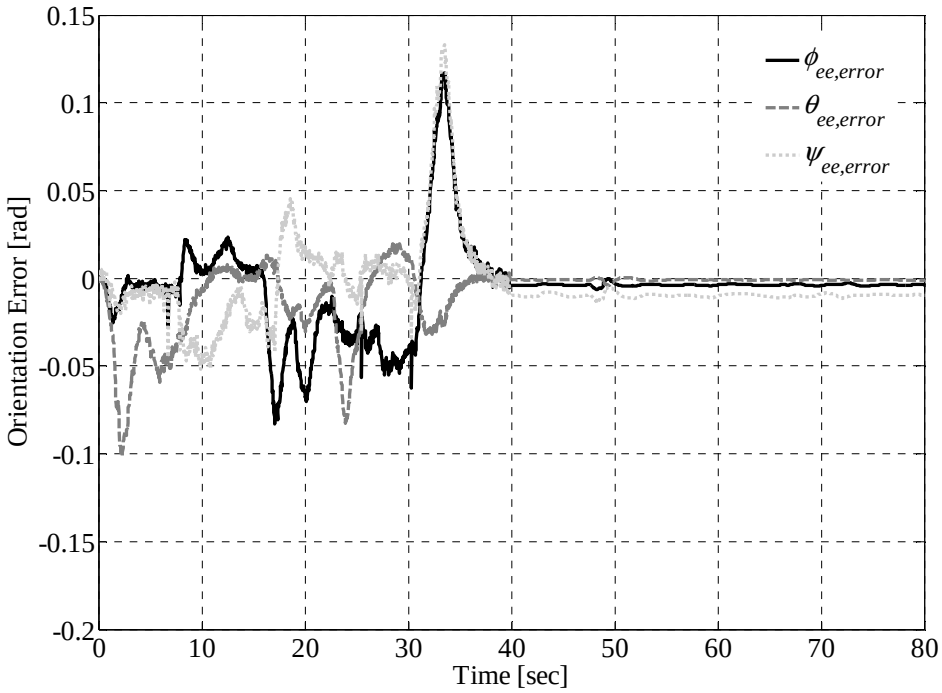


Figure 25: End-Effector orientation (Euler Angles) error values with respect to the inertial reference frame.

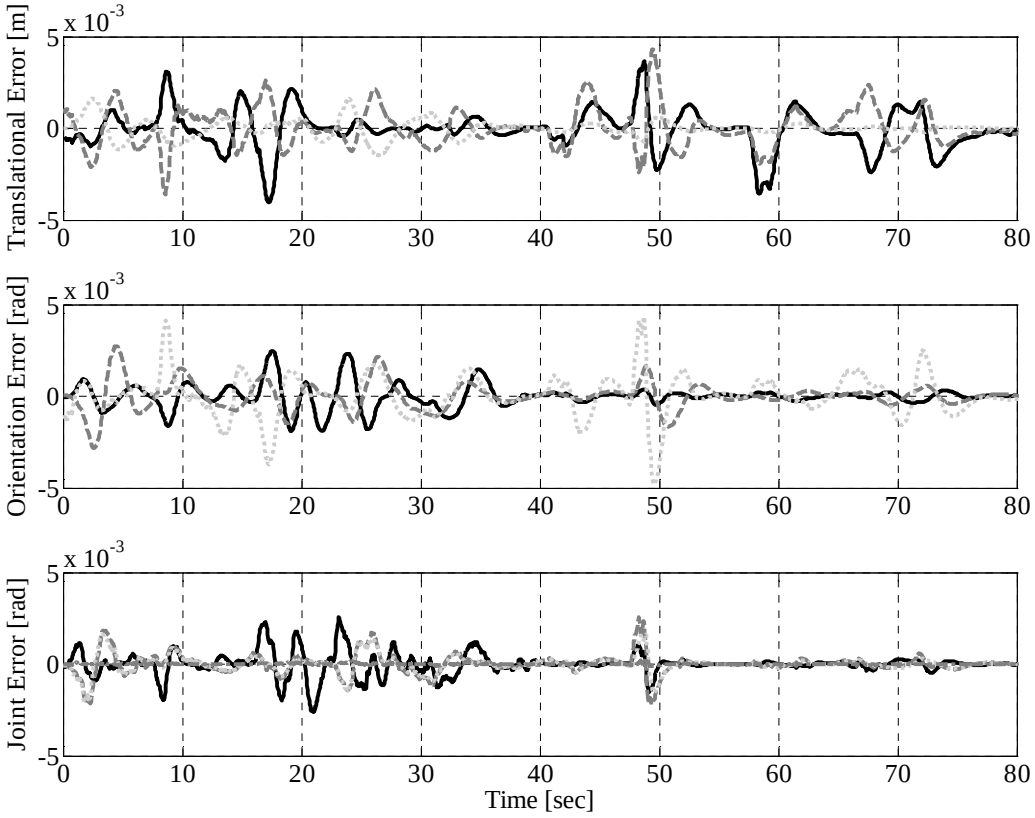


Figure 26: State error values for the vehicle position (top), vehicle orientation (middle) and joint positions (bottom).

7 CONCLUSION

A new HMI has been proposed that consists of the RSI 6-DOF joystick, an OpenGL based 3-D graphical display, and a new GUI to facilitate complex ROVM missions. In the proposed HMI, the human pilot drives the RSI 6-DOF joystick to set the desired the end-effector motion. The desired end-effector motion is then dissected into separate ROV and the manipulator motion in an optimal, coordinated manner by the redundancy resolution algorithm. While generating reference state values in response to the human-pilot input, the redundancy resolution algorithm utilizes the redundant DOFs to accomplish secondary tasks. Once generated, the reference state values are fed into the

controller that is in charge of realizing the desired states by commanding the actuators. To this end, a novel controller has been derived. The controller has two adaptive layers: First layer is for the PID gains whereas the second layer is for the upper bound on the lumped uncertainty vector in lieu of the lumped uncertainty vector itself. The second layer of adaptation relaxes the Lyapunov stability requirements, which in turn leads to a more robust controller in comparison to controllers in which the adaptation takes place to estimate the unknown vector. In addition, methodology is robust to the parameter drift problem and thus offers stable adaptation behaviour when the persistency of excitation condition is not satisfied. The proposed HMI also contains a 3-D visualization tool that emulates the camera feedback that would exist in actual ROVM operation to assist the pilot. As for the GUI, it offers a set of control knobs and monitoring interfaces to facilitate the interaction between the human pilot and a variety of ROVM operation-modes needed for complex ROVM missions requiring extensive end-effector motion capacity. Furthermore, the new GUI also allows the pilot to operate the ROVM system using the conventional one-joint-at-a-time with the optional station-keeping mode. Finally, it has been demonstrated that detailed spatial end-effector manoeuvres that are otherwise not possible can be completed in real-time using the proposed scheme.

Acknowledgment

The authors would like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for providing financial support for this work. Also the authors wish to extend their thanks to Suboceanic Sciences Canada Ltd. in Duncan, BC for their participation and support in the research.

References

[1] S. Soylu, B. J. Buckham, R. P. Podhorodeski, “Redundancy resolution for mobile manipulators,” Ocean Engineering, vol. 37, 2010, pp 325-343.

[2] G. Antonelli, and S. Chiaverini, “Task priority redundancy resolution for underwater-manipulator systems,” In Proceedings of the IEEE Int. Conference on Robotics and Automation, Leuven, Belgium, pp. 768-773, 1998.

[3] N. Sarkar and T. K. Podder, "Coordinated motion planning and control of underwater vehicle-manipulator systems subject to drag optimization," IEEE Journal of Oceanic Engineering, vol. 26, no. 2, 2001, pp. 228–239.

[4] T.K. Podder and N. Sarkar, “Unified dynamics-based motion planning algorithm for autonomous underwater vehicle-manipulator systems,” Robotica, 22, pp. 117-128, 2004.

[5] J. Han and W.K. Chung, “Redundancy resolution for underwater vehicle-manipulator systems with minimizing restoring moments,” In Proceedings of the 2007 IEEE/RSJ Int. Conference on Intelligent Robotics and Systems, San Diego, CA, USA, pp. 3522-3527, 2007.

[6] G. Antonelli, and S. Chiaverini, “Fuzzy Redundancy resolution and motion coordination for underwater vehicle-manipulator systems. IEEE Transactions on Fuzzy Systems, vol. 11, no. 1, pp. 109-120, 2003.

[7] A. Liegeois, “Automatic supervisory control of the configuration and behaviour of multibody mechanisms,” IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-7, 1977, pp. 868-871.

[8] D. Driankov, H. Hellendoorn, and M. Reinfrank, An Introduction to Fuzzy Control. New York, NY: Springer-Verlag, 1995.

[9] G. Antonelli, and S. Chiaverini, “Adaptive tracking control of underwater vehicle-manipulator systems,” In Proceedings of Conference on Control Applications, Trieste, Italy, pp. 1089-1093, 1998.

[10] G. Antonelli, F. Caccavale, and S. Chiaverini, “Adaptive tracking control of underwater vehicle-manipulator systems based on the virtual decomposition approach,” IEEE Transactions on Robotics and Automation, vol. 20, no. 3, 2004, pp. 594–602.

[11] De Wit C. Canuscas, Diaz D. Olgun and M. Perrier, “Nonlinear control of an underwater vehicle/manipulator system with composite dynamics,” IEEE Transactions on Control System Technology, vol. 8, pp. 948-960, 2000.

[12] G. Antonelli, Underwater Robots: Motion and force control of vehicle-manipulator systems, Springer-Verlag, 2003.

[13] B. Xu, S. R. Pandian, and F. Petry, “A sliding-mode fuzzy controller for underwater vehiclemanipulator systems,” In Proceedings of 2005 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS’ 2005), Ann Arbor, MI., 2005.

[14] S. Soylu, B.J. Buckham and R.P. Podhorodeski, “A chattering-free sliding-mode controller for underwater vehicles with fault tolerant infinity norm thrust allocation” Ocean Engineering, vol. 36, no. 16, pp. 1647-1659, 2008.

[15] S. Soylu, B. J. Buckham and R. P. Podhorodeski, “Exploiting Redundancy in Underwater-Manipulator Systems,” International Journal of Offshore and Polar Engineering, vol. 19, no. 2, 2009, pp 115-123.

[16] F. L. Lewis, K. Liu, and A. Yesildirek, “Neural Net Controller with Guaranteed Tracking Performance," IEEE Transactions on Neural Networks,” vol. 6, no. 3, pp. 703-715, 1995.

[17] L. Meirovitch, Dynamics and Control of Structures, John Willey&Sons, 1990.

[18] L. Meirovitch, and T. Stemple, "Hybrid equation of motion for flexible multibody systems using quasicoordinates," Journal of Guidance, Control and Dynamics, vol. 18, no. 4, pp 678-688, 1995.

[19] P. A. Ioannou and P. V. Kokotovic. Adaptive Systems with Reduced Models. New York: Springer-Verlag, 1983.

[20] J. M. Hollerbach and D. M. Lokhorst, “Closed-Loop kinematic calibration of the RSI 6-DOF hand controller,” IEEE Transactions on Robotics and Automation, vol. 11, no. 3, pp. 352-359, 1995.

[21] L. Notash, and R. P. Podhorodeski, “Complete forward displacement solutions for a class of three-branch parallel manipulators,” Journal of Robotic Systems, vol. 11, pp 471-485, 1994.

[22] P. Sobejko, Implementation of Redundantly-Sensed Parallel-Manipulator-Based 6-DOF Joysticks. M.A.Sc. Thesis, University of Victoria, Department of Mechanical Engineering, December 2002.

[23] D. Lim, and H. Seraji, “Configuration control of a mobile dexterous robot: Real-time implementation and experimentation,” International Journal of Robotics Research. vol. 16, pp. 601-618, 1997.

[24] J. J. Craig, Introduction to Robotics, 2nd Edition. Addison-Wesley, 1989.

[25] D. S. Watkins, Fundamentals of Matrix Computations, 2nd Edition, New York, NY: John Wiley & Sons, 2002.

[26] J. Kim, and W. K. Chung, "Dynamic analysis and two-time scale control for underwater vehicle-manipulator systems," in Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), vol. 1, pp. 577–582, October 2003.

[27] T. Fossen, Guidance and Control of Ocean Vehicles, New York NY: John Wiley & Sons, 1994.

[28] F. J. Lin, P. H. Shen, and S. P. Hsu, “Adaptive backstepping sliding-mode control for linear induction motor drive,” IEE Proc of Electric Power Applications, vol. 149, no. 3, pp. 184-194, 2002.

[29] F. J. Lin, and R. J. Wai, “Robust control using neural network uncertainty observer for linear induction motor servo drive,” IEEE Transactions on Power Electronics, vol. 17, no 2, pp 241-254, 2002.

[30] J. J. Slotine, and W. Li, Applied Nonlinear Control. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[31] S. Soylu, B. J. Buckham and R. P. Podhorodeski, “Development of a coordinated controller for underwater vehicle-manipulator systems,” in Proceedings of OCEANS 2008 MTS/IEEE, Quebec City, QC, Canada, September 2008.

[32] F. Sun, Z. Sun, Y. Zhu and W. Lu, “Stable Neuro-adaptive control for robots with the upper bound estimation on the neural approximation errors,” Journal of Intelligent and Robotic Systems,” vol. 26, pp. 91-100, 1999.

[33] F. L. Lewis, K. Liu, and A. Yesildirek, “Neural-net controller with guaranteed tracking performance,” IEEE Trans. Neural Networks, vol. 6, no. 3, pp. 703–715, 1995.

[34] H. Dabney, T. Harman, Mastering SIMULINK, Upper Saddle River, NJ: Prentice Hall, 2004.

[35] R. H. Bishop, Learning with LabVIEW 7 Express, Upper Saddle River, NJ: Prentice Hall, 2004.

[36] www.irrlicht.sourceforce.net.

[37] Soylu, B. J. Buckham, and R. P. Podhorodeski, “Dynamics and control of tethered underwater-manipulator systems,” in Proceedings of OCEANS 2010 MTS/IEEE, Seattle, Washington, USA, 2010.

[38] J. Zand, “Enhanced Navigation and Tether Management of Inspection Class ROVs,” Master’s. Thesis, University of Victoria, 2010.