

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI[®]

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Motion Planning for Flexible Manipulators

by

Christopher Burke Pond
B.Math., University of Waterloo, 1991
M.A.Sc., University of Victoria, 1993

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

DOCTOR OF PHILOSOPHY

in the Department of Mechanical Engineering.

We accept this dissertation as conforming
to the required standard

Dr. I. Sharf, Supervisor (Dept. of Mechanical Engineering)

Dr. A. Dóige, Department Member (Dept. of Mechanical Engineering)

Dr. R. Podhórdeski, Department Member (Dept. of Mechanical Engineering)

Dr. B. Tabarrok, Department Member (Dept. of Mechanical Engineering)

Dr. P. Agathoklis, Outside Member (Dept. of Electrical and Computer Engineering)

Dr. K. Gupta, External Examiner (Simon Fraser University)

© CHRISTOPHER BURKE POND, 1999
University of Victoria

All rights reserved. This dissertation may not be reproduced in whole or in part, by photocopy or other means, without permission of the author.

Supervisor: Dr. Inna Sharf

Abstract

As robotic manipulators become more prevalent, particularly in hazardous environments or for repetitive tasks, demand continues for increased performance and decreased cost. In some applications, both can be achieved by reducing the weight of the manipulator. However, reduced weight often leads to significant structural flexibility and vibration which, for most tasks, is generally regarded as detrimental to performance.

Although there has been a great deal of research in the area of controlling flexible manipulators to follow a desired trajectory, much less work has been directed towards choosing the trajectory itself. The objective of this work is to optimize point-to-point motions in joint space to reduce vibration. This problem is formulated as one of functional optimization and the applicable methods of solution are reviewed. An indirect method is chosen that allows modular software development by preserving the integrity of existing nonlinear dynamics models. Numerical results are compared with trajectories generated by other means and show a significant reduction in vibration possible by optimization, particularly for varying joint paths.

Finally, the effectiveness of the trajectory optimization scheme is further evaluated for high-speed, large-angle motions of an experimental nonplanar two-link flexible manipulator. Such results are lacking in the literature, but are very important for assessing the utility of trajectory optimization in the presence of modelling and tracking errors. Again, significant reductions in vibration are demonstrated by using the global optimization approach for trajectory generation.

Examiners:

Dr. I. Sharf, Supervisor (Dept. of Mechanical Engineering)

~~Dr. A. Doige, Department Member (Dept. of Mechanical Engineering)~~

~~Dr. R. Podhorodeski, Department Member (Dept. of Mechanical Engineering)~~

~~Dr. B. Tabarrok, Department Member (Dept. of Mechanical Engineering)~~

Dr. P. Agathoklis, Outside Member (Dept. of Electrical and Computer Engineering)

Dr. K. Gupta, External Examiner (Simon Fraser University)

Table of Contents

Abstract	ii
Table of Contents	iv
List of Figures	vii
List of Tables	ix
Dedication	x
Acknowledgements	xi
1 Introduction	1
1.1 Literature Review	2
1.1.1 Linear Time-Invariant Models	3
1.1.2 Nonlinear Models	6
1.1.3 Optimization for Nonlinear Models	7
1.2 Thesis Outline	11
2 Optimal Motion Planning Problem	14
2.1 Introduction	14
2.1.1 Mathematical Problem Statement	15
2.2 Numerical Solution Methods	15
2.2.1 Direct Methods	17
2.2.2 Indirect Methods	23
2.2.3 Literature Review	28
2.3 Proposed Formulation	32
2.3.1 Manipulator Dynamics	33
2.3.2 Objective Functional	34
2.3.3 The Pontryagin Maximum Principle	38
2.3.4 Application of the PMP	40
2.3.5 Comparison to Other Methods	43

2.4	Summary	46
3	Numerical Solution	47
3.1	Introduction	47
3.2	Numerical Implementation	48
3.2.1	Spline Interpolation	49
3.2.2	Numerical Integration	51
3.2.3	Nonlinear Optimization	52
3.2.4	Verification	57
3.3	The Coupling Map	58
3.4	Numerical Examples	61
3.4.1	REMSPATIAL	61
3.4.2	LAURA-SARA	65
3.4.3	REMGRAVITY	72
3.5	Summary	79
4	Experimental Evaluation	80
4.1	Introduction	80
4.2	The Experimental Flexible Manipulator	81
4.2.1	Manipulator Construction	81
4.2.2	Manipulator Model	82
4.2.3	Vibration Testing and Parameter Identification	84
4.2.4	Joint Controller	88
4.3	Trajectory Comparison	88
4.3.1	Task Descriptions	88
4.3.2	Trajectory Generation	89
4.3.3	Trajectory Tracking Performance	95
4.3.4	Strain Energy Comparison	99
4.3.5	Cost Comparison	103
4.4	Summary	105
5	Conclusions	106
	References	109
A	A Proposed Motion Planner	115
A.1	Background	116
A.2	Dynamics	119
A.3	Choosing the Outputs	120
A.4	Solving for the Second Output Function	123
A.4.1	Solving the Characteristic Equations	125

TABLE OF CONTENTS

A.5 Coordinate Transformation Conditioning	127
A.6 Summary	129

List of Figures

2.1	Dynamic programming discretization for the scalar case ($n = 1$). . . .	22
3.1	The Quasi-Newton minimization algorithm, incorporating a line-search and restart, with function value and gradient subroutines.	56
3.2	REMSPATIAL manipulator.	62
3.3	REMSPATIAL joint path and path speed.	64
3.4	REMSPATIAL joint path speed superimposed on joint path.	64
3.5	Optimal joint path (solid line) superimposed on the Coupling Map for the REMSPATIAL manipulator.	65
3.6	REMSPATIAL joint accelerations.	66
3.7	REMSPATIAL joint rates.	66
3.8	REMSPATIAL joint angles.	67
3.9	REMSPATIAL elastic coordinates, the flexible link tip deflection and slope.	67
3.10	REMSPATIAL strain energy.	69
3.11	Photo of LAURA-SARA manipulator [VAN VLIET AND SHARF, 1998].	69
3.12	SARA joint space path and path speed.	70
3.13	SARA path speed superimposed on joint space path.	70
3.14	Optimal joint path (solid line) superimposed on the Coupling Map for the LAURA-SARA manipulator.	71
3.15	SARA joint accelerations.	73
3.16	SARA joint rates.	73
3.17	SARA joint angles.	74
3.18	Elastic coordinates of LAURA's two flexible links.	74
3.19	LAURA's strain energy.	76
3.20	REMGRAVITY joint path and speed—slow maneuver.	76
3.21	REMGRAVITY path speed superimposed on joint path for slow maneuver.	77
3.22	REMGRAVITY joint path and speed—fast maneuver.	77
3.23	REMGRAVITY path speed superimposed on joint path for fast maneuver.	78
3.24	Optimal path for slow maneuver (white) and fast maneuver (black) superimposed on Coupling Map for REMGRAVITY.	78

4.1	Experimental flexible manipulator.	82
4.2	Experimental manipulator model, including virtual hinge and payload, shown in the zero configuration.	84
4.3	Power spectral density of flexible link tip slope under different modes of excitation.	86
4.4	Fifth-order polynomial joint profile, along with joint rate and acceleration.	90
4.5	Task 1: Paths and path speeds.	92
4.6	Task 2: Paths and path speeds.	92
4.7	Overhead view of 120° slew of shoulder joint.	94
4.8	Coupling Map for Task 1: (0°, 0°) to (120°, 0°).	96
4.9	Radial and tangential components of the linear acceleration of the elbow motor, as well as total magnitude, during the slewing of the shoulder joint.	96
4.10	Model and experimental joint rates for Task 1: “Polynomial” trajectory.	98
4.11	Model and experimental joint rates for Task 1: “Constrained” trajectory.	98
4.12	Model and experimental joint rates for Task 1: “Optimal” trajectory.	99
4.13	Task 1: Simulated strain energies.	101
4.14	Task 1: Experimental strain energies.	101
4.15	Task 2: Simulated strain energies.	102
4.16	Task 2: Experimental strain energies.	102

List of Tables

3.1	Physical properties of REMSPATIAL. The stiffness of the flexible link is $EI = 74.7 \text{ N}\cdot\text{m}^2$	62
4.1	Properties of the steel cylinder payload.	83
4.2	Parameters of the virtual hinge.	87
4.3	Independent PD controller feedback gains for joint angle and angular rate.	88
4.4	Maximum absolute trajectory tracking errors.	97
4.5	Task 1: Objective function costs for simulated trajectories.	104
4.6	Task 1: Objective function costs for experimental motions.	104
4.7	Task 2: Objective function costs for simulated trajectories.	104
4.8	Task 2: Objective function costs for experimental motions.	104

In memory of George William Pond.

Acknowledgements

Much effort has gone into completing this dissertation, and I would like to acknowledge the contributions and influence of several people. Foremost among them is my supervisor, Dr. Inna Sharf. Besides providing me with interesting and challenging opportunities, as well as financial support, she has been a prime source of motivation and valuable feedback through careful editing of my work. Most importantly, she had the patience to give me the freedom to learn from my own mistakes, and for that I am extremely grateful.

Without the help of John Van Vliet, the experiments presented here would have been an exceedingly slow and difficult process. I thank him for many helpful discussions and suggestions, and for providing the software required to conduct the experiments.

I would also like to thank Qiao Sun for her support and confidence in me, without which I cannot imagine finishing this work.

Finally, I would like to acknowledge the financial support of the University of Victoria.

Chapter 1

Introduction

Robotic manipulators are becoming commonplace components in factory automation around the world [KAPLAN, 1999]. The robots allow precise repetition of assembly, welding, and painting tasks beyond the endurance of a human worker. As robotic capabilities have developed, robots have not only replaced humans, but have acted as human surrogates in dangerous and hostile environments, such as in nuclear reactors, in outer space, and for hazardous waste remediation.

Inevitably, there is demand for increased performance with decreased cost. In some applications, both can be achieved by reducing the weight of the manipulator. A lighter manipulator allows faster task execution and less energy consumption. However, reduced weight often leads to significant structural flexibility. For example, harmonic-drive motors produce more torque in a lighter package but have a flexible transmission, compared to direct-drive motors. Weight is also a large factor in the cost of lifting manipulators into orbit, leading to the use of long, lightweight, and extremely flexible robots for space applications [BOOK, 1993].

Flexibility in robotic manipulators is generally regarded as troublesome and much

effort is typically expended in avoiding or cancelling vibration. For example, robot tasks are often defined in terms of the end-point trajectory in a particular task space. Following such a specified trajectory is known as tip-trajectory tracking. Vibration disturbs the motion of the end-point, however, and is inherently detrimental to performance for such tasks.

Although there has been a great deal of research in the area of controlling flexible manipulators to follow a desired tip-trajectory, much less work has been conducted towards choosing the trajectory itself. For point-to-point motions, the task specifies only a desired point, either in task space or joint space, allowing the intervening trajectory from the current point to be optimized. Determining optimal point-to-point motions in joint space is the objective of this work.

1.1 Literature Review

This section reviews research related to trajectory generation for flexible manipulators. To facilitate the following discussion, a set of features is first described by which each work may be classified. Precise definitions will be furnished later, but for the present purposes these brief descriptions should suffice.

Dynamics model: The dynamics equations modelling flexible manipulators are non-linear in general, though researchers have sometimes approximated them by linear time-invariant (LTI), quasi-static, or even rigid models to simplify trajectory generation.

Global vs. local optimization: Global, or functional, optimization minimizes the integral of a cost function evaluated over the entire time interval of the trajectory

and provides the most desirable measure of the manipulator's performance. However, local (*i.e.* instantaneous) optimization requires less computational effort.

Optimization objective: Objective functions have included minimization of tip-trajectory tracking errors, joint rate norm, time, strain energy, residual vibration, "vibration energy", or weighted combinations thereof.

Candidate trajectory class: A joint trajectory may be decomposed into a path in joint space and a speed along that path. Specifying the joint angles as functions of time directly gives both the path and speed implicitly. Some researchers have optimized either the path or the speed, assuming the other is given. Depending on the application or the numerical method, optimal joint accelerations or torques may also be sought. In these cases, motion planning is only concerned with open-loop solutions, that is, functions of time and not state. Typically, the optimization determines the parameters of a trajectory profile expressed in terms of polynomials or sinusoids.

Experimental evaluation: A small fraction of the trajectory generation methods have been evaluated experimentally, particularly for general, nonlinear dynamics models.

1.1.1 Linear Time-Invariant Models

The simplest flexible manipulator is a single flexible link attached to a rigid, rotating hub. The hub has a single rotational degree of freedom about a fixed axis and the deflection of the flexible link is constrained to bending in the plane perpendicular to

the rotational axis. Slew maneuvers for point-to-point positioning of similar systems have been studied extensively, particularly for space applications [HYLAND ET AL., 1993]. The joint space is one dimensional, however, so the joint path is fixed and trivial from a motion planning perspective. That leaves only the speed profile along the path to be determined, perhaps implicitly through the applied joint torque profile.

This problem is simplified considerably by the fact that the manipulator dynamics are well-approximated by an LTI system. We give here a sampling of the literature on the point-to-point positioning of a single flexible link. The general approach is to smooth the discontinuous time-optimal solution to reduce vibration. For example, SERNA AND BAYO [1990] assume a rectangular tip acceleration profile whose discontinuities are smoothed by cubic polynomials. The profile parameters are determined to minimize travel time subject to torque and frequency constraints. Because the dynamics are LTI, the inverse dynamics are solved in the frequency domain to give the required torque. For the same problem, YAO AND CHENG [1995b] specify the end-point trajectory as a fifth-order polynomial plus a truncated Fourier series. The inverse dynamics are then calculated in the time domain. The trajectory series coefficients are determined to minimize a weighted sum of the global vibration energy and the residual vibration energy, where vibration energy is the strain energy plus some cost related to the elastic rates (see §2.3.2 for definitions). THOMPSON ET AL. [1989] consider a nominal, discontinuous bang-bang torque profile smoothed by arctangent functions. The smoothing parameters are chosen and then the optimal torque is determined to minimize the slewing time. Although the smoothing gives a near-minimum-time solution, it allows a trade-off between minimizing maneuver time and minimizing residual vibration. The same trade-off is illustrated by BHAT AND MIU [1992], although in a different manner. The cost function in their global minimization

objective is a weighted sum of squares of time derivatives of the input torque up to the n th order. The optimal solution is continuous up to the $(n - 1)$ th derivative, enforcing the smoothness constraint. An appropriate choice of cost weights allows the selective penalization of input frequencies above a particular roll-off frequency. Experimental results show that, in the presence of input bounds, the smoothed inputs reduce residual vibration but require longer slewing time. Rather than smooth the time-optimal bang-bang solution, however, ÖNSAY AND AKAY [1991] construct an actuator with increased switching speed and present experimental results. The switching speed limits the number of modes for which residual vibration can be eliminated.

Experimental results are also given by TSUJI ET AL. [1995] for a vertical flexible link on a cart. The objective is to find the commanded force on the cart which minimizes residual vibration for a given terminal position and time. An LTI model of the apparatus is obtained as a serial chain of lumped masses with a single spring and damper between each adjacent pair of masses. The commanded force is smoothed by applying it to an additional virtual mass placed before the cart mass and connected to the ground by a virtual spring and damper. Given bounds on the cart displacement, velocity, and acceleration, the commanded force is optimized using linear programming to minimize residual vibration. The authors show that residual vibration is further reduced by designing the PD feedback controller first, then designing the trajectory (given implicitly by the commanded force) while taking the controller into account.

An alternative method for eliminating residual vibration in LTI systems is referred to as input shaping [SINGER AND SEERING, 1990]. A copy of the commanded input (of the same shape but perhaps delayed and scaled) is superimposed on the com-

manded input to exactly cancel residual vibration. Some robustness properties are derived and the method is successfully applied to a nonlinear model of the Shuttle Remote Manipulator System undergoing a relatively small angle point-to-point maneuver. During large angle ($> 45^\circ$) slewing of a planar, two link flexible manipulator, however, HILLSLEY AND YURKOVICH [1993] show experimentally that changes in the system frequencies with configuration render the performance of input shaping alone less than satisfactory. We will therefore turn our focus to the use of nonlinear models to further reduce vibration in flexible manipulators.

1.1.2 Nonlinear Models

In general, the equations describing the dynamics of robot manipulators, whether rigid or flexible, are nonlinear. Even when the theory of linear elasticity is employed to model flexible link deflection, as is typically the case, large configuration changes or high joint rates may result in significant deviations from an LTI model.

For rigid serial-chain manipulators, it is possible to choose a coordinate transformation and a feedback such that the nonlinear dynamics are represented by an LTI system in the new coordinates. This technique, referred to as *feedback linearization*, entails no approximation of the dynamics and so it, and the related concept of *differential flatness* [VAN NIEUWSTADT AND MURRAY, 1997], have proven to be powerful tools for analysis and synthesis. Unfortunately, WANG AND VIDYASAGAR [1991] provide an example of a flexible manipulator which is not input-state feedback linearizable. Thus, trajectory generation, tracking, and control for flexible manipulators is fundamentally more difficult than for rigid manipulators.

Other than an approximate linearization of the dynamics, one of the simplest ap-

proximations is to assume quasi-static operation of the manipulator during sufficiently slow motions. Given a desired tip position and initial joint configuration, XI AND FENTON [1995] solve the flexible manipulator inverse kinematics using the Newton-Raphson method to generate a sequence of joint angles. This sequence converges to a joint configuration which compensates for static deflections and positions the tip correctly. VAILLANCOURT AND GOSSELIN [1994] analyzed quasi-static compensation for a simple model of the rigid Special Purpose Dexterous Manipulator (SPDM) at the end of the flexible Space Station Remote Manipulator System (SSRMS). They found, however, that the accuracy of the quasi-static deflection was not sufficient when the locked SSRMS was oscillating—a common scenario. The accuracy was improved by including a simple dynamics model.

PETTERSON ET AL. [1990] consider a slow maneuver of a symmetric rod sagging under gravity. The nonlinear dynamics model includes coupling between horizontal and vertical motion. They assume a trapezoidal speed profile and are able to determine the profile parameters in closed form which suppress residual vibration, as demonstrated experimentally. In general, however, one must resort to numerical optimization methods, which we now discuss.

1.1.3 Optimization for Nonlinear Models

In this section we review research directed towards optimizing point-to-point maneuvers for nonlinear models of flexible manipulators. Each work is distinguished by the optimization objective and the numerical method employed to determine the optimal trajectory. To compare entire trajectories, the objective is given by a cost function integrated over the time interval of the motion. Optimization with respect to such

an objective is known as *functional*, or *global*, optimization and will be elaborated in §2.1.1.

Local Optimization

In some cases, it may be possible to generate an adequate trajectory by solving an instantaneous, or *local*, optimization at the current time and state. For example, when the manipulator is kinematically redundant, the tip-trajectory is specified but the joint motion is not completely determined. Kinematic redundancy occurs if there are more actuated degrees of freedom than necessary for the task. KIM AND PARK [1997] resolve the redundancy locally to reduce vibration while simultaneously tracking the tip-trajectory. Although local optimization is much simpler than global optimization computationally, it generally does not guarantee that all states will attain a desired final value and is therefore not appropriate for point-to-point maneuvers of non-redundant manipulators. Appendix A outlines a proposal for a global optimization approach which is locally computable but does not suffer from this drawback.

The Coupling Map, proposed by TORRES AND DUBOWSKY [1993], combines aspects of both local and global optimization. Developed particularly for planning joint paths of a rigid manipulator mounted on a flexible structure, its central concept is the measure of “coupling”, the disturbance caused by motion of the rigid manipulator to the flexible structure. The Map graphically presents local information in a two-dimensional joint space. For example, instantaneous directions of least coupling are indicated for joint configurations throughout the Map while regions of relatively high coupling are darkly shaded. Instead of global optimization, a human operator then uses the local information and heuristics to plot a joint path between the desired configurations. The speed along the chosen joint path follows a predetermined

polynomial profile. TORRES ET AL. [1994] give experimental results for a two-link planar, rigid arm attached to the end of a flexible structure. Although easier to calculate relative to global optimization, trajectory planning based on the Coupling Map is not automated and is restricted to planning up to three degrees-of-freedom only. Because of its simplicity, the Coupling Map provides a ready comparison for more complex schemes and so will be discussed in more detail in §3.3.

Global Optimization

Since the global optimization methods are relatively intense computationally, the associated numerical implementation issues assume a greater importance. For this reason, a thorough comparison of global trajectory optimization for flexible manipulators will be given in §2.2.3, after the requisite terminology and problem definition establish a suitable foundation. Here, we survey the relevant work in this area with respect to the basic optimization approach, the objective, and the manipulator system considered.

For example, DOHRMANN AND ROBINETT [1994] used dynamic programming to minimize the square of the joint torque during the point-to-point maneuver of a single rigid link with a flexible payload. MEIROVITCH AND CHEN [1995] employed perturbation methods to separate the rigid and elastic dynamics of a flexible, redundant, space robot. For a given tip trajectory, they then resolved the redundancy of the rigid manipulator to globally minimize the norm of the joint rates, and designed a feedback control to reduce the elastic disturbance to the nominal joint trajectory. Note that neither of these works determines a trajectory based on a cost related to vibration.

A typical approach to global optimization is to assume some functional form for the solution and to determine the optimal parameters. MOHRI ET AL. [1998] calculate

the optimal knot points in a B-spline interpolation of the speed profile along a given joint path. Their objective is to minimize the residual vibration energy of a general flexible manipulator at a fixed terminal time. Rastegar *et al.* have used truncated Fourier series to synthesize joint accelerations for point-to-point maneuvers [TU AND RASTEGAR, 1993; KIM AND RASTEGAR, 1997]. First, the frequencies of the basis functions are chosen not to coincide with the natural frequencies of the system. Then, the series coefficients are found to minimize the energy of the higher harmonics in the required joint torques, as determined by a rigid manipulator model. However, no evaluation of the efficacy of this method for flexible manipulator models is given. The same method is used by TU ET AL. [1994] for a flexible positioning system. In this case, the series coefficients are determined by minimizing the least squares error with a specified tip trajectory. Vibration reduction is again not explicitly considered. EISLER ET AL. [1993] also did not penalize link vibration, but minimized travel time or tip tracking error for a planar two-link flexible manipulator subject to joint torque constraints. Optimization was of 27 joint torque values, which were linearly interpolated for simulation. To speed optimization, LIU AND KUJATH [1996] generated point-to-point joint trajectories for a two-link flexible manipulator using profiles with only one unknown parameter. It was then possible to graphically determine the optimal parameter value to minimize the norm of the elastic coordinates and rates. Finally, YAO AND CHENG [1995a] expanded the joint angles as fifth-order polynomials plus truncated Fourier series. They determined the series coefficients to minimize the global vibration energy, and then calculated the polynomial coefficients to satisfy the point-to-point maneuver with zero joint rates and accelerations at the end-points. The optimal trajectory generated with a four term Fourier series was compared in simulation to the (speed) optimized straight-line path in joint space for

a planar, two-link flexible manipulator.

To summarize, this section has reviewed the literature on trajectory optimization for general, nonlinear models of flexible manipulators. Of the works considered, only the last two cited above have considered global trajectory optimization to reduce vibration. The first severely limited the class of comparison trajectories, while the second only presented results for the planar, two-link case. Furthermore, no experimental results have been reported other than for the heuristic path-planning tool referred to as the Coupling Map.

1.2 Thesis Outline

The purpose of this dissertation is to determine minimal vibration joint trajectories for point-to-point maneuvers of flexible manipulators. It is assumed that the dynamics model of the manipulator is given and that it may be nonlinear and time-varying in general. The objective is to minimize an explicit, quantitative measure of the vibration, where the vibration is determined in accordance with the dynamics equations. In addition, a restriction is placed on the manner in which the dynamics equations may be satisfied. A significant amount of effort is often invested in the formulation, software implementation, verification, and validation of the dynamics model, particularly for space-based manipulators. To leverage previous work and preserve the integrity of the model, we therefore require that the motion planning algorithm determine the vibration only through the existing dynamics simulation, rather than employing some alternative approximation which may aid trajectory optimization.

In Chapter 2, the motion planning problem is formulated mathematically as a functional optimization problem and a broad array of applicable numerical solution

methods are reviewed. The numerical methods employed in the literature for flexible manipulators are then interpreted and compared within this context. Finally, a suitable indirect optimization formulation based on the Pontryagin Maximum Principle is chosen.

For general, nonlinear dynamics models, the functional optimization problem cannot be solved analytically but must instead be tackled numerically. Chapter 3 therefore begins with a detailed description of the numerical methods employed during each step of the solution. First, a class of comparison trajectories is constructed using a finite number of parameters and cubic spline interpolation. For a given set of parameter values, the motion of the manipulator is then determined by numerical integration of the dynamics equations. Finally, an iterative numerical optimization algorithm updates the solution parameters to obtain a joint trajectory which reduces vibration.

The optimal joint trajectories are presented, along with pertinent simulation output, for three example manipulator models. Two of these manipulator models are constructed specifically to help verify the trajectory optimization, while the third corresponds to an existing experimental manipulator. The point-to-point maneuvers are also chosen for verification purposes, in the first case by admitting a nontrivial zero-vibration joint path. In all cases, the optimal joint path can be physically justified. The globally optimal joint paths are also interpreted with respect to the Coupling Map path-planning heuristics, and are consistent when the assumptions of the Coupling Map derivation hold.

While the previous results were obtained through simulation only, Chapter 4 provides an experimental evaluation of optimal trajectories obtained for a laboratory-based flexible manipulator. Again for verification purposes, the physical construction

of the manipulator is chosen to admit a nontrivial zero-vibration joint path. Vibration tests are conducted to characterize the natural frequencies of the assembled system and to allow identification of model parameters. Using this model, the optimal joint trajectory is generated and compared to simpler trajectories, both in simulation and experimentally. Although modelling and trajectory tracking errors prevent the experimental performance from matching the performance predicted by simulation, the experimental evaluation nevertheless shows significant reduction in vibration for the optimized trajectory.

Finally, Chapter 5 presents conclusions drawn from the research and discusses possible avenues for further investigation.

Chapter 2

Optimal Motion Planning Problem

2.1 Introduction

This chapter considers the trajectory optimization problem for which the objective is to minimize the value of a cost function integrated over the time interval of the trajectory. The form of this problem is one of functional optimization and, generally, no closed-form solution is available. However, functional optimization problems arise in many application areas of science and engineering and a vast array of techniques for solving them are available. Therefore, after formally stating the problem, a summary of numerical methods for solving functional optimization problems is given in §2.2, highlighting the advantages and disadvantages of each. Based on this comparison, §2.3 presents a formulation of a solution technique appropriate to the trajectory optimization problem for flexible manipulators.

2.1.1 Mathematical Problem Statement

The general optimization problem considered in this chapter can be stated as follows.

The objective is to minimize the cost functional

$$J[\mathbf{u}] = \int_0^T f^0(\mathbf{x}, \mathbf{u}) dt, \quad (2.1)$$

over all admissible control functions $\mathbf{u}(\cdot) \in \mathcal{U}$, subject to the state dynamics

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (2.2)$$

and boundary conditions

$$\mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{x}(T) = \mathbf{x}_d \quad (2.3)$$

for state $\mathbf{x}(t) \in \mathbb{R}^n$, control $\mathbf{u}(t) \in \mathbb{R}^m$, fixed time T , and fixed states \mathbf{x}_0 and \mathbf{x}_d .

It is assumed that $\mathbf{f} \in \mathbb{R}^n$ and $f^0 \in \mathbb{R}$ are continuous in (\mathbf{x}, \mathbf{u}) and continuously differentiable in \mathbf{x} .

We will see in the next section that numerical solution techniques, which are inherently approximate, can be distinguished by the manner in which they satisfy the dynamics constraint, Eqn. (2.2), the boundary condition constraints, Eqns. (2.3), and optimality of the cost functional, Eqn. (2.1). Furthermore, for each technique, the specification of the set of admissible control functions \mathcal{U} is crucial to the determination of a solution to this problem.

2.2 Numerical Solution Methods

The solution to the functional optimization problem posed above is a pair of vector-valued functions: the control input $\mathbf{u}^*(\cdot)$ from the set of admissible functions \mathcal{U} , and

the state $\mathbf{x}^*(\cdot)$ satisfying the dynamics constraint Eqn. (2.2) for $\mathbf{u} = \mathbf{u}^*$. In general, \mathcal{U} is an infinite dimensional function space from which we choose an optimal solution, if one exists. Since closed-form solutions to functional optimization problems cannot be obtained in general, discrete numerical techniques must be employed. Such numerical techniques in turn require that the stated infinite dimensional optimization problem be reformulated as a finite dimensional problem.

In this section we describe such methods, broadly classified as being *direct* or *indirect*. In §2.3 we will see that the proposed formulation combines aspects of both classes of methods. Throughout this section, technical details such as theoretical derivation and convergence analysis are kept to a minimum to highlight the salient features of each method with respect to numerical implementation. The purpose is to convey sufficient information to justify the proposed formulation, for which such details will be discussed in the next section.

The direct methods first restrict the admissible functions to a set of finite dimension $\mathcal{U}_p \subset \mathcal{U}$ and then determine the optimal solution on \mathcal{U}_p by applying necessary and sufficient conditions. Since \mathcal{U}_p is finite-dimensional, application of these conditions is amenable to discrete numerical techniques. Under certain conditions, the sequence of solutions obtained as $p \rightarrow \infty$ with $\mathcal{U}_p \subset \mathcal{U}_{p+1}$ converges (in some sense) to the solution of the original problem.

In contrast, the indirect methods first apply a necessary condition for the minimization of the functional J , obtaining a set of ordinary differential equations (ODE's) to be satisfied by the optimal solution. Unlike the simpler initial value problem for ODE's, however, the boundary conditions are not all specified at a single time, but are partitioned between the initial and final times. This problem is referred to as a *two-point boundary value problem* (TPBVP). This infinite dimensional problem may

be rendered finite by discretizing the ODE's. Alternatively, the correct determination of the unknown boundary values at one endpoint is a finite-dimensional optimization problem whose dimension depends on the problem, not on any approximation.

Thus we are again reminded that an important characteristic distinguishing the various numerical methods is the manner in which the optimal solution is approached. Of course, the final solution obtained by any method must simultaneously satisfy the dynamics constraints, the boundary conditions, and the optimality conditions. However, since iterative methods are used to solve the nonlinear optimization problem in general, each of these conditions may either be satisfied identically, for each iteration, or may only be satisfied in the limit as the optimal solution is attained. This distinction will be a key factor in our choice of formulation.

2.2.1 Direct Methods

This section describes three main types of direct method, each with different possible modifications which may be necessary according to the particular problem under consideration. The first type is the Rayleigh-Ritz method, with variations known as the enhanced Rayleigh-Ritz method and the assumed-modes (assumed-time-modes) method. The second and third types of methods are the well-known finite element and dynamic programming methods. The Rayleigh-Ritz and its variations will be described in more detail since the proposed method relies on some of the associated formulation.

Before describing the direct methods, we will provide some common notation. Our problem, as stated in §2.1.1, is to minimize an objective functional subject to constraints given by ordinary differential equations. However, the direct methods

apply to the minimization of the objective functional free of such constraints. We will give examples in §2.2.3 of how our problem can be reformulated to apply direct methods, but in the meantime we consider the following problem:

$$\min_{\mathbf{w} \in \mathcal{W}} J[\mathbf{w}] := \int_0^T F(\mathbf{w}) dt \quad (2.4)$$

where we assume that F is sufficiently differentiable and that the set \mathcal{W} is given. The function F could also include time derivatives of \mathbf{w} as additional arguments; they are omitted here for simplicity.

Rayleigh-Ritz Methods

The *Rayleigh-Ritz* method considers a sequence of optimization problems, each on a finite dimensional space, such that the solutions converge to the solution of the original problem as the dimension goes to infinity. First, the space of admissible functions \mathcal{W} is defined to be those functions which are sufficiently differentiable and satisfy all of the boundary conditions. Then, a sequence of finite dimensional subspaces \mathcal{R}_p are formed, known as *Ritz* spaces [MEIROVITCH, 1997], with the property that $\mathcal{R}_p \subset \mathcal{R}_{p+1} \subset \mathcal{W}$.

An approximate solution to the functional optimization problem is then sought in the form

$$\mathbf{w}^{(p)}(t) = \sum_{i=1}^p \mathbf{a}_i \phi_i(t) =: \mathbf{A}\boldsymbol{\phi} \quad (2.5)$$

where $\mathbf{a}_i \in \mathbb{R}^n$ (the columns of $\mathbf{A} \in \mathbb{R}^{n \times p}$) are constants to be determined. The functions ϕ_i are sufficiently smooth on the interval $(0, T)$, satisfy the given boundary conditions, and form a basis of the p dimensional function space \mathcal{R}_p . To ensure convergence as $p \rightarrow \infty$, the set of functions $\phi_1, \phi_2, \dots, \phi_p, \dots$ is chosen such that

any subset of p members is linearly independent, and the entire set is *complete* in \mathcal{W} , where

Definition 1 *The set $\{\phi_i\}_{i=1}^{\infty}$ is complete if there exists a sufficiently large value of p (possibly depending on ε) such that*

$$\|\mathbf{w}^* - \mathbf{w}^{(p)}\|^2 < \varepsilon \quad (2.6)$$

for any $\mathbf{w}^* \in \mathcal{W}$, arbitrarily small $\varepsilon > 0$, and $\mathbf{w}^{(p)}$ given by Eqn. (2.5) [GELFAND AND FOMIN, 1963].

To determine the optimal solution $\mathbf{w}^{(p)} \in \mathcal{R}_p$, we now consider the objective functional J directly, substituting the expansion $\mathbf{w}^{(p)}$ of Eqn. (2.5) into the expression for J of Eqn. (2.1). The objective is now a functional of the elements a_{ij} of the coefficient matrix \mathbf{A} and the basis functions ϕ_i ,

$$J(a_{ij}) = \int_0^T F(\mathbf{A}\phi(t)) dt. \quad (2.7)$$

Since the basis functions ϕ_i are known, and their associated integrals can be calculated, J becomes a function of the np unknown constant coefficients a_{ij} . The infinite dimensional optimization problem has thus been reduced to finite dimensions by confining the search to a finite dimensional subspace of the space of admissible solution functions. Assembling the elements a_{ij} into a single vector \mathbf{a} , two necessary conditions for \mathbf{a} to locally minimize J are $\nabla J = \mathbf{0}$, and $\nabla^2 J > 0$. Together, these conditions are sufficient for \mathbf{a} to be a local minimum of J on the subspace \mathcal{R}_p .

Numerous other methods for the solution of the functional optimization problem are extensions of, or are at least related to, the Rayleigh-Ritz method. To improve convergence (with increasing p) in the presence of boundary conditions involving

derivatives, the *enhanced Rayleigh-Ritz* method expands the set of admissible functions to include *quasi-comparison functions*, which satisfy some of the boundary conditions identically and the rest in the limit [MEIROVITCH, 1997]. Another approach, referred to as the *assumed-modes* method, is applicable to problems with more than one independent variable, for example, where the unknown w is a function of both time t and state \mathbf{x} . The approximate solution is expanded as

$$\mathbf{w}^{(p)}(\mathbf{x}, t) = \sum_{i=1}^p \mathbf{q}_i(t) \phi_i(\mathbf{x}) \quad (2.8)$$

where the unknown coefficients $\mathbf{q}_i(t)$ are no longer constant but functions of time. When the basis functions are functions of time, the above methods are sometimes referred to as the *assumed-time-modes* method [ADIGÜZEL AND ÖZ, 1995].

Finite Elements

In describing the above methods, we stated that the basis functions are chosen from the set of admissible functions and therefore should be sufficiently smooth and satisfy all of the boundary conditions. Since the convergence rate as $p \rightarrow \infty$ is affected by the choice of basis functions, some skill must be exercised by the designer in making this sometimes difficult choice. The popularity of the *finite element* method stems in a large part from the automation of this process. Instead of requiring that the basis functions be defined over the entire interval $[0, T]$, each basis function is defined (and continuous) over a small sub-interval $[t_{i-1}, t_i]$ only. The linear combination of such basis functions, subject to suitable continuity constraints at the nodes t_i , forms a quasi-comparison function and is thus a solution as would be obtained by the enhanced Rayleigh-Ritz method [MEIROVITCH, 1997]. The disadvantage of the finite element method is that the number of elements needed to achieve a desired

level of convergence is generally greater than the number of basis functions required by the Rayleigh-Ritz method to achieve the same level of convergence. However, the ability to automatically formulate large classes of problems using standard sets of basis functions is highly suitable for computer implementation and far outweighs this slight disadvantage. Since the proposed formulation contains aspects of the finite element method, further details of the method will be discussed in subsequent sections.

Dynamic Programming

Although derived from a perspective different from the previous direct methods for functional optimization, the numerical implementation of the *dynamic programming* approach is similar in many respects [BRYSON AND HO, 1975]. As in the finite element method, for example, a sequence of discrete times is selected on the interval $[0, T]$. However, the key feature of the dynamic programming approach is to discretize the state at those times as well. The development of the numerical algorithm is then founded on the *principle of optimality*, which states that any sub-arc of an optimal curve is also optimal between the endpoints of the arc. The problem is now to search over the grid of discrete states to determine the optimal endpoints for each sub-arc.

For example, consider two successive discrete times t_{i-1} and t_i , $i = 1, \dots, N$ and particular states $w_{i-1,j}$ and $w_{i,k}$, $j, k = 1, \dots, M$ at those times respectively. Figure 2.1. The principle of optimality implies that the optimal curve from $w_{i-1,j}$, passing through $w_{i,k}$, to the desired final state w_T is the concatenation of the optimal curve from $w_{i-1,j}$ to $w_{i,k}$ and the optimal curve from $w_{i,k}$ to w_T . Assuming that the latter curve and its cost are known for each k , then we must determine the optimal curve between the points $w_{i-1,j}$ and $w_{i,k}$ for all pairs of points indexed by j and k . Each of these M^{2n} sub-problems (for $w(t) \in \mathbb{R}^n$) can in fact be solved by any of the

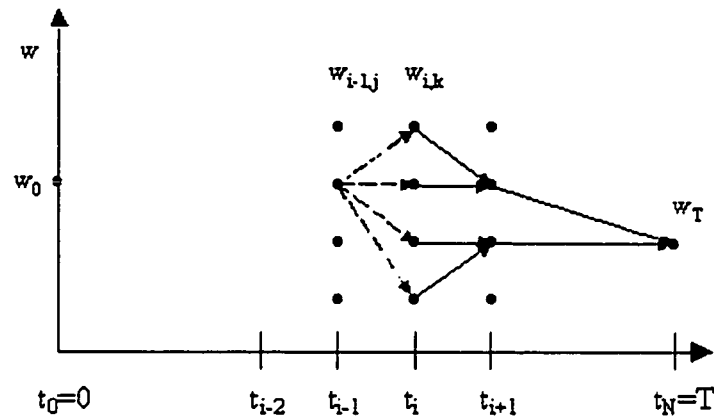


Figure 2.1: Dynamic programming discretization for the scalar case ($n = 1$). A solid line indicates the known optimal solution for a sub-problem, whereas dashed lines are candidates to be evaluated.

methods for functional optimization described in this section. In particular, simpler techniques can be used on the small time interval $[t_{i-1}, t_i]$ than would be possible on the entire interval $[0, T]$. The optimal solution from $w_{i-1,j}$ to w_T is finally determined by searching over all of the intermediate points $w_{i,k}$. In this way, the optimal solution to the original problem from w_0 to w_T is determined.

The primary disadvantage of dynamic programming is that the amount of computation grows exponentially with the dimension of the state. Also, minimization techniques that use discrete search converge slower than techniques that utilize derivative information. However, the increased computation leads directly to two significant advantages. The first is that, to within discretization error, dynamic programming is able to find the global minimum by searching the state space rather than resorting to conditions involving derivatives, which are inherently local. The second advantage is that necessary and sufficient conditions for an optimal solution do not need to be eval-

uated separately—the candidate solution has already been compared to neighbouring curves by construction.

2.2.2 Indirect Methods

Indirect methods for functional optimization are based on the consideration of a system of ODE's which represent the first order necessary condition that the first variation of the objective functional vanish. As for the direct methods, we will first provide a standard notation by which to compare the methods and in the next section will give examples of how these ODE's can be obtained. The system of ODE's we consider will be expressed by the n dimensional vector equation

$$\dot{\mathbf{w}} = \mathbf{g}(\mathbf{w}) \quad (2.9)$$

where, as usual, we assume that the function \mathbf{g} is sufficiently differentiable. Although \mathbf{g} may be an explicit function of time t , we omit this argument in the following discussion for simplicity.

In addition to the system of ODE's, the necessary conditions specify the value of some of the components of \mathbf{w} at $t = 0$ and the value of the remaining components at $t = T$. These partitioned boundary conditions, together with the system of ODE's, constitute a *two-point boundary value problem* (TPBVP). This section will describe four classes of numerical methods for solving TPBVP's: weighted residuals, finite differences, quasilinearization, and the shooting methods.

Weighted Residual Methods

It may happen that the system of ODE's to be satisfied was not derived from a functional optimization problem. Although this is not the case for our problem, the

numerical techniques developed for this situation have been applied to functional optimization and provide distinct advantages over the other indirect methods.

When there is no associated objective functional, the methods of *weighted residuals* form an objective functional from the approximation error, or *residual*. The approximate solution $\mathbf{w}^{(p)}$ is the same as for the Rayleigh-Ritz method, Eqn. (2.5). but the error is determined by how well this approximate solution satisfies the ODE's. Eqn. (2.9). The residual \mathbf{r} is therefore

$$\mathbf{r}(\mathbf{w}^{(p)}) = \dot{\mathbf{w}}^{(p)} - \mathbf{g}(\mathbf{w}^{(p)}) \quad (2.10)$$

and is an n dimensional vector-valued function of time. The residual is weighted by a complete set of functions ψ_i . To minimize the approximation error, we seek the coefficients \mathbf{A} in the expansion of $\mathbf{w}^{(p)}$ such that

$$\int_0^T \psi_i r_j dt = 0, \quad i = 1, \dots, p, \quad j = 1, \dots, n. \quad (2.11)$$

Interpreting Eqn. (2.11) as an inner product on a function space, this condition is equivalent to requiring that each element of the residual, r_j , be orthogonal to each of the p weighting functions ψ_i . The result is np (nonlinear) algebraic equations for the np unknown coefficients a_{ij} . Since the weighting functions ψ_i form a complete set, it can be argued that the only way r_j can remain orthogonal to every member as $p \rightarrow \infty$ is if $\lim_{p \rightarrow \infty} r_j = 0$. Such convergence in terms of inner products, such as those of Eqn. (2.11), is referred to as *weak* convergence.

Particular weighted residual methods are derived according to the choice of weighting functions [MEIROVITCH, 1997]. If the weighting functions coincide with the basis functions, $\psi_i = \phi_i$, we obtain the well-known *Galerkin's* method. A computationally simple choice of weighting functions is the set of polynomials $\psi_i = t^{i-1}$, appropriately

referred to as the method of *moments*. Choosing instead the Dirac delta functions, $\psi_i = \delta(t - t_i)$ where $t_i \in [0, T]$ are fixed, gives the method of *collocation*. In this case, a zero residual implies that the ODE's are satisfied exactly at the p specified points in the interval $[0, T]$. In the *least squares* method, the square of the norm of the residual is minimized. The disadvantage of this method is that the order of the algebraic equations to be solved is now doubled. As a whole, however, the weighted residual methods retain many advantages of the direct methods.

Finite Differences

Rather than forming approximate solutions by linear combinations of basis functions, an alternate approach is to approximate the derivatives in the ODE's by *finite differences*. First, the time interval $[0, T]$ is partitioned by N internal points t_i so that $t_0 = 0$ and $t_{N+1} = T$. In the simplest case, the points are equally spaced so that the subinterval length $h := t_{i+1} - t_i$ is constant. Next, the ODE's are approximated at the discretization points by finite differences. There is no unique finite difference formula for a derivative, so the choice is guided by accuracy, stability, and simplicity of computation. For example, by considering appropriate Taylor series expansions, a common finite difference formula for the first derivative is the central difference formula [ROBERTS AND SHIPMAN, 1972]

$$\frac{d\mathbf{x}(t_i)}{dt} = \frac{\mathbf{x}(t_{i+1}) - \mathbf{x}(t_{i-1})}{2h} + O(h^2), \quad (2.12)$$

which is modified appropriately at the boundary points. After substituting such expressions for the derivatives and truncating the higher order terms, the dynamics constraint then becomes a system of nN (nonlinear) algebraic equations in the nN unknowns $\mathbf{x}(t_i)$. As in the finite element method, accuracy is increased by increasing

the number of discretization points (decreasing h) or by retaining higher order terms in the finite difference formulae.

Quasilinearization

In general, linear TPBVP's are easier to solve than nonlinear problems. The *quasilinearization* method is based on the truncated Taylor series expansion of the nonlinear ODE's about a nominal solution [ROBERTS AND SHIPMAN, 1972]. The resulting linear TPBVP, which retains the original boundary conditions, is solved to obtain a new nominal solution. This process is repeated until the solution converges. The disadvantage of this approach is that the partial derivatives of the ODE's with respect to the independent variables must be obtained for the linearization.

Shooting Methods

The methods discussed in this section differ from all of the methods discussed earlier in one key respect. Whereas the previous methods always satisfy the boundary conditions and approximate the solution or the ODE's, the basic *shooting* method always satisfies the ODE's, to within numerical integration tolerance. This is made possible by estimating the missing boundary conditions at one end of the time interval $[0, T]$, say $t = 0$, and numerically integrating the ODE's from this end to the other as an initial value problem. A root-finding algorithm then updates the initial boundary value estimate using the error in the final boundary value. For nonlinear problems, this process is iterated until the boundary values converge to their prescribed values. A significant advantage of the shooting method over other indirect approaches is the reduced effort required for problem preparation, analysis, and computer implementation [ROBERTS AND SHIPMAN, 1972].

Two problems are commonly known to plague the shooting methods, however [VAN DE MOLENGRAFT ET AL., 1994; MEIROVITCH AND CHEN, 1995; BURDEN AND FAIRES, 1989]. The first is that the initial boundary value estimate may not lie within the region of convergence of the optimal value for the particular root-finding algorithm employed. For the second problem of instability, the numerical integration, beginning from the initial boundary value estimate, cannot proceed to the final time T . Both of these problems can be remedied by providing accurate initial boundary value estimates, but this is difficult to do in most cases, particularly since these estimates may not have any physical meaning.

To improve stability, another approach is to partition the time interval $[0, T]$ into N subintervals $[t_{i-1}, t_i]$ and employ the shooting method on each subinterval separately. Enforcing continuity of the solutions provides boundary conditions for each subinterval. However, the disadvantage of this so-called *parallel* or *multiple* shooting method is that the number of unknown functions is increased by a factor of N .

A more popular approach is to combine the shooting algorithm with a *continuation* or *homotopy* method [MEIROVITCH AND CHEN, 1995]. The idea is to use the solution to a simpler problem as the initial guess for a more difficult problem, constructing a sequence of problems which terminate with the original problem. More formally, if we seek the solution to $\mathbf{F}(\mathbf{w}) = \mathbf{0}$ but can easily solve $\mathbf{G}(\mathbf{w}) = \mathbf{0}$, then we solve $\mu\mathbf{F}(\mathbf{w}) + (1 - \mu)\mathbf{G}(\mathbf{w}) = \mathbf{0}$, $0 \leq \mu \leq 1$, beginning with $\mu = 0$ and ending with $\mu = 1$. How \mathbf{G} is chosen affects how quickly the solution to $\mathbf{F}(\mathbf{w}) = \mathbf{0}$ can be obtained, if at all. Typically, simplifying \mathbf{F} , for example by linearization, provides a suitable \mathbf{G} . Another approach suggested for dealing with instability is to perform the numerical integrations over the time interval $[0, \mu T]$ [ROBERTS AND SHIPMAN, 1972]. An

important problem, seldom considered, is whether the solutions to $\mathbf{G}(\mathbf{w}) = \mathbf{0}$ and $\mathbf{F}(\mathbf{w}) = \mathbf{0}$ belong to the same *continuation brunch* or *homotopy class* [KELLER, 1976]. If not, then the initial solution can not be continuously deformed into the desired solution, resulting in failure to find a solution for some $\mu < 1$.

2.2.3 Literature Review

The previous discussion makes it clear that there are many ways to solve functional optimization problems. Some methods differ primarily in their theoretical foundations, which we have not discussed in depth, but in fact result in equivalent numerical algorithms. For example, KELLER [1976] shows that collocation is equivalent to finite differences for particular interpolating polynomials and difference formulas. Many solutions proposed in the literature also combine aspects of more than one method. This section reviews applications of functional optimization to trajectory generation for flexible manipulators.

The most common approach is direct optimization based on some form of the Rayleigh-Ritz method. Recall that this method represents the solution as a series with given basis functions and unknown coefficients to be determined, Eqn. (2.5). To maintain the constraint that the dynamics equations always be satisfied, the series expansion can not be for all n of the states, but for either the m inputs or m of the states, where $m < n$ for flexible manipulators. The remaining $n - m$ states are determined by integrating the dynamics equations. In the works that follow, Fourier series are the most popular, with general polynomial or spline basis functions also used. Once the problem has been discretized, the unknown parameters are determined by a nonlinear optimization method which, unless otherwise specified, is

a direct search and does not employ gradient information.

For example, Rastegar *et al.* have synthesized joint accelerations for point-to-point maneuvers as truncated Fourier series [TU AND RASTEGAR, 1993; KIM AND RASTEGAR, 1997]. The frequencies of the basis functions are chosen not to coincide with the natural frequencies of the manipulator. However, the authors point out that the nonlinear terms in the dynamics give the input torques higher harmonics than are present in the accelerations. The objective is therefore to minimize induced vibration by minimizing the energy of the higher harmonics in the torques, as determined by a rigid manipulator model. However, no comparison is made of the reduction in vibration as only the rigid model is simulated. The same method is used by TU ET AL. [1994] to expand the tip accelerations of a flexible positioning system. In this case, the series coefficients are determined by minimizing the least squares error with a specified tip trajectory. Vibration reduction is again not explicitly considered. Results are presented for globally defined basis functions but segmenting the time interval and using local bases is also proposed (as in the finite element method).

MOHRI ET AL. [1998] optimize the speed profile along a given joint path to minimize the residual vibration energy at a fixed terminal time. The speed profile is represented by a B-spline interpolation of knot points, which are the unknown parameters to be optimized. Two direct search methods, not requiring gradient information, are proposed to iteratively reduce the objective while maintaining the fixed terminal time.

To speed computation, LIU AND KUJATH [1996] generate point-to-point joint trajectories for a two-link flexible manipulator using profiles with only one unknown parameter. The objective is to minimize the integral of the norm of the elastic coordinates and rates. Since there is only one parameter to be optimized, the search is conducted graphically. The first joint trajectory profiles considered are termed

“linear segment with parabolic blend” and have a fixed terminal time with a trapezoidal joint rate profile for each joint independently. The optimization parameter is the duration of the linear segment, that is, the length of the top of the trapezoidal joint rate profile. The search concludes that this time should be as short as possible, giving longer acceleration and deceleration times. Fifth-order polynomial joint trajectories are also considered with final joint acceleration to be optimized. This parameter, together with the boundary conditions of given joint angles, zero joint rates, and zero initial acceleration determined the six coefficients of the trajectory. The optimal final acceleration was zero, a value commonly used for polynomial joint trajectories. Recognizing that the previous profiles only generate straight-line paths in joint space, the authors also consider simple joint sequences. Each joint is moved in turn, rest-to-rest, according to the optimal fifth-order polynomial while the other joint remains fixed. In this case, the duration of the first joint’s motion is optimized.

A broader class of trajectories was considered for a two-link flexible manipulator by EISLER ET AL. [1993]. In this paper the authors tackled the problems of minimum time and minimum global tip-tracking error subject to torque limits and final-time constraints on tip position, joint angles, and rates. No explicit cost was associated with vibration. For simulation, the control torques were approximated by linear interpolation of 27 node values. The optimization of the node values employed gradient information but, because of the complexity of the dynamics and constraints, the derivatives were obtained by central finite differences rather than analytically. Finally, an optimal motion for a rigid manipulator initialized a homotopy algorithm ($\mu = 0$), although this solution was sufficiently accurate to form the initial guess for the optimization of the original flexible manipulator problem ($\mu = 1$).

YAO AND CHENG [1995a] considered fixed terminal time point-to-point maneuvers

of a planar, two-link flexible manipulator. They expanded the joint angles as a Fourier series but added a fifth-order polynomial, citing faster convergence for point-to-point maneuvers [NAGURKA AND YEN, 1990]. Only the series coefficients were optimized and then the polynomial coefficients were calculated to satisfy the point-to-point maneuver with zero joint rates and accelerations at the end-points. The objective was to minimize the global vibration energy, with a penalty function included to eliminate residual vibration. Optimization was performed by a quasi-Newton method with the gradient approximated numerically. Vibration minimization was compared for three cases: the optimal trajectory generated with a two- and four-term Fourier series for each joint and the (speed) optimized straight-line path in joint space. All three cases give a noticeably different solution and although the four-term series shows the least vibration, it is not clear that the series has converged, that is, that the solution could not be significantly improved by including more terms.

For flexible manipulator trajectory generation, few researchers have considered functional optimization methods other than the above Rayleigh-Ritz methods. For example, DOHRMANN AND ROBINETT [1994] used dynamic programming to minimize the square of the joint torque during the point-to-point maneuver of a single rigid link with a flexible payload. No other details of the numerical implementation are given.

The only indirect method applied to trajectory generation for flexible manipulators has been the shooting method, which MEIROVITCH AND CHEN [1995] employed for a flexible, redundant space robot. However, the elastic deflections are only considered to be perturbations to the rigid-body motion and so the trajectory is generated by regarding the manipulator as rigid. An additional feedback control is designed to reduce the elastic disturbance to the nominal joint trajectory. The authors resolved

the kinematic redundancy of the rigid manipulator by choosing the joint accelerations which satisfied a given tip trajectory and globally minimized the norm of the joint rates. The ordinary differential equations describing the optimal accelerations, together with specified initial joint angles and final joint rates, gave a two-point boundary value problem. To aid convergence of the shooting method, a homotopy algorithm is used to solve simpler problems before tackling the original TPBVP.

To summarize, few researchers have tackled the problem of global trajectory optimization for flexible manipulators and only LIU AND KUJATH [1996] and YAO AND CHENG [1995a] have attempted to determine *minimal vibration* motions. Both works employed a Rayleigh-Ritz method with a small number of parameters to be optimized, only one in the first case and at most eight in the latter. It is therefore not clear in either case that the approximate solution has converged to the solution of the original functional optimization problem. The next section proposes an alternative numerical solution method which combines the advantages of both direct and indirect methods and has not previously been applied to motion planning of flexible manipulators.

2.3 Proposed Formulation

This section begins by providing specific details of the functional optimization problem posed in §2.1.1. First, the general structure of the dynamics equations for flexible manipulators is given in §2.3.1. This structure allows for some reduction in the complexity of the dynamics constraint for our particular choice of unknown control inputs. The form of the objective functional is elaborated in §2.3.2 where, in addition to the particular objective—the strain energy, the boundary conditions and computational tractability also play a role.

To solve the functional optimization problem, an indirect numerical method based on the well-known Pontryagin Maximum Principle, §2.3.3, will be applied to the specific problem, §2.3.4. Finally, §2.4 compares the advantages of the proposed method with those of the many other methods described earlier.

2.3.1 Manipulator Dynamics

Throughout this work we consider fixed-base, serial-chain flexible manipulators with revolute joints only. Although the formulation given here may be more general, this configuration is the most common for flexible manipulators. For example, we write the dynamics equations governing the motion, partitioned into the “rigid” and “elastic” part, as

$$\mathbf{M}_{rr}(\mathbf{q}_r, \mathbf{q}_e) \ddot{\mathbf{q}}_r + \mathbf{M}_{re}(\mathbf{q}_r, \mathbf{q}_e) \ddot{\mathbf{q}}_e + \mathbf{h}_r(\mathbf{q}_r, \mathbf{q}_e, \dot{\mathbf{q}}_r, \dot{\mathbf{q}}_e) = \boldsymbol{\tau}. \quad (2.13)$$

$$\mathbf{M}_{re}^T(\mathbf{q}_r, \mathbf{q}_e) \ddot{\mathbf{q}}_r + \mathbf{M}_{ee}(\mathbf{q}_r, \mathbf{q}_e) \ddot{\mathbf{q}}_e + \mathbf{K}\mathbf{q}_e + \mathbf{h}_e(\mathbf{q}_r, \mathbf{q}_e, \dot{\mathbf{q}}_r, \dot{\mathbf{q}}_e) = \mathbf{0}. \quad (2.14)$$

In these equations, \mathbf{q}_r is a column vector of m joint angles, and \mathbf{q}_e contains s elastic coordinates obtained from a finite element model of the flexible links. Also, the symmetric manipulator mass matrix \mathbf{M} is partitioned as shown, $\boldsymbol{\tau}$ represents the applied joint torques, \mathbf{K}^1 is the manipulator stiffness matrix, and the \mathbf{h}_r and \mathbf{h}_e vectors contain the nonlinear inertia terms and gravitational forces [HUGHES AND SINCARSIN, 1989].

In typical control formulations, the choice of unknown inputs $\mathbf{u} = \boldsymbol{\tau}$ is physically appropriate. For path planning purposes, \mathbf{u} can be chosen to be the position coordi-

¹Here the manipulator stiffness matrix is assumed to be constant, according to the common assumption of linear elasticity theory. This assumption is made to simplify some of the calculations, but is not necessary to employ the proposed algorithms.

nates \mathbf{q} . In the presence of non-integrable velocity constraints, the velocity $\dot{\mathbf{q}}$, of the system could be chosen [SUNDAR AND SHILLER, 1994]. However, given the dynamics constraints in our problem, Eqn. (2.14), we choose $\mathbf{u} = \ddot{\mathbf{q}}_r$. The advantage of our choice is that the *rigid* dynamics equations, Eqn. (2.13), are not necessary for the solution of this problem, but can be used to calculate the torques if desired. Such a calculation is also simpler than the converse one of calculating the accelerations given the torques since the mass matrix \mathbf{M} does not need to be inverted. Furthermore, approximations of \mathbf{u} by splines or Fourier series can be integrated analytically rather than numerically to determine \mathbf{q}_r . In accordance with the notation introduced in §2.1.1, the dynamics constraint can therefore be summarized by:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (2.15)$$

$$= \begin{pmatrix} \dot{\mathbf{q}}_r \\ \dot{\mathbf{q}}_e \\ \mathbf{u} \\ -\mathbf{M}_{ee}^{-1} (\mathbf{M}_{re}^T \mathbf{u} + \mathbf{h}_e) \end{pmatrix}. \quad (2.16)$$

where the n -dimensional state vector is defined to be

$$\mathbf{x} := (\mathbf{q}_r^T, \mathbf{q}_e^T, \dot{\mathbf{q}}_r^T, \dot{\mathbf{q}}_e^T)^T \quad (2.17)$$

for $n = 2(m + s)$ and $\mathbf{q}_r \in \mathbb{R}^m$, $\mathbf{q}_e \in \mathbb{R}^s$.

2.3.2 Objective Functional

As discussed in the literature review of §2.2.3, several objective functionals have been considered for global trajectory optimization of flexible manipulators. These have included minimization of joint rate norm [MEIROVITCH AND CHEN, 1995], joint

torque norm [DOHRMANN AND ROBINETT, 1994], time [EISLER ET AL., 1993], squared tip tracking error [TU ET AL., 1994], and maximum absolute tracking error. though “this did not perform as well” [EISLER ET AL., 1993].

For flexible manipulators, an important objective is to reduce vibration, though this may be quantified in various ways. For example, LIU AND KUJATH [1996] chose to minimize the norm of the elastic coordinates and rates

$$J = \int_0^T (\mathbf{q}_e^T \mathbf{q}_e + \dot{\mathbf{q}}_e^T \dot{\mathbf{q}}_e) dt, \quad (2.18)$$

though this expression is not dimensionally consistent. A more common choice is to minimize an energy quantity related to vibration. TORRES AND DUBOWSKY [1993] minimize the strain energy of the manipulator

$$V(\mathbf{q}_e) := \frac{1}{2} \mathbf{q}_e^T \mathbf{K} \mathbf{q}_e \quad (2.19)$$

locally but propose heuristics for global minimization. MOHRI ET AL. [1998] instead consider the so-called “vibration energy”,

$$V_e(\mathbf{q}_r, \mathbf{q}_e, \dot{\mathbf{q}}_e) := \frac{1}{2} \mathbf{q}_e^T \mathbf{K} \mathbf{q}_e + \frac{1}{2} \dot{\mathbf{q}}_e^T \mathbf{M}_{ee} \dot{\mathbf{q}}_e, \quad (2.20)$$

but only minimize the residual $V_e|_{t=T}$, which can also be expressed as a functional². YAO AND CHENG [1995a] globally minimize the vibration energies of the flexible links

$$V_b(\mathbf{q}_r, \mathbf{q}_e, \dot{\mathbf{q}}_e) := \frac{1}{2} \sum_i \left(\mathbf{q}_{i,e}^T \mathbf{K}_i \mathbf{q}_{i,e} + \frac{1}{2} \dot{\mathbf{q}}_{i,e}^T \mathbf{M}_{i,ee} \dot{\mathbf{q}}_{i,e} \right), \quad (2.21)$$

with an additional penalty on the residual vibration energy $V_b|_{t=T}$. Here, the index i varies over the flexible links and $\mathbf{q}_{i,e}$ are the finite element coordinates of link i , while $\mathbf{M}_{i,ee}$ and \mathbf{K}_i are the finite element mass and stiffness matrices, respectively.

² $V_e(T) = \int_0^T \dot{V}_e(t) dt$ for $V_e(0) = 0$.

When there are massive rigid bodies, such as a payload, outboard of the flexible links. however, this measure V_b gives similar results as the strain energy, Eqn. 2.19, since it does not include the kinetic energy of such bodies.

Although minimum time motions are often important for performance, the discontinuous torque solutions tend to excite excessive vibration. The effective total time of the motion in fact increases if residual vibration must be damped out before the task continues. Typically, the discontinuous solutions are smoothed either by explicit modification or by weighting the conflicting objectives of minimum time and minimum vibration. Our objective considers minimum vibration alone to highlight the benefits of motion planning. The primary objective we choose to optimize for motion planning of flexible manipulators is therefore the total strain energy of the maneuver. This choice allows comparison with results generated by the Coupling Map [TORRES AND DUBOWSKY, 1993].

For now we will also include penalty functions on the final state and control inputs with justification for this approach to follow. Therefore, we choose the following general expression for the cost functional:

$$J[\mathbf{u}] = \frac{1}{2} \mathbf{x}_e^T \mathbf{P} \mathbf{x}_e + \frac{1}{2} \int_0^T (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt. \quad (2.22)$$

Here, $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{n \times n}$ are constant, positive semi-definite weighting matrices on the final state error \mathbf{x}_e and the state \mathbf{x} , respectively, while $\mathbf{R} \in \mathbb{R}^{m \times m}$ is a constant, positive definite weighting matrix on the control inputs. The final state error is

$$\mathbf{x}_e := \mathbf{x}(T) - \mathbf{x}_d \quad (2.23)$$

where the desired final state \mathbf{x}_d , prescribed at the final time T , is constant for point-to-point maneuvers. Note that the penalty on the final state error could equivalently be included within the integrand but is written thus for clarity.

We now specify in detail the weighting matrices \mathbf{P} , \mathbf{Q} , and \mathbf{R} appropriate to the stated objectives. In accordance with our objective of minimizing the strain energy given by Eqn. (2.19), we choose the block diagonal matrix

$$\mathbf{Q} = \frac{1}{ET} \text{diag}(\mathbf{0}, \mathbf{K}, \mathbf{0}, \mathbf{0}) \quad (2.24)$$

where \mathbf{K} is the manipulator's constant stiffness matrix. Each term in J , Eqn. (2.22), is scaled by a constant characteristic energy value E to make the objective dimensionless. Scaling does not change the optimal solution but facilitates comparison of J values for different manipulators and trajectories. We have chosen $E = ML^2/T^2$ with M and L being the total mass and length of the manipulator respectively.

Although our primary interest lies in minimizing the strain energy, choosing the input weighting matrix \mathbf{R} to be positive definite ensures that the optimization problem is not *singular*. This concept will be clarified in §2.3.4. For now we choose

$$\mathbf{R} = \gamma T^3 \mathbf{I} \quad (2.25)$$

where $0 < \gamma \ll 1$ is a dimensionless constant and \mathbf{I} is the $m \times m$ identity matrix. To penalize errors in the final state, we choose

$$\mathbf{P} = \frac{\beta}{E} \left[\begin{array}{cccc} \alpha E \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{rr} & \mathbf{M}_{re} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{re}^T & \mathbf{M}_{ee} \end{array} \right] \Big|_{\mathbf{x}=\mathbf{x}_d}, \quad (2.26)$$

thus weighting the components of the final state error according to an associated energy, either the strain energy, the kinetic energy, or an “artificial potential energy” of the final joint angle error. The matrix \mathbf{P} is made constant by evaluating the

constituent block matrices at the desired final state $\mathbf{x} = \mathbf{x}_d$. The factors α and β are positive, dimensionless constants.

The three terms of the cost J in Eqn. (2.22) represent a penalty on the final state, and the integrals of the strain energy and penalty on the inputs, respectively. Thus, β represents the relative weighting assigned to achieving the final state versus minimizing the strain energy en route. Similarly, α weights the “artificial potential energy” of the final joint angle error relative to the physical energies associated with the other final states. Note that the weight factor γ is chosen to be very small, but not zero, so that the optimization problem is not singular, yet the input costs remain insignificant relative to the strain energy and final state costs.

2.3.3 The Pontryagin Maximum Principle

In this section, we state the well-known Pontryagin Maximum Principle (PMP). The PMP provides first-order necessary conditions for an optimal control. The statement given here will be slightly more general than we will need, though not as general as possible. The next section applies the PMP to our particular problem.

The set of admissible controls is

$$\mathcal{U} := \{\mathbf{u}(\cdot) \mid \mathbf{u}(\cdot) \text{ is piecewise continuous, } \mathbf{u}(t) \in \Psi \text{ for } 0 \leq t \leq T\} \quad (2.27)$$

for a given fixed, bounded set Ψ in \mathbb{R}^m . Next, define the new state

$$x^0(t) = \int_0^t f^0(\mathbf{x}(s), \mathbf{u}(s)) ds, \quad (2.28)$$

the augmented state $\hat{\mathbf{x}}(t) := (x^0(t), \mathbf{x}^T(t))^T$, and the augmented dynamics $\hat{\mathbf{f}}(\hat{\mathbf{x}}, \mathbf{u}) := (f^0, \mathbf{f}^T)^T$. The problem can now be stated as:

Find an admissible control $\mathbf{u}(\cdot)$ such that the solution of

$$\dot{\hat{\mathbf{x}}} = \hat{\mathbf{f}}(\hat{\mathbf{x}}(t), \mathbf{u}(t)), \quad \hat{\mathbf{x}}[0] = \begin{bmatrix} 0 \\ \mathbf{x}_0 \end{bmatrix} \quad (2.29)$$

minimizes $x^0(T)$.

Now we state the following necessary conditions for an optimal control for the given problem [MACKI AND STRAUSS, 1982; ATHANS AND FALB, 1966; PINCH, 1993].

Theorem 1 (Pontryagin Maximum Principle) *Suppose $\mathbf{u}^* \in \mathcal{U}$ and $\hat{\mathbf{x}}^*$ are optimal for the problem just stated. Then there exists a non-trivial, continuous vector $\hat{\boldsymbol{\lambda}}$ satisfying*

$$\dot{\hat{\boldsymbol{\lambda}}}^T = -\frac{\partial H}{\partial \hat{\mathbf{x}}} \quad (2.30)$$

and a scalar function

$$H(\hat{\mathbf{x}}, \hat{\boldsymbol{\lambda}}, \mathbf{u}) := \hat{\boldsymbol{\lambda}}^T \hat{\mathbf{f}} \quad (2.31)$$

such that, partitioning $\hat{\boldsymbol{\lambda}}$ as $(\lambda^0, \boldsymbol{\lambda}^T)^T$,

1. for every $t \in [0, T]$,

$$H(\hat{\mathbf{x}}^*(t), \hat{\boldsymbol{\lambda}}^*(t), \mathbf{u}^*(t)) = \sup_{\mathbf{u} \in \mathcal{U}} H(\hat{\mathbf{x}}^*(t), \hat{\boldsymbol{\lambda}}^*(t), \mathbf{u}) \quad (2.32)$$

where $\hat{\boldsymbol{\lambda}}^*(t)$ is the solution of (2.30) for $\mathbf{u} = \mathbf{u}^*(t)$.

2. $\lambda^0(T) \leq 0$.

3. $\lambda^0(t)$ is constant and $H(\hat{\mathbf{x}}^*(t), \hat{\boldsymbol{\lambda}}^*(t), \mathbf{u}^*(t))$ is constant on $[0, T]$.

4. $\boldsymbol{\lambda}(T) = \mathbf{0}$.

Note that this statement of the PMP has been restricted by the choice of \mathcal{U} , and by the conditions that the final time T be constant and the final state $\mathbf{x}(T)$ be free. More general statements are made in the references.

2.3.4 Application of the PMP

This section restates the PMP for the particular application of motion planning for flexible manipulators, providing further detail of the problem formulation necessary for numerical solution. To apply the theorem, we will have to transform the problem specific objective functional into a suitable form.

To write our chosen objective (2.22) in the general form (2.1) we must first include the final state penalty within the integrand. Noting that

$$\rho(\mathbf{x}(T)) = \rho(\mathbf{x}(0)) + \int_0^T \frac{\partial \rho}{\partial \mathbf{x}} \dot{\mathbf{x}} dt \quad (2.33)$$

for some smooth function $\rho(\mathbf{x})$, we see that minimizing

$$J[\mathbf{u}] = \rho(\mathbf{x}(T)) + \int_0^T f^0 dt \quad (2.34)$$

is equivalent to minimizing

$$J[\mathbf{u}] = \int_0^T f^0 + \frac{\partial \rho}{\partial \mathbf{x}} \mathbf{f} dt \quad (2.35)$$

since $\rho(\mathbf{x}(0))$ is a known constant. Now to apply the PMP we form the scalar function defined by Eqn. (2.31)

$$H(\mathbf{x}, \hat{\boldsymbol{\lambda}}, \mathbf{u}) = \hat{\boldsymbol{\lambda}}^T \hat{\mathbf{f}} \quad (2.36)$$

$$= \lambda^0 \left(f^0 + \frac{\partial \rho}{\partial \mathbf{x}} \mathbf{f} \right) + \boldsymbol{\lambda}^T \mathbf{f} \quad (2.37)$$

$$= \lambda^0 f^0 + \left(\lambda^0 \frac{\partial \rho}{\partial \mathbf{x}} + \boldsymbol{\lambda}^T \right) \mathbf{f} \quad (2.38)$$

where we have written the argument \mathbf{x} rather than $\hat{\mathbf{x}}$ since H is not a function of x^0 .

Defining a new adjoint vector $\hat{\mathbf{p}} = (p^0, \mathbf{p}^T)^T$ by

$$p^0 := -\lambda^0 \quad (2.39)$$

$$\mathbf{p}^T := -\lambda^0 \frac{\partial \rho}{\partial \mathbf{x}} - \boldsymbol{\lambda}^T \quad (2.40)$$

leads us to define a new scalar function

$$H'(\mathbf{x}, \hat{\mathbf{p}}, \mathbf{u}) := \hat{\mathbf{p}}^T \hat{\mathbf{f}} \quad (2.41)$$

$$= p^0 f^0 + \mathbf{p}^T \mathbf{f} \quad (2.42)$$

$$= -H(\mathbf{x}, \hat{\boldsymbol{\lambda}}, \mathbf{u}). \quad (2.43)$$

Note that, for numerical purposes, we choose to minimize H' rather than maximize H .

Now we summarize and apply the PMP for our particular objective of minimizing (2.22). First, since we are not considering bounds on the inputs, we will only look for solutions $\mathbf{u}^*(t)$ whose values remain interior to the set Ψ (appearing in the definition of the set of admissible controls \mathcal{U} , page 38) and at the same time consider Ψ to be arbitrarily large. In this case, the necessary conditions now apply only to local maxima of H rather than to global suprema. Similarly, the constant λ^0 cannot be 0 or no local maxima of H will exist. Since the adjoint equations (2.30) are linear, any scalar multiple of a solution is also a solution, so we are free to choose $\lambda^0 = -1 \Rightarrow p^0 = 1$. Finally, we drop the ' on H and restate the PMP as:

Theorem 2 *Suppose $\mathbf{u}^* \in \mathcal{U}$ and $\hat{\mathbf{x}}^*$ are optimal in the sense discussed. Then there exists a scalar function*

$$H(\mathbf{x}, \mathbf{p}, \mathbf{u}) = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \frac{1}{2} \mathbf{u}^T \mathbf{R} \mathbf{u} + \mathbf{p}^T \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (2.44)$$

and a non-trivial, continuous vector \mathbf{p} satisfying

$$\dot{\mathbf{p}}^T = -\frac{\partial H}{\partial \mathbf{x}} \quad (2.45)$$

$$= -\mathbf{x}^T \mathbf{Q} - \mathbf{p}^T \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \quad (2.46)$$

such that,

1. for every $t \in [0, T]$,

$$H(\mathbf{x}^*(t), \mathbf{p}^*(t), \mathbf{u}^*(t)) = \min_{\mathbf{u} \in \mathcal{U}} H(\mathbf{x}^*(t), \mathbf{p}^*(t), \mathbf{u}) \quad (2.47)$$

where $\mathbf{p}^*(t)$ is the solution of (2.46) for $\mathbf{u} = \mathbf{u}^*(t)$.

2. $H(\mathbf{x}^*(t), \hat{\mathbf{p}}^*(t), \mathbf{u}^*(t))$ is constant on $[0, T]$.

3.

$$\mathbf{p}(T) = \mathbf{P}\mathbf{x}_e \quad (2.48)$$

Note that, since we are only seeking solutions $\mathbf{u}^*(t)$ taking values interior to Ψ , conditions sufficient to satisfy Eqn. (2.47) are

$$\frac{\partial H}{\partial \mathbf{u}} = \mathbf{0} \quad \text{and} \quad \frac{\partial^2 H}{\partial \mathbf{u}^2} > 0, \quad \forall t \in [0, T], \quad (2.49)$$

which simplify to

$$\mathbf{u}^T \mathbf{R} + \mathbf{p}^T \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} = \mathbf{0} \quad \text{and} \quad \mathbf{R} > 0, \quad \forall t \in [0, T]. \quad (2.50)$$

A control problem is singular if the first-order necessary conditions embodied by the PMP do not explicitly determine \mathbf{u} as a single-valued function of \mathbf{x} and \mathbf{p} [MACKI AND STRAUSS, 1982]³. However, the inclusion of a penalty term on the control inputs weighted by a symmetric, positive definite matrix \mathbf{R} allows us to solve Eqn. (2.50) and write

$$\mathbf{u}^* = -\mathbf{R}^{-1} \left[\begin{array}{c} 0 \\ 0 \\ \mathbf{I} \\ -\mathbf{M}_{ee}^{-1} \mathbf{M}_{re}^T \end{array} \right] \Big|_{\mathbf{x}=\mathbf{x}^*} \mathbf{p}^*. \quad (2.51)$$

³See BELL AND JACOBSON [1975] for higher-order necessary conditions for singular optimal control problems.

Thus, there exists a solution \mathbf{u}^* which, at least locally, minimizes H pointwise, which in turn is necessary to minimize the objective functional J .

As is the case for indirect methods for functional optimization, we have now derived a TPBVP whose solution represents a necessary condition for the optimum. The ODE's of the boundary value problem are represented by the state dynamics, Eqn. (2.16), and the adjoint or *costate* dynamics, Eqn. (2.46). The boundary conditions are the given initial value of the state, \mathbf{x}_0 , and the final value of the costate, Eqn. (2.48). Note that the algebraic optimality condition, Eqn. (2.51), must also be satisfied by the solution at each point in time.

2.3.5 Comparison to Other Methods

Since the proposed formulation does not correspond directly to any of the many functional optimization methods discussed in §2.2, it is helpful here to relate the proposed formulation to those methods. In the process, this brief comparison will help to justify our choice of formulation.

One of our specifications for an optimal motion planning algorithm (§1.2) is that the dynamics equations and their solution not be approximated, except to within numerical integration tolerance. A straight-forward application of the Rayleigh-Ritz method and its variations, including finite elements, is therefore not suitable since these methods approximate the states by a series expansion. However, as pointed out in §2.2.3, it is possible to satisfy the dynamics identically by expanding only $m < n$ of the states (*e.g.* $\mathbf{q}_r \in \mathbb{R}^m$) as a series and integrating the elastic dynamics, Eqn. (2.14), to determine the remaining $n - m$ states. The dependence of the elastic states on the series parameters is therefore not known explicitly. Hence, the depen-

dence of the objective functional on the series parameters is also not known explicitly and optimization must proceed either by direct search or by utilizing a numerical approximation of the gradient of the functional. Dynamic programming, because of its recursive nature, could satisfy the dynamics by employing a suitable functional optimization method between each pair of discrete states. However, the large number of states required for flexible manipulator dynamics and the resulting computational complexity make the required amount of computation prohibitive.

At this point we turn our attention to indirect approaches. These methods reformulate the optimization problem as a TPBVP, as is the case for the proposed formulation. Recall that the TPBVP has a particular structure, consisting of the state dynamics, the costate dynamics, and an algebraic condition for the optimal inputs. This structure allows the state dynamics to be numerically integrated as an initial value problem while a different approximation scheme is employed to solve the costate dynamics.

A straightforward numerical approach to solving this TPBVP would be to substitute the algebraic solution for the optimal inputs into both the state and costate dynamics and employ a shooting method. In this case, both the optimality condition and the ODE's are satisfied at every iteration, while the optimal boundary conditions are only determined in the limit. A numerical root-finding algorithm would update the initial value of the costate until all boundary conditions were satisfied. However, as mentioned in §2.2.2, shooting methods are particularly sensitive to initial guesses and can result in instability, requiring the addition of a continuation algorithm to ensure convergence to the optimal solution.

Instead, we choose to satisfy the boundary conditions at every iteration and converge to the optimal input. This approach entails three conceptual changes to the

shooting method scenario. First, the state dynamics are integrated alone, separate from the costate dynamics. The integration requires an estimate of the inputs as a function of time or, if the algebraic optimality condition has been substituted, of the costate. The second change is to retain the inputs in the state dynamics explicitly. Thus, the inputs will be estimated and updated by the optimization rather than the costate. We make this choice because the inputs have physical meaning (joint accelerations), which enables a more reasonable initial guess, and because the vector dimension is reduced at least in half (from $2m + 2s$ to m). Finally, rather than having boundary conditions on both the initial and final state and none on the costate, the third change is to place a penalty function on the final state, allowing the final boundary conditions to be transferred to the costate instead. Now boundary conditions are available which allow the costate dynamics to be determined by numerical integration once the state and inputs are known.

In the next chapter we use a method which combines the advantages of both the direct and indirect methods to determine the optimal inputs. After expanding the inputs in terms of cubic spline basis functions (as for the direct methods), the algebraic optimality condition provides the gradient for minimization of J as a function of the parameterized inputs \mathbf{u} (as for the indirect methods). Utilizing gradient information explicitly provides a distinct computational advantage over the aforementioned direct Rayleigh-Ritz based methods previously employed in the literature [YAO AND CHENG, 1995A]. A nonlinear numerical minimization algorithm then determines the expansion coefficients which minimize J .

2.4 Summary

This chapter has formulated a functional optimization problem appropriate for motion planning for flexible manipulators. Beginning with a mathematical definition of the problem, an extensive review of the available solution methods was conducted. A literature survey illustrated the application of these methods, particularly to problems involving flexible manipulators, and concluded that very little work has been done to determine motions which minimize vibration in flexible manipulators. The heart of this chapter was the detailed formulation of a functional optimization method based on the Pontryagin Maximum Principle. This formulation was related to the other available methods and was justified with respect to the structure and goals of the particular problem under consideration. Furthermore, the proposed formulation possesses computational advantages over methods previously employed for this problem.

Chapter 3

Numerical Solution

3.1 Introduction

This chapter presents the numerical implementation of the proposed functional optimization method. The first section completes the formulation proposed in the previous chapter by detailing the numerical interpolation of the inputs, integration of the state and costate dynamics, and finally optimization of the inputs. To verify the implementation, another method for motion planning of flexible manipulators, the Coupling Map, is summarized in §3.3. Finally, §3.4 presents extensive optimization results for three example manipulators. The particular configurations and maneuvers of these manipulators were chosen specifically to help verify the optimization. The optimal solutions are justified physically and compared with the Coupling Map.

3.2 Numerical Implementation

Before presenting the numerical solution procedure for determining the optimal motion of a flexible manipulator, we first summarize the problem as formulated at the end of the previous chapter. From the functional optimization problem presented in §2.1.1 to minimize the objective of Eqn. (2.22),

$$J[\mathbf{u}] = \frac{1}{2} \mathbf{x}_e^T \mathbf{P} \mathbf{x}_e + \frac{1}{2} \int_0^T (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt, \quad (3.1)$$

the following necessary conditions have been derived:

1. From Eqns. (2.44) and (2.47), find

$$\min_{\mathbf{u} \in \mathcal{U}} H(\mathbf{x}, \mathbf{p}, \mathbf{u}) = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \frac{1}{2} \mathbf{u}^T \mathbf{R} \mathbf{u} + \mathbf{p}^T \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \forall t \in [0, T] \quad (3.2)$$

from which the necessary condition (2.50)

$$\frac{\partial H}{\partial \mathbf{u}} = \mathbf{u}^T \mathbf{R} + \mathbf{p}^T \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} = 0 \quad (3.3)$$

follows for optimal solutions interior to \mathcal{U} .

2. State dynamics from Eqn. (2.16)

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (3.4)$$

3. Costate dynamics from Eqn. (2.46)

$$\dot{\mathbf{p}} = -\mathbf{Q} \mathbf{x} - \left(\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right)^T \mathbf{p} \quad (3.5)$$

4. Boundary conditions from Eqns. (2.3), (2.23), and (2.48)

$$\mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{p}(T) = \mathbf{P} \mathbf{x}_e. \quad (3.6)$$

The approach we follow to approximate the solution can be roughly outlined as follows. First, the input function \mathbf{u} is parameterized by a linear combination of a finite number of basis functions. An initial guess for \mathbf{u} then allows the state dynamics, Eqn. (3.4), to be integrated forward in time from the given initial condition. Knowing both \mathbf{x} and \mathbf{u} , the costate dynamics, Eqn. (3.5), are integrated backwards in time from the given final condition, Eqn. (3.6). Finally, the objective J is evaluated and its gradient with respect to the parameters of \mathbf{u} is determined from Eqn. (3.3). With this information, a nonlinear optimization routine updates the approximation of \mathbf{u} and the cycle repeats until convergence.

The remainder of this section provides specific details of each of the above steps: first the approximation and interpolation of \mathbf{u} , then the numerical integration for \mathbf{x} and \mathbf{p} , and finally the numerical minimization of J .

3.2.1 Spline Interpolation

To solve infinite dimensional optimization problems numerically, the problem must be approximated by a finite dimensional optimization problem (or a finite series of such problems). We choose to accomplish this by selecting a finite set of times in the given time interval and representing \mathbf{u} by its values at those times. The value of \mathbf{u} at any other time is obtained by interpolation, which we now describe.

The set of N times is chosen uniformly from the given time interval $[0, T]$ and is denoted by

$$t_{\Sigma} := \{t_i | t_i = ih, i = 1, \dots, N, h := T/(N + 1)\}. \quad (3.7)$$

The points t_i are known as *knot* points, or simply knots [PRENTER, 1975]. We seek to optimize the values $\mathbf{u}_i := \mathbf{u}(t_i)$. Note that t_{Σ} does not include the end-points of

the interval, $t_0 = 0$ and $t_{N+1} = T$, since we have constrained $\mathbf{u}(0) = \mathbf{u}(T) = \mathbf{0}$ to enforce continuity¹.

For both theoretical and practical reasons, the approximate function \mathbf{u} must be defined on the entire interval $[0, T]$. We therefore approximate \mathbf{u} by the *interpolatory cubic spline* $\mathbf{s}_N(t)$. This cubic spline is the unique piecewise cubic polynomial that satisfies the following conditions [PRENTER, 1975]:

1. $\mathbf{s}_N(t_i) = \mathbf{u}_i$, for $i = 0, \dots, N + 1$,
2. $\mathbf{s}_N(t)$ is twice continuously differentiable on $[0, T]$, and
3. $\mathbf{s}_N(t)$ is “clamped” at the end-points: $\dot{\mathbf{s}}_N(0) = \dot{\mathbf{s}}_N(T) = \mathbf{0}$.

Within the context of the Rayleigh-Ritz method described in §2.2.1, it is significant from a theoretical perspective that the interpolatory cubic spline $\mathbf{s}_N(t)$ satisfying the above conditions can be expressed as a linear combination of $N + 4$ independent *B-splines* [PRENTER, 1975]. We modify the standard B-splines slightly to derive N basis functions ϕ_i which satisfy condition 3 above and $\mathbf{u}_0 = \mathbf{u}_{N+1} = \mathbf{0}$. The interpolatory spline is then expressed as

$$\mathbf{s}_N(t) = \sum_{i=1}^N \mathbf{a}_i \phi_i(t). \quad (3.8)$$

Whenever the parameters \mathbf{u}_i are updated by the numerical optimization, the linear algebraic equations

$$\mathbf{U} = \mathbf{BA} \quad (3.9)$$

¹Such boundary conditions may arise when the objective contains higher-order time derivatives of \mathbf{u} [BHAT AND MIU, 1992].

must be solved for the expansion coefficients $\mathbf{A} \in \mathbb{R}^{N \times m}$, where the rows of $\mathbf{U} \in \mathbb{R}^{N \times m}$ are $\mathbf{u}_i = \mathbf{u}(t_i)$ and the values of the B-splines $\phi_j(t_i)$ are the elements B_{ij} of the full-rank matrix $\mathbf{B} \in \mathbb{R}^{N \times N}$ [PRENTER, 1975].

As $N \rightarrow \infty$, it is possible to choose an infinite set of independent B-splines which is complete² in the space of twice continuously differentiable functions that satisfy the above boundary conditions [SCHULTZ, 1973]. This in turn implies that the interpolants will converge to a solution which minimizes the objective functional. that is, $\lim_{N \rightarrow \infty} J[\mathbf{s}_N] = \inf_{\mathbf{u} \in \mathcal{U}} J[\mathbf{u}]$ [GELFAND AND FOMIN, 1963].

3.2.2 Numerical Integration

Now that the input function \mathbf{u} has been approximated, the state \mathbf{x} and costate \mathbf{p} are determined by numerical integration. From the given initial condition \mathbf{x}_0 , the state dynamics, Eqn. (3.4), are integrated forward in time using a widely available numerical integration package called CVODE [COHEN AND HINDMARSH, 1994]. The error between the calculated value $\mathbf{x}(T)$ and the desired final state provides the final value of the costate, $\mathbf{p}(T)$, according to Eqn. 3.6.

All the necessary information is now available to determine the costate \mathbf{p} . From the final value $\mathbf{p}(T)$, the costate dynamics, Eqn. (3.5), are numerically integrated backwards in time. To accommodate a variable stepsize numerical integration scheme, however, the values of $\mathbf{x}(t)$ and $\mathbf{u}(t)$ must be known for any $t \in [0, T]$. The inputs \mathbf{u} are of course obtained from the aforementioned cubic spline interpolation. The states \mathbf{x} are obtained by saving $\mathbf{x}_i := \mathbf{x}(t_i)$, $t_i \in t_\Sigma$ during the state integration and interpolating as needed during the costate integration.

²See Definition 1, page 19, for the definition of a complete set.

To integrate the costate dynamics, we also require an expression for the Jacobian of the state dynamics, $\partial\mathbf{f}/\partial\mathbf{x}$. We have chosen to approximate this Jacobian by finite differences because of the complexity of the dynamics model. Although each evaluation of the Jacobian may be computationally expensive, the number of evaluations can be reduced by observing the following relationship characteristic of adjoint systems such as the costate dynamics. If we write the costate dynamics equations (3.5) as $\dot{\mathbf{p}} = \mathbf{g}(\mathbf{x}, \mathbf{p}, \mathbf{u})$, then the Jacobian of the costate dynamics, $\partial\mathbf{g}/\partial\mathbf{p}$, is related to the state Jacobian by

$$\frac{\partial\mathbf{g}(\mathbf{x}, \mathbf{p}, \mathbf{u})}{\partial\mathbf{p}} = - \left(\frac{\partial\mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial\mathbf{x}} \right)^T. \quad (3.10)$$

At each timestep of the numerical integration of the costate dynamics, CVODE will have to evaluate \mathbf{g} , and therefore also $\partial\mathbf{f}/\partial\mathbf{x}$. However, numerical integration packages such as CVODE permit the user to supply a routine to calculate the Jacobian in this case $\partial\mathbf{g}/\partial\mathbf{p}$. By providing such a routine, the Jacobian is not updated by CVODE at every timestep, but only as needed based on local integration error estimates. Noting that the same state Jacobian $\partial\mathbf{f}/\partial\mathbf{x}$ appears in both Eqn. (3.10) and Eqn. (3.5), it follows that fewer costate Jacobian updates leads to fewer state Jacobian evaluations when calculating \mathbf{g} .

3.2.3 Nonlinear Optimization

Finally we discuss the central aspect of the numerical implementation of our motion planning formulation, optimization. Recall that in the last chapter the motion planning problem was formulated to minimize the functional $J[\mathbf{u}]$. Necessary conditions for optimality were then derived as summarized in §3.2. Our approach is to satisfy identically three of the four necessary conditions represented by Eqns. (3.4-3.6), as

would be done for an indirect method. However, subject to those conditions, we then perform a direct minimization of J as a function of the parameterized inputs. For a nominal input \mathbf{u} , the value of the functional J is obtained by simulation of the state dynamics as described in the previous section. We now show that the gradient of J with respect to the input parameters is related to the remaining necessary condition Eqn. (3.3).

At this point in the analysis we assume the inputs $\mathbf{u}(t)$ are parameterized by the cubic spline interpolation as given in Eqn. 3.8. We wish to determine the partial derivatives of the objective J with respect to the parameters $\mathbf{u}_i = \mathbf{u}(t_i)$. Consider a perturbation $\delta\mathbf{u}$ of \mathbf{u} of the form

$$\delta\mathbf{u}(t) = \sum_{i=1}^N \Delta\mathbf{a}_i \phi_i(t) \quad (3.11)$$

or in component form

$$\delta u_k(t) = \sum_{i=1}^N \Delta A_{ik} \phi_i(t), \quad k = 1, \dots, m \quad (3.12)$$

$$= \sum_{i=1}^N \sum_{j=1}^N B_{ij}^{-1} \Delta U_{jk} \phi_i(t) \quad (3.13)$$

using Eqn. 3.9 where B_{ij}^{-1} are the elements of \mathbf{B}^{-1} . The corresponding perturbation of J is

$$\Delta J := J(\mathbf{u} + \delta\mathbf{u}) - J(\mathbf{u}) \quad (3.14)$$

$$= \int_0^T \left(\frac{\partial H}{\partial \mathbf{u}} \delta\mathbf{u} \right) dt + O(\|\Delta\mathbf{A}\|^2) \quad (3.15)$$

$$= \int_0^T \left(\sum_{k=1}^m \frac{\partial H}{\partial u_k} \sum_{i=1}^N \sum_{j=1}^N B_{ij}^{-1} \Delta U_{jk} \phi_i(t) \right) dt + O(\|\Delta\mathbf{U}\|^2) \quad (3.16)$$

under the conditions that Eqns. (3.4-3.6) are satisfied identically and that $\partial H/\partial \mathbf{u}$ is

evaluated at \mathbf{u} . We can then write the required partial derivatives as

$$\frac{\partial J}{\partial U_{jk}} = \int_0^T \left(\frac{\partial H}{\partial u_k} \sum_{i=1}^N B_{ij}^{-1} \phi_i(t) \right) dt, \quad j = 1, \dots, N, \quad k = 1, \dots, m. \quad (3.17)$$

Discretizing the time interval into $M + 2$ equally spaced points and expanding the integral using the trapezoidal rule, we get

$$\frac{\partial J}{\partial U_{jk}} = \lim_{M \rightarrow \infty} \frac{\Delta t}{2} \sum_{r=0}^M \left(\left[\frac{\partial H}{\partial u_k} \sum_{i=1}^N B_{ij}^{-1} \phi_i \right]_{t=r\Delta t} + \left[\frac{\partial H}{\partial u_k} \sum_{i=1}^N B_{ij}^{-1} \phi_i \right]_{t=(r+1)\Delta t} \right) \quad (3.18)$$

$$= \lim_{M \rightarrow \infty} \Delta t \sum_{r=1}^M \left[\frac{\partial H}{\partial u_k} \sum_{i=1}^N B_{ij}^{-1} \phi_i \right]_{t=r\Delta t} \quad (3.19)$$

where $\Delta t := T/(M + 1)$ and we have used the fact that $\phi_i = 0$ at the endpoints $t = 0$ and $t = T$ for all i . Finally, approximating M by N we get

$$\frac{\partial J}{\partial U_{jk}} \approx \Delta t \sum_{r=1}^N \sum_{i=1}^N B_{ij}^{-1} \phi_i(t_r) \frac{\partial H}{\partial u_k} \Big|_{t=t_r} \quad (3.20)$$

$$= \Delta t \sum_{r=1}^N \sum_{i=1}^N B_{ri} B_{ij}^{-1} \frac{\partial H}{\partial u_k} \Big|_{t=t_r} \quad (3.21)$$

$$= \Delta t \sum_{r=1}^N \delta_{rj} \frac{\partial H}{\partial u_k} \Big|_{t=t_r} \quad (3.22)$$

$$= \Delta t \frac{\partial H}{\partial u_k} \Big|_{t=t_j} \quad (3.23)$$

or

$$\frac{\partial J}{\partial u_j} \approx \Delta t \frac{\partial H}{\partial u_j}, \quad j = 1, \dots, N. \quad (3.24)$$

Thus the gradient of the objective J with respect to the input parameters \mathbf{u}_i is proportional to the gradient of H as $N \rightarrow \infty$. This result agrees with the infinite dimensional case for which it can be shown that the bounded variation $\delta \mathbf{u}$ which minimizes

$$\delta J = \int_0^T \left(\frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u} \right) dt \quad (3.25)$$

is proportional to $\partial H/\partial \mathbf{u}$ [MUFTI, 1970; BRYSON AND HO, 1975].

For the purposes of the finite dimensional numerical optimization, the gradient of J is approximated by the gradient of H . The latter is determined from the values of the states and costates at each knot point according to Eqn. (2.50) evaluated only at the knot points:

$$\frac{\partial H}{\partial \mathbf{u}_i} = \mathbf{u}_i^T \mathbf{R} + \mathbf{p}_i^T \frac{\partial f(\mathbf{x}_i, \mathbf{u}_i)}{\partial \mathbf{u}_i}. \quad (3.26)$$

Using this expression, the remaining necessary condition, $\partial H/\partial \mathbf{u} = 0$. Eqn. (3.3). is satisfied at the knot points upon convergence of the numerical optimization.

To implement the numerical optimization, the input vectors \mathbf{u}_i , $i = 1, \dots, N$, are assembled into a single vector \mathbf{u}_Σ . With the function value $J(\mathbf{u}_\Sigma)$ and its gradient $\partial J/\partial \mathbf{u}_\Sigma$ available, determining the optimal value of \mathbf{u}_Σ is now a standard nonlinear multivariable optimization problem. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton numerical optimization scheme, including a line search algorithm, is used to determine the optimal joint accelerations \mathbf{u}_Σ [PRESS ET AL., 1991]. The organization of the numerical optimization routines is shown in the main blocks of Fig. 3.1. The search direction vector \mathbf{d} is initially the negative gradient and is updated according to the BFGS algorithm. The user specifies the parameter MAXITERS, the maximum number of iterations, and the convergence tolerances tol and $gtol$. For all of the numerical results presented, convergence was obtained after a restart for $tol = 4\epsilon$, where ϵ is the machine precision. The restart was added to the algorithm to aid verification of the algorithm and is described in the next section. Finally it should be mentioned that, depending on the initial guess for the knot points, computation time was typically many hours and sometimes up to a few days on a Sun 5 system.

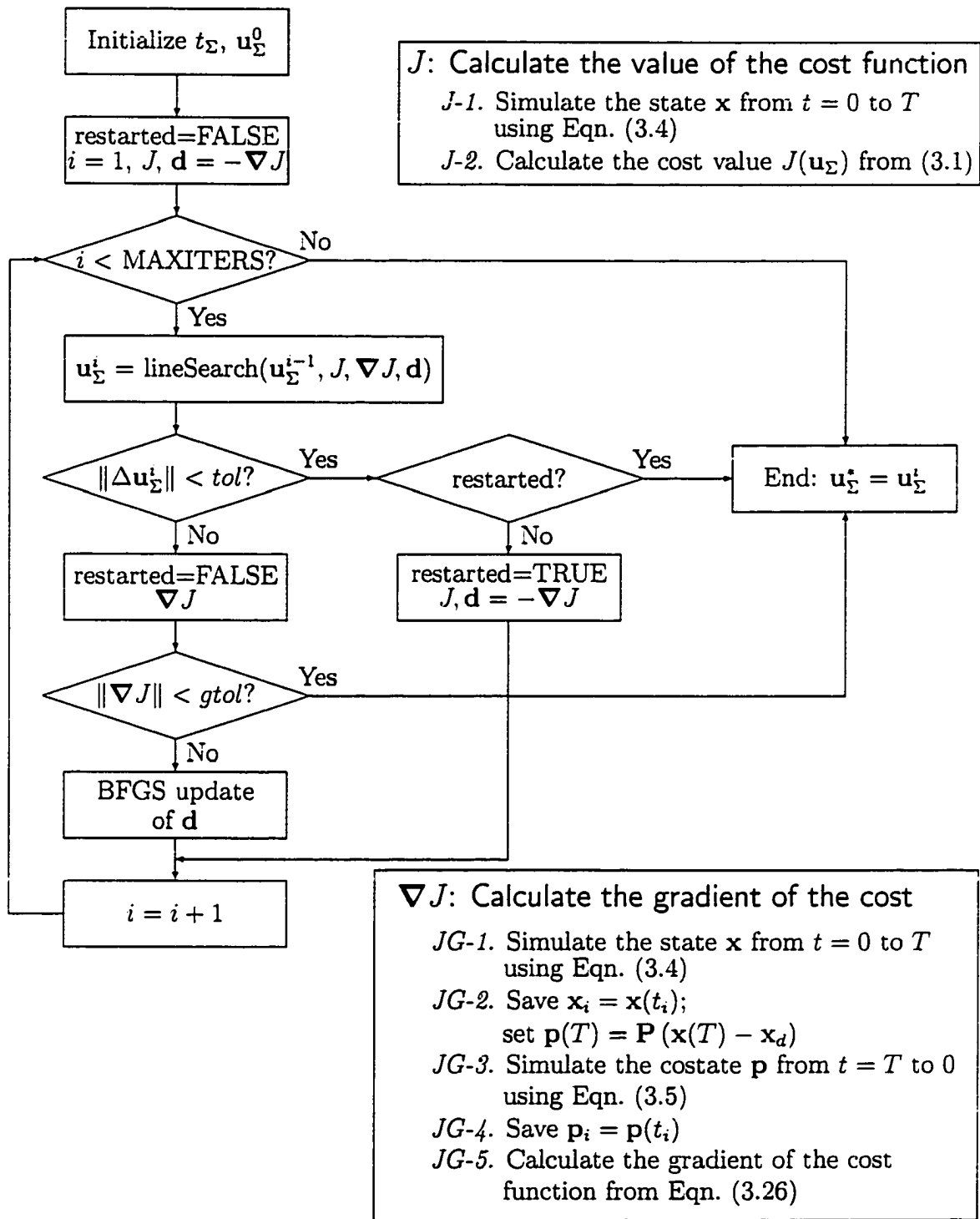


Figure 3.1: The Quasi-Newton minimization algorithm, incorporating a line-search and restart, with function value and gradient subroutines.

3.2.4 Verification

To have some degree of confidence in the accuracy of the numerical solutions provided by the proposed algorithm, its computer implementation must be verified. The verification consists of several parts, addressing the individual subroutines for the algorithms described above, as well as the final output of the assembled routine as a whole.

One of the reasons for choosing the particular functional optimization formulation proposed here is that the existing dynamics simulation code remains intact. A previous independent verification of this code [STANWAY, 1996] is therefore preserved. Our only modification was to allow the joint accelerations to be specified as the inputs, rather than the joint torques. Instead, the joint torques are calculated from the dynamics model to allow an energy check. During the simulation, the energy check verifies that the energy input to the system by the torques is accounted for in the state of the system. This check has proven to be sensitive to modelling and computational errors.

With confidence in the simulation comes confidence in the value of the objective function. The optimization process is verified in part by observing a decrease in the objective after each step. Occasionally, the optimization would fail to find a decrease in a particular search direction. Although the gradient direction gives the most decrease in a single step and is the initial search direction, subsequent search directions are chosen by the quasi-Newton optimization algorithm to speed convergence over many iterations and are generally not in the gradient direction. Therefore, an automatic restart was incorporated to verify that no improvement could be made in the estimated direction of the gradient. The algorithm halts after an unsuccessful restart.

To improve the accuracy of the gradient estimate, as well as the interpolation of the inputs, more knot points are added. The increased accuracy requires increased computation, however, so a balance is sought. If more points are added and the previously existing solution parameters change little after subsequent optimization, then the solution is assumed to have converged. Such tests were performed to verify both the solution and the algorithm.

A more meaningful verification of the optimization is possible by comparing the results with other methods. The next section presents a formulation of the Coupling Map, a graphical tool for planning motions of flexible manipulators [TORRES AND DUBOWSKY, 1993]. In §3.4, trajectories generated by optimization are evaluated with respect to the heuristics provided by the Coupling Map. Both the candidate solutions and the Coupling Map heuristics are also interpreted physically. To facilitate this verification, example manipulators and point-to-point motion tasks are chosen to give nontrivial trajectories which are physically justified, such as motions with zero vibration.

3.3 The Coupling Map

A brief development of the Coupling Map is presented here, following the original detailed derivation in TORRES AND DUBOWSKY [1993]. In that article, Torres and Dubowsky consider free-floating systems consisting of a rigid manipulator attached to a massless flexible structure. The inertial coordinates of the attachment point, also referred to as the manipulator base degrees of freedom, are represented by $\phi \in \mathbb{R}^6$; the manipulator joint angles are $\mathbf{q}_r \in \mathbb{R}^m$ as before.

In addition to the massless flexible structure requirement, the derivation in TOR-

TORRES AND DUBOWSKY [1993] is based on the following critical assumptions:

1. The manipulator begins its motion from rest and does not interact with the environment.
2. Internal forces between the flexible structure and the manipulator are small and gravitational forces are negligible.
3. The manipulator does not excite resonance of the flexible structure.

With the above assumptions, conservation of generalized momenta can be assumed, that is

$$\mathbf{p} = \mathbf{M}\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{M}_{\phi\phi} & \mathbf{M}_{\phi r} \\ \mathbf{M}_{\phi r}^T & \mathbf{M}_{rr} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\mathbf{q}}_r \end{bmatrix} \approx \mathbf{0} \quad (3.27)$$

where \mathbf{M} is the appropriate system mass matrix. In TORRES AND DUBOWSKY [1993], the first equation above is used to establish a relationship between small manipulator joint motion and the resulting base displacement as:

$$\delta\phi = -\mathbf{M}_{\phi\phi}^{-1}\mathbf{M}_{\phi r}\delta\mathbf{q}_r \quad (3.28)$$

The next part of the development involves formulating the strain energy of the elastic structure as a function of the manipulator degrees of freedom. To this end, it is stated in TORRES AND DUBOWSKY [1993] that

$$\tau_\phi \sim \delta\phi \quad (3.29)$$

and a standard definition of the strain energy V of the supporting structure is used:

$$V = \frac{1}{2}\tau_\phi^T \mathbf{K}_\phi^{-1} \tau_\phi \quad (3.30)$$

where $\mathbf{K}_\phi \in \mathbb{R}^{6 \times 6}$ is a symmetric stiffness matrix. Combining Eqns. (3.30) and (3.29) with the momentum conservation relation (3.28) gives the desired form of V :

$$V \sim \delta\phi^T \mathbf{K}_\phi^{-1} \delta\phi \quad (3.31)$$

$$\sim \delta\mathbf{q}_r^T \mathbf{Q} \delta\mathbf{q}_r \quad (3.32)$$

where \mathbf{Q} is the *coupling matrix* defined as:

$$\mathbf{Q} := \mathbf{M}_{\phi r}^T \mathbf{M}_{\phi\phi}^{-1} \mathbf{K}_\phi^{-1} \mathbf{M}_{\phi\phi}^{-1} \mathbf{M}_{\phi r}. \quad (3.33)$$

The largest eigenvalue of the coupling matrix \mathbf{Q} indicates the degree of coupling between manipulator joint motions and the strain energy induced in the flexible structure. The eigenvector associated with the minimum eigenvalue provides a direction of least coupling which is useful for path planning. Thus, the Coupling Map is a graphical representation of regions of high relative coupling, indicated by darker shading, superimposed by lines showing the directions of least coupling.

We can extend the Coupling Map concept to general flexible manipulators by representing the vibration by the elastic coordinates $\mathbf{q}_e \in \mathbb{R}^s$ rather than the inertial coordinates of a particular point $\phi \in \mathbb{R}^6$. Partitioning the flexible manipulator mass matrix as in Eqns. (2.13,2.14), we follow the same derivation as above to get the new coupling matrix

$$\tilde{\mathbf{Q}} := \mathbf{M}_{re} \mathbf{M}_{ee}^{-1} \mathbf{K}^{-1} \mathbf{M}_{ee}^{-1} \mathbf{M}_{re}^T. \quad (3.34)$$

The Coupling Maps presented later are derived from this coupling matrix.

Given a Coupling Map for a particular manipulator, a joint path between two configurations to reduce vibration can be chosen heuristically. The Coupling Map heuristics [TORRES AND DUBOWSKY, 1993] specify motion along the minimum cou-

pling lines in areas of higher coupling (darkly shaded) and, if necessary, against the minimum coupling lines only in areas of lower coupling.

3.4 Numerical Examples

This section presents three example manipulators, REMSPATIAL, LAURA-SARA, and REMGRAVITY, along with a point-to-point motion task for each. Complete simulation results are presented for the optimal trajectory in each example, and comparisons are made to the Coupling Map. A discussion of the physical aspects of the optimal solution is included to help verify the solution.

3.4.1 REMSPATIAL

The first manipulator, REMSPATIAL, is chosen as a convenient example for which the optimal paths are nontrivial yet physically justifiable. It has two joints (two rigid degrees-of-freedom) and two links, with the first link rigid and the second link flexible with planar bending only (see Figure 3.2). A steel sphere at the tip of the flexible link acts as the payload. Table 3.1 lists the physical properties of the bodies. The flexible link is modelled by a single beam element, giving two elastic coordinates, the deflection and slope of the tip of the link.

Rotation of the second joint allows any orientation of the bending plane relative to the rotation plane of the first link. In particular, the bending plane of the flexible link can be oriented to decouple the link vibration from motion of the first joint. This configuration thus helps verify motion planning schemes by providing an intuitive solution for trajectories with minimal vibration excitation.

The task is to move the joints from $(-90^\circ, 0^\circ)$ to $(90^\circ, 0^\circ)$, rest to rest, in 5 seconds.

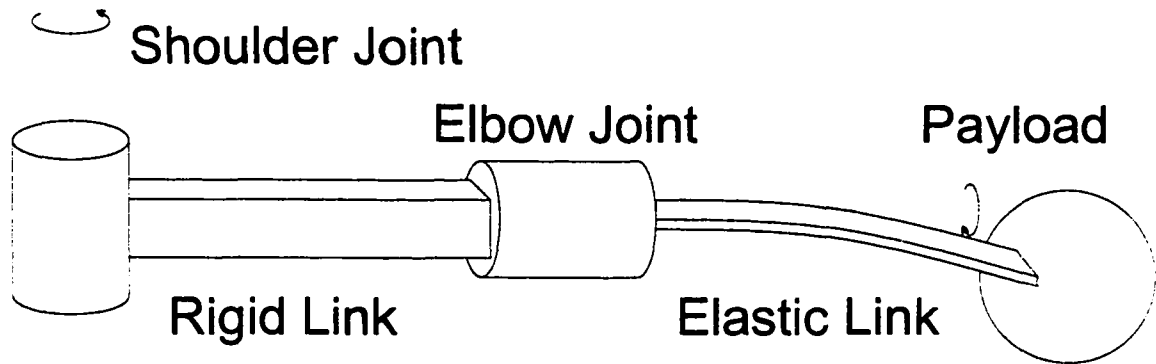


Figure 3.2: REMSPATIAL manipulator.

Body	Dimensions (cm)	Mass (kg)
Rigid	length: 53	0.8
Elastic	$50 \times 5 \times 0.6$	0.4
Sphere	diameter: 13.4	10

Table 3.1: Physical properties of REMSPATIAL. The stiffness of the flexible link is $EI = 74.7 \text{ N}\cdot\text{m}^2$.

The joint accelerations were interpolated using 500 knot points. The weighting factors in the objective are $\alpha = 10^{-1}$, $\beta = 10$, $\gamma = 10^{-10}$. The solution for the optimal joint path and path speed are plotted separately in Figure 3.3 as functions of time, while Figure 3.4 shows the path speed as a function of location along the path. As expected, the optimal path first prescribes elbow rotation that orients the flexible link to decouple vibration from shoulder joint rotation. After slewing the shoulder joint, the elbow joint is returned to the desired angle. Because decoupling is complete when the elbow is oriented at $\pm 90^\circ$, the optimal path incurs zero vibration regardless of the speed along the path. Note that, for the optimal trajectory, each joint moves while the other remains fixed, giving zero speed at the sharp corners in the joint path. Simple speed profiles commonly used for point-to-point maneuvers, such as polynomial and trapezoidal, are not well suited for such joint paths.

Figure 3.5 depicts the optimal path superimposed on the Coupling Map for the REMSPATIAL manipulator. In this case, the optimal path (solid line) agrees well with the Coupling Map. Both methods also agree with our intuition. The instantaneous direction of minimum coupling (dashed lines) requires that the elbow rotate while the shoulder does not. Once the joint configuration is in an area of relatively low coupling (lighter areas), the joint path crosses the minimum coupling lines.

For reference, the solution for the optimal joint accelerations is shown in Figure 3.6, and the corresponding joint rates and angles in Figures 3.7-3.8. The resulting vibration is represented by the elastic coordinates, Figure 3.9, and by the strain energy, Figure 3.10, which was the primary component of the optimization objective. Although we expect there to be zero vibration along the optimal path, there is a small amount of strain incurred just before the first corner of the joint path after 1 second. By the 1.5 second mark, it is clear in Figure 3.7 that both joints are moving

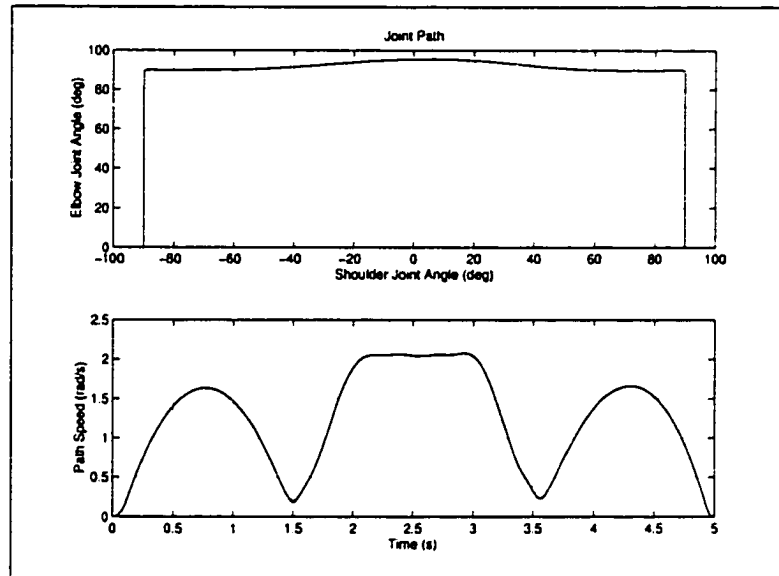


Figure 3.3: REMSPATIAL joint path and path speed.

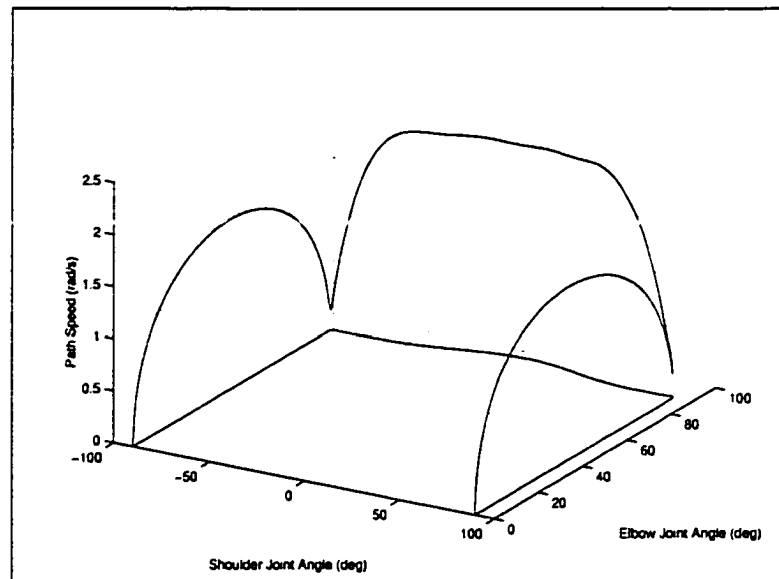


Figure 3.4: REMSPATIAL joint path speed superimposed on joint path.

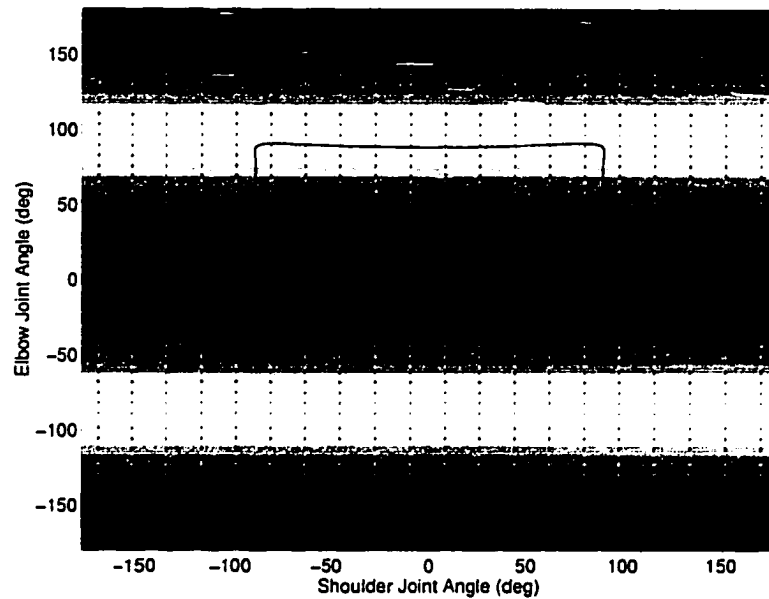


Figure 3.5: Optimal joint path (solid line) superimposed on the Coupling Map for the REMSPATIAL manipulator.

simultaneously. Since the elbow joint is not yet at 90° , there is a small amount of coupling between shoulder joint acceleration and vibration. Damping of this vibration may also explain the small deviation of the elbow joint from 90° at the mid-point of the motion. Comparison of previous results using 50 knot points (not shown) to the results presented here for 500 knot points demonstrated that the significance of both phenomena was greatly reduced with the increase in the number of knot points.

3.4.2 LAURA-SARA

The LAURA-SARA manipulator, shown in Figure 3.11, was constructed in the Space and Subsea Robotics Laboratory at the University of Victoria specifically for exper-

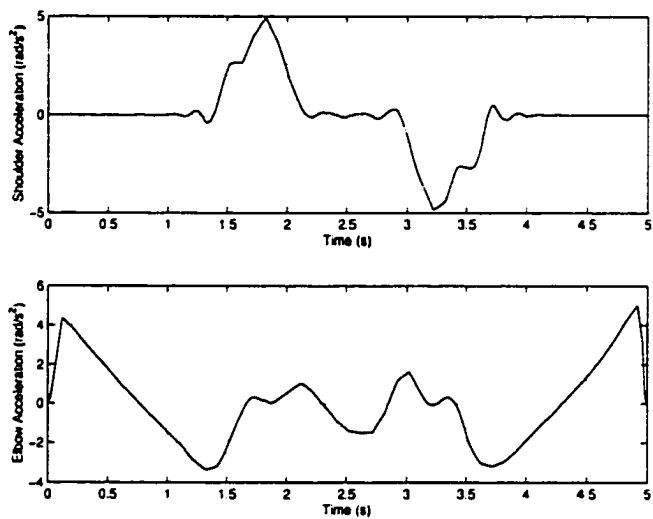


Figure 3.6: REMSPATIAL joint accelerations.

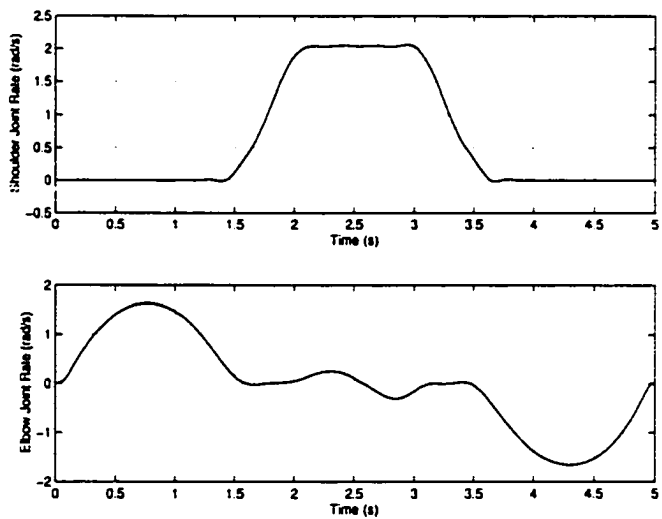


Figure 3.7: REMSPATIAL joint rates.

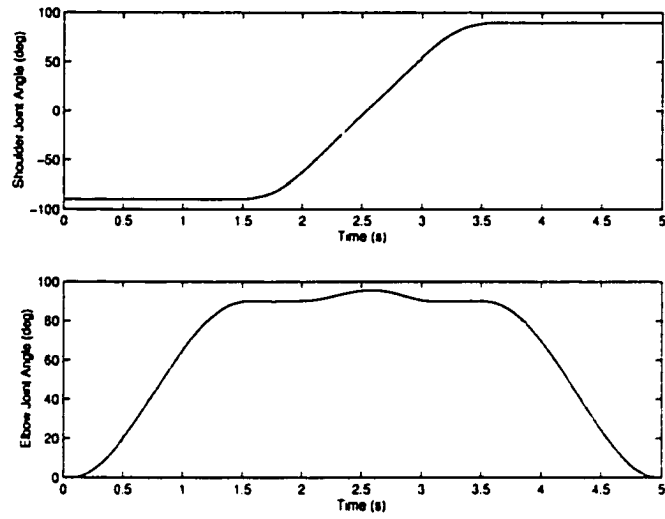


Figure 3.8: REMSPATIAL joint angles.

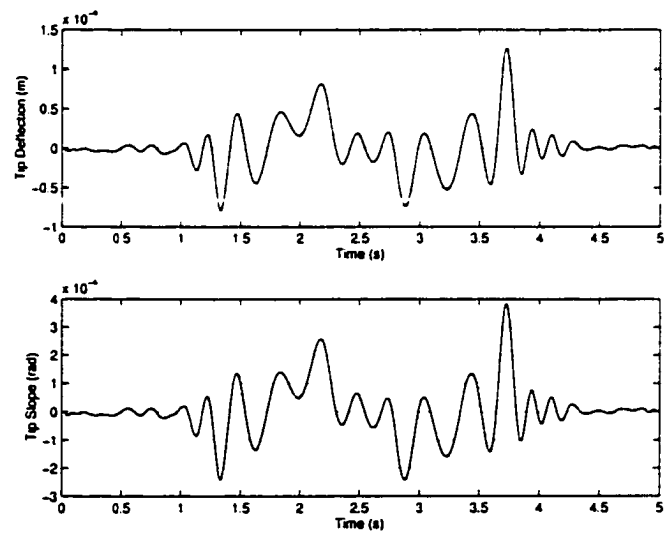


Figure 3.9: REMSPATIAL elastic coordinates, the flexible link tip deflection and slope.

imental validation of active damping techniques for macro-micro manipulators [VAN VLIET AND SHARF, 1998].

LAURA is an acronym for Long Articulated Uvic Robotic Arm and SARA for Small Articulated Rigid Arm. This macro-micro manipulator is supported on a large glass table with air bearings under each joint to minimize friction with the table. LAURA consists of two flexible links while SARA is a three-DOF rigid robot. A detailed description of the macro-micro manipulator design, geometry, stiffness, and inertia properties can be found in VAN VLIET [1998]. This section presents global trajectory optimization and simulation results for a model of the experimental manipulator.

The specified point-to-point task is to move SARA's shoulder and elbow joints from $(-180^\circ, 0^\circ)$ to $(0^\circ, 0^\circ)$, rest to rest, in 1 second. The SARA wrist and two LAURA joints are locked in the 0° , straight-out configuration. The joint accelerations are interpolated using 250 knot points. The weighting factors in the objective are $\alpha = 10^{-1}$, $\beta = 100$, $\gamma = 10^{-10}$. As before, the optimal joint path and path speed solutions are plotted separately in Figure 3.12, while the speed is plotted above the path in Figure 3.13. Although there is no zero vibration path as for REMSPATIAL, the optimal path has similar characteristics. The motion can be physically interpreted by decomposing it into three stages: folding of the elbow, shoulder slewing, and unfolding of the elbow. Initially, both of SARA's links are folded back over LAURA, the elbow link an extension of the shoulder link. In the first stage, the elbow link is folded back over the shoulder link (to -180°) in such a way that SARA's centre of mass moves axially, along the length of LAURA. The resulting reaction force on LAURA is thus in the stiffest direction. Then the elbow remains fixed while the shoulder slews towards the desired angle. Finally, the elbow link unfolds in the same manner until fully extended (0°).

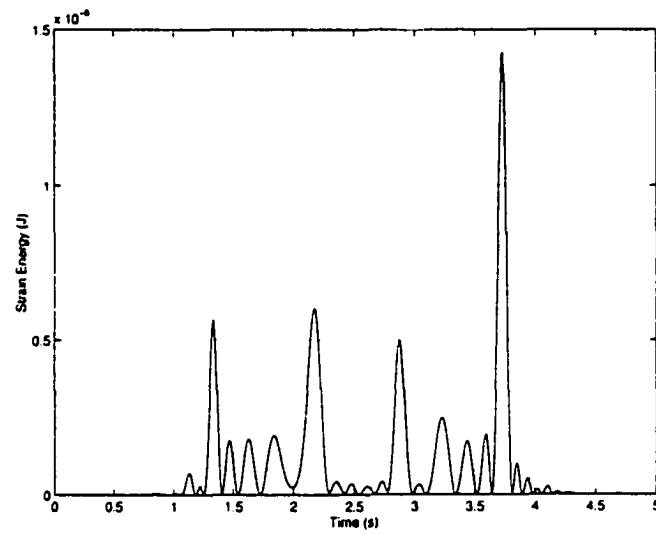


Figure 3.10: REMSPATIAL strain energy.

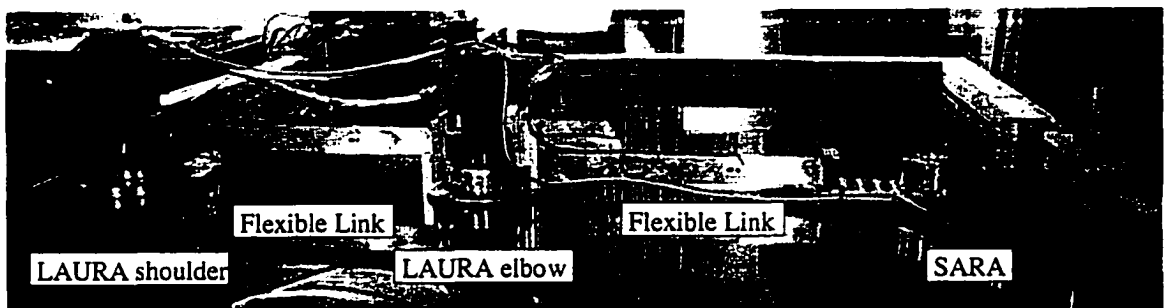


Figure 3.11: Photo of LAURA-SARA manipulator [VAN VLIET AND SHARF, 1998].

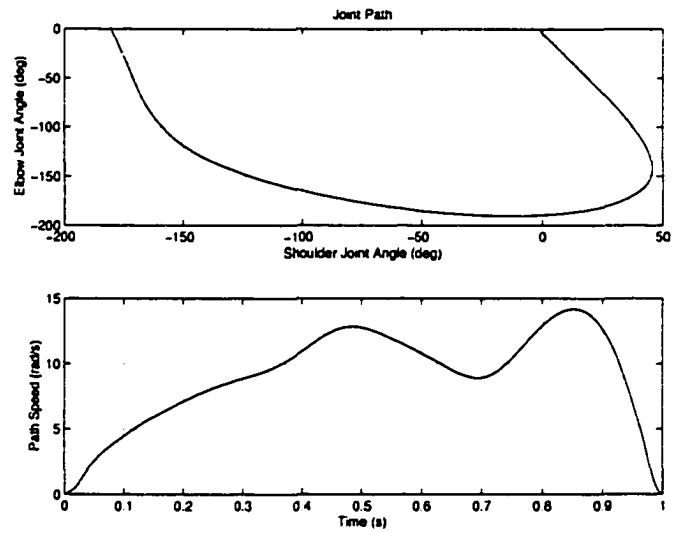


Figure 3.12: SARA joint space path and path speed.

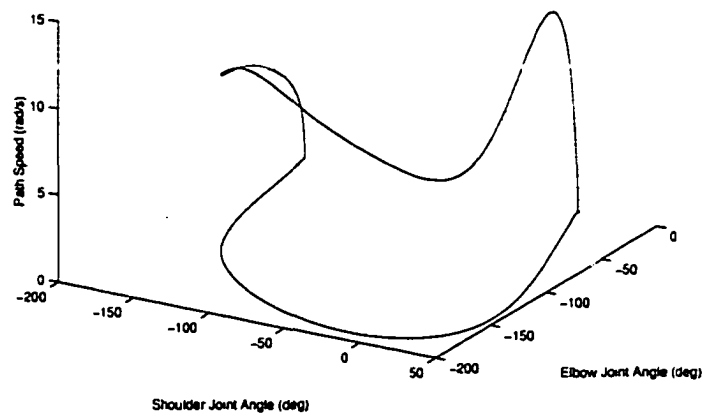


Figure 3.13: SARA path speed superimposed on joint space path.

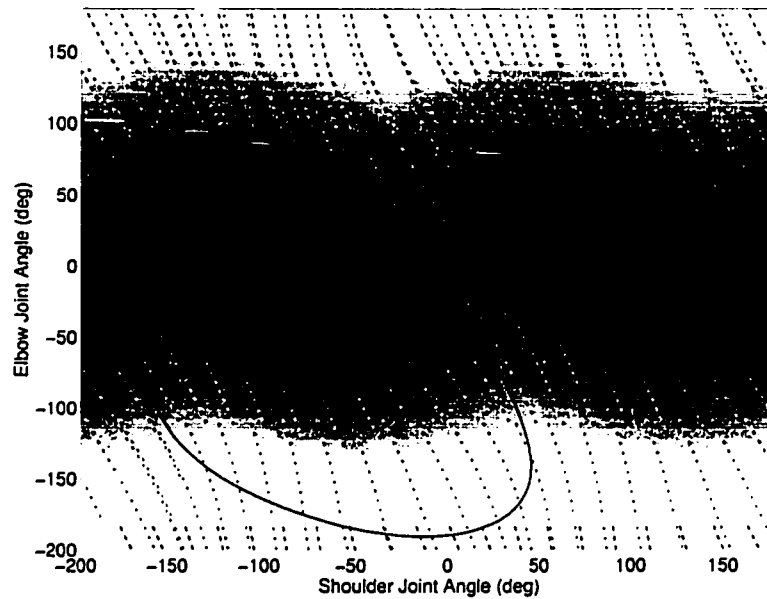


Figure 3.14: Optimal joint path (solid line) superimposed on the Coupling Map for the LAURA-SARA manipulator.

Figure 3.14 also demonstrates agreement between the optimal path and the Coupling Map. The minimum coupling lines show how both joints move to fold the elbow link over the shoulder link while restraining the motion of SARA's centre of mass to LAURA's axial direction. Once the elbow link is folded over the shoulder link at -180° , the moment applied to LAURA's tip by rotation of SARA's shoulder joint is decreased, thereby reducing vibration excitation. The same physical interpretation motivated an heuristic alternative to the Coupling Map for a similar manipulator configuration [YOSHIDA ET AL., 1996]. These results all help to verify the optimality of the global trajectory generation procedure.

The solution for the optimal joint accelerations is presented in Figure 3.15, and the corresponding joint rates and angles in Figures 3.16-3.17. The elastic deflections

(Figure 3.18) and strain energy (Figure 3.19) show that most vibration occurs during the slewing of the shoulder joint. The effectiveness of folding the elbow link over the shoulder link to reduce the reaction moment is thus again confirmed.

3.4.3 REMGRAVITY

For the final example, we again consider the REMSPATIAL manipulator depicted in Figure 3.2. However, the environment is now modified to include gravity acting perpendicularly to the rotation plane of the first, rigid link. The orientation of the elbow joint thus determines the orientation of the flexible link relative to gravity. For example, when the elbow is in the 0° configuration, the bending plane is perpendicular to gravity (decoupled), but is parallel to gravity at $\pm 90^\circ$ (coupled). Recall from the discussion of the REMSPATIAL maneuver in §3.4.1, however, that the coupling of vibration to shoulder joint acceleration displays the opposite characteristics: coupled when the elbow is at 0° but decoupled at $\pm 90^\circ$. The orientation of the elbow joint must therefore be chosen according to the net force resulting from gravity and shoulder joint rotation.

Figures 3.20-3.21 show the optimal trajectory for a 5 second maneuver from $(-90^\circ, 0^\circ)$ to $(90^\circ, 0^\circ)$. The characteristics of the optimal motion are similar to that for the REMSPATIAL example. At first the elbow rotates to the orientation for which the flexible link's bending plane is perpendicular to the net external force. Then the shoulder slews and finally the elbow rotates back to the desired angle. However, unlike the REMSPATIAL maneuver, the optimal orientation of the elbow link changes as the shoulder slews. In particular, the magnitude of the shoulder joint acceleration is maximum at the beginning and end of the motion, but zero in the middle, whereas

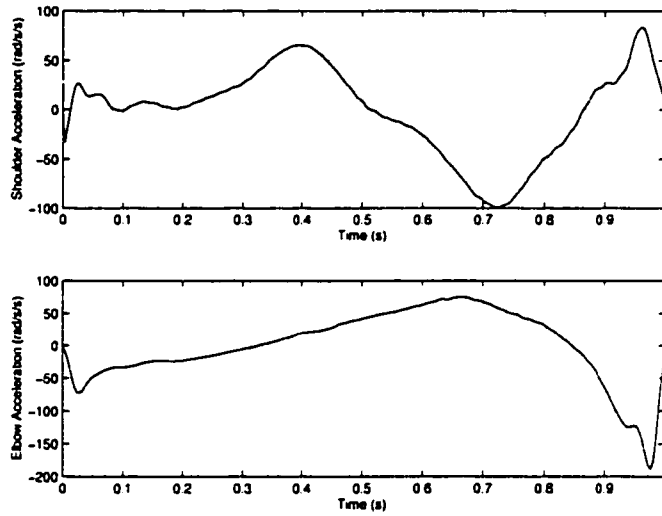


Figure 3.15: SARA joint accelerations.

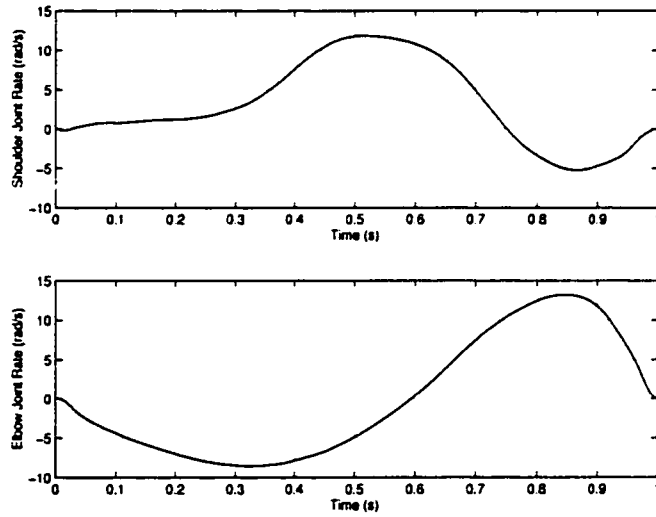


Figure 3.16: SARA joint rates.

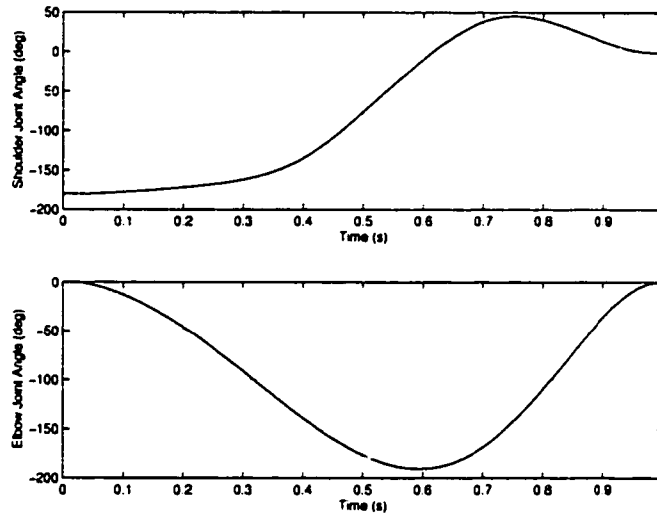


Figure 3.17: SARA joint angles.

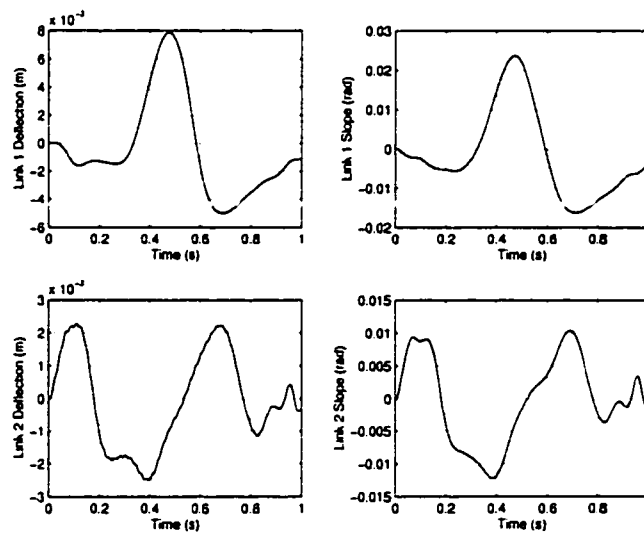


Figure 3.18: Elastic coordinates of LAURA's two flexible links.

the gravitational component of the external force remains constant. Note also that the “corners” of the joint path are rounded compared to the REMSPATIAL results because the gravitational disturbance begins to act as soon as the elbow joint angle is nonzero.

Because the optimal orientation of the elbow joint depends on the direction of the net external force, it also depends on the magnitude of the initial shoulder joint acceleration. This acceleration is in turn related to the total time specified for the point-to-point maneuver. As can be seen in Figures 3.22-3.23, a maneuver time of only 1 second requires greater initial shoulder joint acceleration and therefore a greater initial elbow re-orientation (almost 90°) for decoupling. Comparing the horizontal and vertical components of the acceleration of the base of the flexible link, we can classify maneuvers as being *slow* if the vertical (gravitational) component is larger, or *fast* if the horizontal component (from shoulder joint acceleration) is larger.

Although gravity was not included in the Coupling Map derivation, Figure 3.24 compares the fast and slow joint paths in the context of the Coupling Map information. Note that both paths violate the Coupling Map heuristics when gravity dominates the disturbance to the flexible link. For the slow maneuver, gravity is dominant throughout most of the motion and hence the joint path shown is not predicted by the Coupling Map heuristics. The joint path for the fast maneuver agrees with the Coupling Map at the beginning and end of the motion when shoulder acceleration dominates, but disagrees in the middle of the motion when shoulder joint acceleration is zero and gravity dominates. Together with the results for REMSPATIAL, we see that the Coupling Map heuristics predict suitable joint paths when the joint accelerations represent the dominant vibration disturbance throughout the motion.

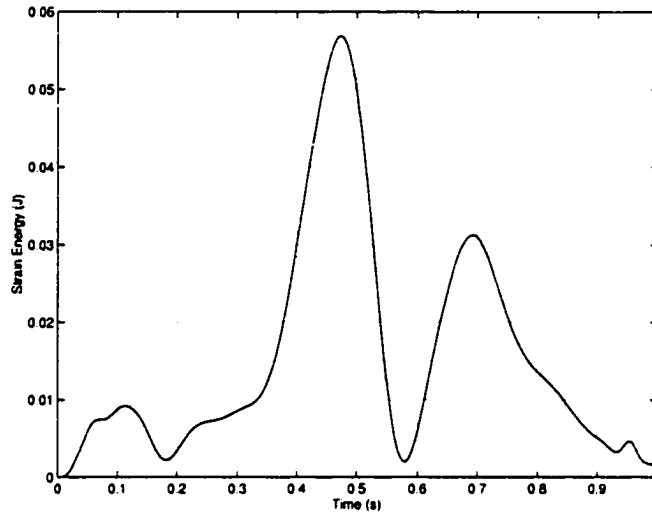


Figure 3.19: LAURA's strain energy.

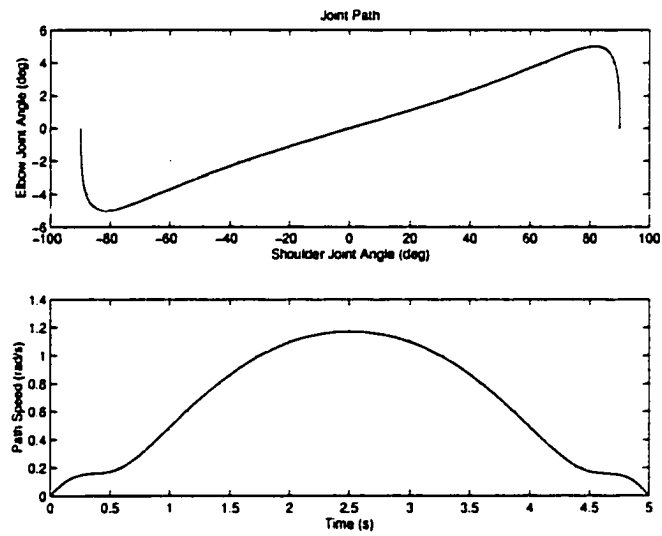


Figure 3.20: REMGRAVITY joint path and speed—slow maneuver.

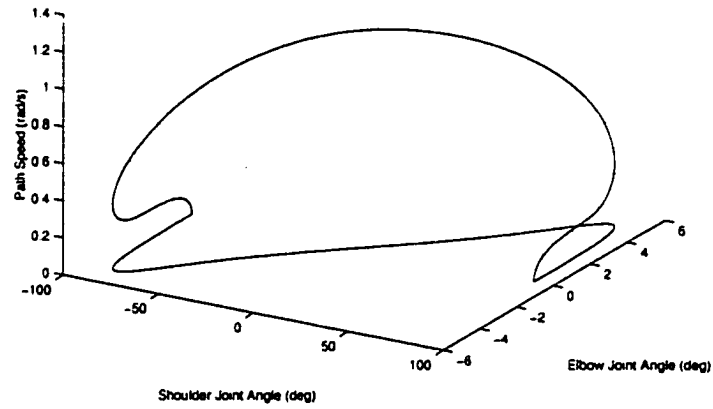


Figure 3.21: REMGRAVITY path speed superimposed on joint path for slow maneuver.

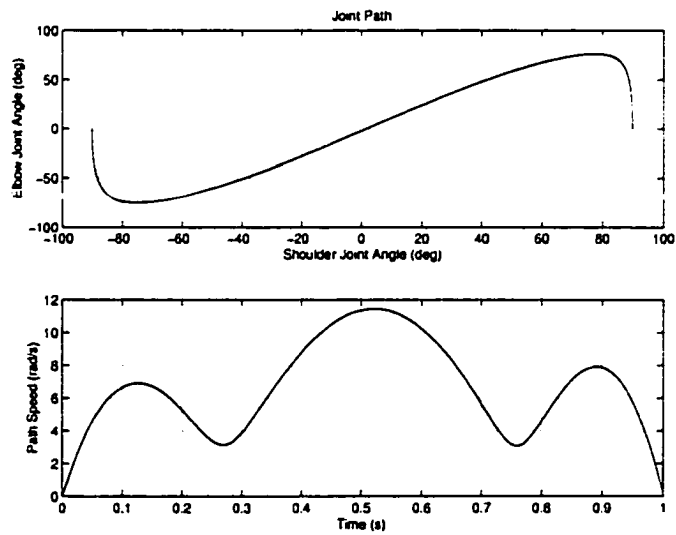


Figure 3.22: REMGRAVITY joint path and speed—fast maneuver.

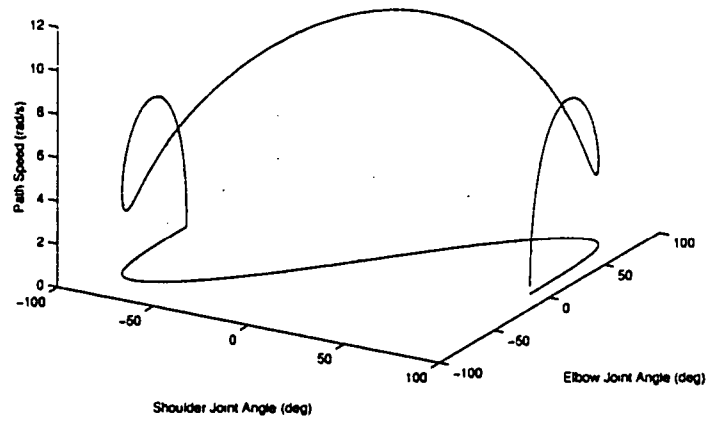


Figure 3.23: REMGRAVITY path speed superimposed on joint path for fast maneuver.

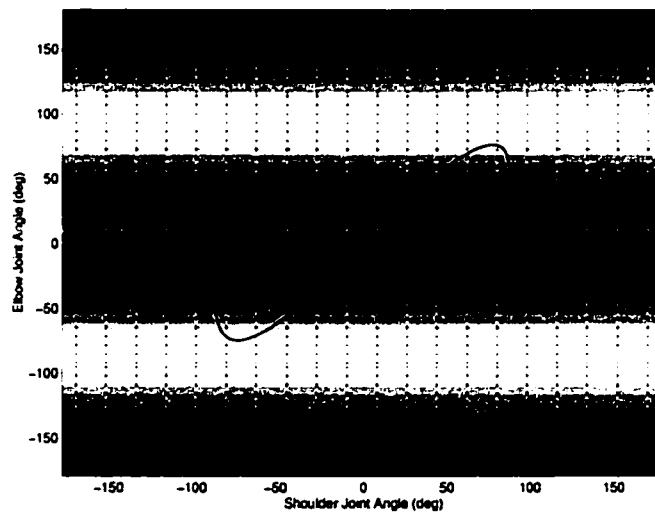


Figure 3.24: Optimal path for slow maneuver (white) and fast maneuver (black) superimposed on Coupling Map for REMGRAVITY.

3.5 Summary

This chapter has made the functional optimization formulation of the previous chapter concrete by providing details of the numerical methods employed to solve the motion optimization problem. The resulting algorithm was applied to three flexible manipulator examples. These examples helped to verify the numerical solution by admitting simple, yet nontrivial, solution paths which could be justified physically. In addition, the results were evaluated in the context of the Coupling Map heuristics and were found to be in excellent agreement whenever the assumptions of the Coupling Map derivation were valid.

Chapter 4

Experimental Evaluation

4.1 Introduction

As mentioned in Chapter 1, motion planning is a practical problem for the efficient use of flexible manipulators. Ultimately, therefore, any solution to the motion planning problem should be relevant to actual flexible manipulators in real environments. This chapter presents the experimental evaluation of the trajectories produced by the global trajectory optimization procedure of the previous chapter.

The purpose of the experimental demonstration is to determine the extent to which the generated trajectory improves the actual performance of the manipulator with respect to other trajectories and a given objective functional. The experimental flexible manipulator, described in §4.2, is configured so that the globally optimal path in joint space differs in an intuitive way from a straight line. After identifying a suitable model, we compare the globally optimal trajectory with two sub-optimal trajectories having straight-line joint paths: one generated by a polynomial and the other by optimizing the speed along the straight-line path. In theory, the globally

optimal trajectory minimizes the strain energy integrated over the entire motion. Experimentally, all trajectories are evaluated relative to that same objective. Section 4.3 compares the total strain energy incurred by the three trajectories, both in simulation and experimentally. These results allow us to conclude that the globally optimal trajectory provides significant performance improvement over the other trajectories.

4.2 The Experimental Flexible Manipulator

4.2.1 Manipulator Construction

The experiments were conducted on a flexible robotics testbed within the Space and Subsea Robotics Laboratory [NAHON ET AL., 1995]. The flexible manipulator was constructed of two NSK direct-drive motors, the “shoulder” and the “elbow”, which were connected by a rigid link and supported by air bearings on a glass table-top. Figure 4.1. A flexible aluminum link (6.35 mm × 50.8 mm × 500 mm) was mounted vertically along the axis of the elbow motor and a payload was attached to the tip of the flexible link. This configuration was chosen so that the orientation of the link’s bending plane with respect to the direction of motion of the base of the link could be determined by the elbow joint angle. Thus, the coupling of shoulder joint rotation to link vibration could be varied. Local strain of the flexible link is sensed by two strain gauges permanently affixed at one-third and two-thirds of the length of the link. A 4.4 kg steel cylinder bolted to the top of the flexible link formed the payload as seen in Figure 4.1. Also visible is a horizontal rigid link attached to the output coupler of the elbow motor. This link was included simply to increase the rotational inertia of the payload, thereby improving the trajectory tracking performance of the controller.



Figure 4.1: Experimental flexible manipulator.

4.2.2 Manipulator Model

To optimize the point-to-point trajectory which the manipulator will follow, the global optimization procedure requires a model. This section presents a model of the experimental manipulator whose parameters were obtained partly from a previous analytical model and partly from experimental identification.

Since the manipulator was constructed from existing hardware, the mass properties and dimensions of the motors and links were obtained from an earlier analytical model, as was the stiffness of the link [STANWAY, 1996]. Because of the vertical orientation of the flexible link, gravity was included in the simulation, although a simple analytical model showed that the first natural frequency would be lowered by only 0.1 Hz compared to the gravity-free case. The finite element model of the cantilevered

Diameter:	127 mm
Height:	44.4 mm
Mass:	4.4 kg

Table 4.1: Properties of the steel cylinder payload.

flexible link employed a single beam element having two elastic degrees of freedom, the deflection and slope at the tip. More elements were not used for three reasons:

1. The continuous model (being the limit as more elements are added) remained insufficient to predict the observed first natural frequency.
2. A single element model is much quicker to simulate since it does not include the higher (and less significant) natural frequencies.
3. Since there are only two strain gauges attached to the link, only two elastic degrees of freedom can be estimated without implementing a state observer.

Apart from the motors and the links, two additional components were added to the model. The first was the steel cylinder forming the payload. The height and diameter of the cylinder were measured while the mass properties were calculated from the density of steel (Table 4.1).

The second additional component of the model was a virtual “hinge” located at the base of the flexible link. Because the elbow motor was observed to “rock” within its casing during vibration of the flexible link, the hinge was added to allow an additional degree of freedom between the elbow joint and the flexible link. The hinge was modelled as a rod in torsion with a single elastic degree of freedom and was oriented to rotate in the bending plane of the flexible link, Fig. 4.2. Experimental

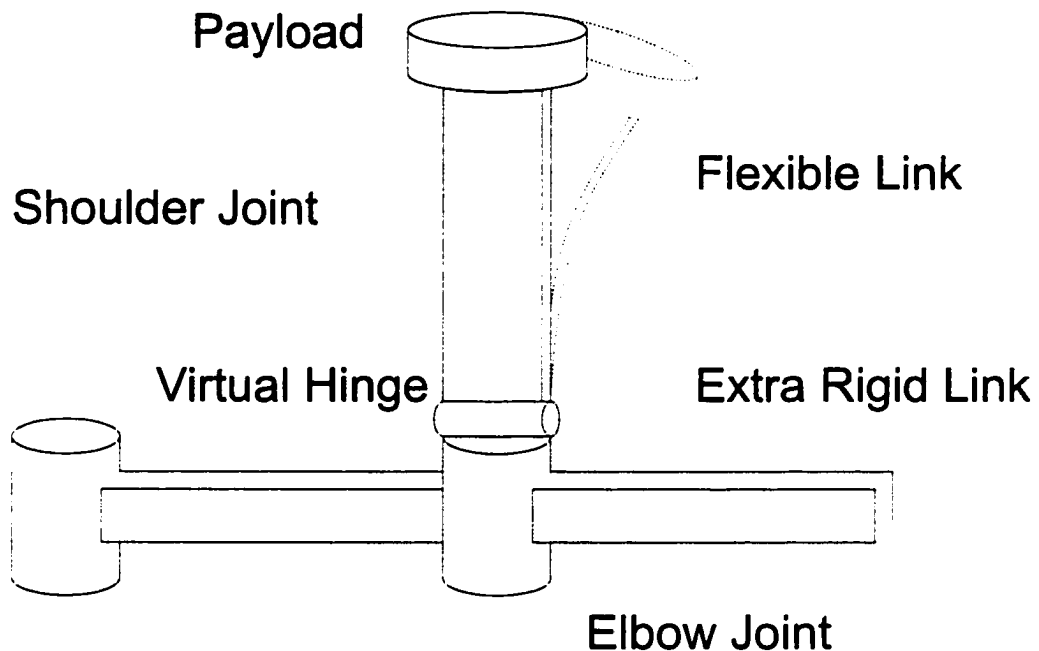


Figure 4.2: Experimental manipulator model, including virtual hinge and payload, shown in the zero configuration.

identification of the radius, rotational inertia, and torsional stiffness of the hinge is described in the next sub-section.

4.2.3 Vibration Testing and Parameter Identification

Because the trajectory optimization procedure prescribes trajectories that attempt to dampen vibration at the frequency predicted by the model, and there is no feedback of elastic deformation in these experiments, it is important for the manipulator model to have natural frequencies similar to the experimental manipulator. Otherwise, a large mismatch between actual and model frequencies results in poor damping or even vibration excitation. The parameters of the virtual hinge were therefore chosen to

attain a satisfactory agreement between the natural frequencies of the model and the experimental manipulator, particularly at the lowest frequency.

To determine the natural frequencies of the assembled system, vibration of the flexible link was excited and the strain gauge data was recorded with both the joint motors and the air bearing system off. The bending moments at the strain gauge locations (and, in general, the joint angles as well) were calculated at 100 Hz. The vibration was induced in three different ways:

Plucked: In the first test, data recording began after the payload was pulled back and released. Mostly bending was excited although significant rocking of the elbow motor was also induced.

Struck: In the second test, the link was struck halfway along its length, and on the corner of the cross-section (*i.e.* not evenly on the flat face). There was visible torsional vibration of the flexible link as well as bending.

Slapped: Finally, the end of the extra rigid link was slapped. There was significant torsional vibration of the flexible link and vibration of the tip of the extra “rigid” link.

The peaks in the power spectral density of the estimated tip slope, Figure 4.3, identify the natural vibration frequencies of the flexible link. All three spectra agree in placing the first natural frequency at 2.5 Hz. In contrast, the model without the hinge predicted a first natural frequency of about 3.75 Hz.

For the purpose of demonstrating the trajectory optimization procedure, Section 4.3 shows that matching only the first natural frequency was sufficient to give satisfactory results. To achieve such a match, the radius, rotational inertia, and tor-

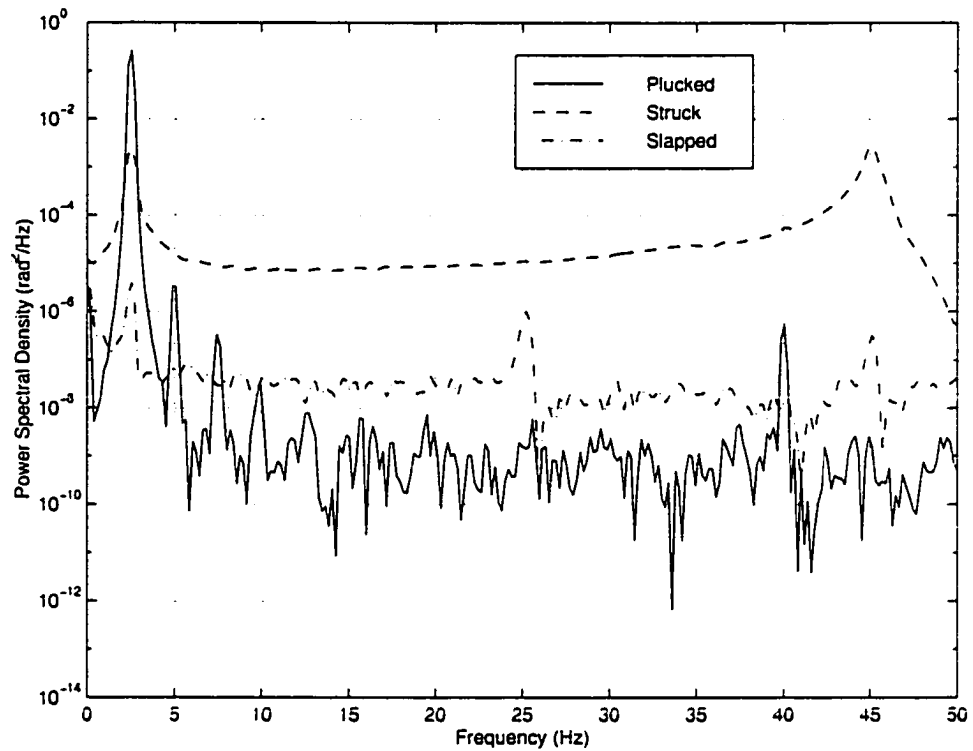


Figure 4.3: Power spectral density of flexible link tip slope under different modes of excitation.

sional stiffness of the virtual hinge were adjusted. These three parameters allowed a secondary goal to be achieved, which was to lower the highest natural frequency of the composite model. The highest frequency component has negligible influence on the results, however it limits the size of the time steps during numerical simulation. Therefore, simulation speed can be increased by lowering the highest frequency without introducing inaccuracies in the model.

Table 4.2 gives the parameters of the virtual hinge obtained by matching the first natural frequency and minimizing the highest frequency. The frequencies were determined from the finite element model of the link, payload mass, and virtual hinge, and the minimization was performed graphically. Clearly the hinge is a non-physical addition to the analytical model of the manipulator. The hinge does, however, allow the numerical simulation to incorporate easily an empirical property of the experimental manipulator, the first natural vibration frequency. For future investigations, the analytical model can be refined by determining the physical nature of the rocking of the elbow motor.

Radius:	0 m
Rotational Inertia:	0.05 kg·m ²
Torsional Stiffness:	672 N·m

Table 4.2: Parameters of the virtual hinge.

As a final note, we mention that the damping ratio of the flexible link (including rocking of the elbow motor) was determined to be less than 1% by fitting a sinusoid to the tip slope. Damping was therefore not included in the simulation model.

Gains \ Joint	Shoulder	Elbow
Proportional (N·m)	6400	500
Derivative (N·m·s)	150	12

Table 4.3: Independent PD controller feedback gains for joint angle and angular rate.

4.2.4 Joint Controller

Throughout all experiments, a simple proportional-derivative (PD) controller was used for joint trajectory tracking. One hundred set points for each of the desired joint angles and desired joint rates were linearly interpolated by the controller, which ran at 4000 Hz [VAN VLIET AND SHARF, 1998]. The feedback gains were adjusted independently according to heuristics provided by the motor manufacturer. The resulting gains are listed in Table 4.3. The performance of the controller will be summarized at the end of the experimental results presented in the next section.

4.3 Trajectory Comparison

Having defined a suitable model of the experimental manipulator for input to the global optimization procedure, we can generate point-to-point trajectories for comparison. In this section, we present simulation and experimental results showing the performance improvement possible with trajectory optimization.

4.3.1 Task Descriptions

Two point-to-point motion tasks will be presented in parallel. Both tasks consist of a shoulder joint slew with the elbow joint finishing at the same angle it started at.

Specifically, the first task was to move from $(0^\circ, 0^\circ)$ to $(120^\circ, 0^\circ)$ in joint space, rest to rest, in 4 seconds. The second task was identical, but the initial and final elbow joint angles were 90° rather than 0° . Recall that only planar bending of the flexible link was modelled. Elbow joint rotation thus varies the orientation of the bending plane with respect to the motion of the base of the flexible link. In the 0° configuration, the bending plane of the flexible link is perpendicular to the rigid link. These tasks were therefore chosen to highlight the vibration decoupling possible by appropriate motion of the elbow joint, that is, by not following a straight-line in joint space.

The tasks were also chosen to ensure that physical constraints of the experimental arm would not be active. These constraints included joint limits, workspace boundaries, and motor torque limits. Thus, the start and end points, and the total time of the trajectory, were determined on an *ad hoc* basis. The optimal trajectory would therefore not be influenced by any efforts to maintain such constraints.

4.3.2 Trajectory Generation

The motion of the manipulator in joint space can be decomposed into two independent parts: the path of the joints in joint space, and the speed at which the path is traversed. As elaborated in Chapter 1, motion planning methods typically vary either the path or the speed, few vary both.

As an illustration of the effects of varying the joint path and speed separately and in combination, we consider the following trajectories for comparison:

Fifth-order polynomial: The first trajectory generation method expresses the joint angles as a fifth-order polynomial in time. Besides the specified end-points, the boundary conditions are zero velocity and acceleration at each end of the

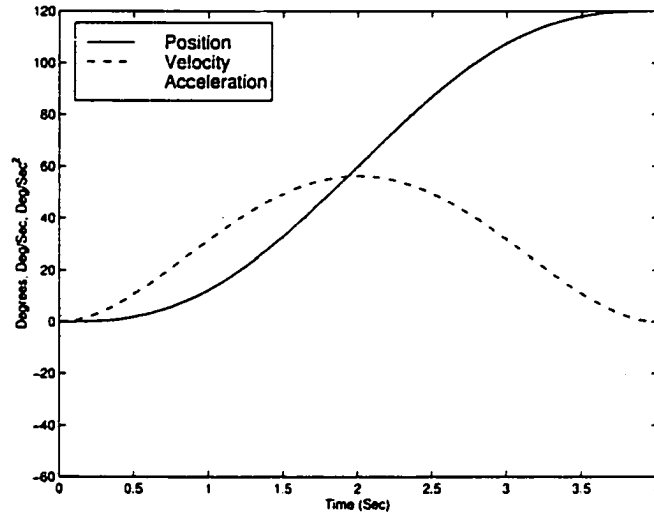


Figure 4.4: Fifth-order polynomial joint profile, along with joint rate and acceleration. The initial and final joint angles and total time are as used in the experiments.

trajectory. The polynomial, for angle x , total time T , and end-points $x(0) = x_0$ and $x(T) = x_T$, is:

$$x(t) = \left(\frac{t}{T}\right)^3 \left[6 \left(\frac{t}{T}\right)^2 - 15 \left(\frac{t}{T}\right) + 10 \right] (x_T - x_0) + x_0 \quad (4.1)$$

and is plotted in Figure 4.4. This common method of trajectory generation always gives straight line paths in joint space. The speed along the path is also predetermined by the polynomial, so no optimization takes place. Thus, this particular trajectory demonstrates the case when the trajectory between the end-points of the maneuvers is specified.

Globally optimal straight line: The second method shows the best that can be achieved without considering different paths in joint space. Only the speed is optimized by constraining the motion to the straight line path in joint space.

By choosing equal initial and final elbow joint angles for the comparison tasks, the straight line path requires a shoulder joint slew only while the elbow joint angle remains constant. The straight line path constraint was thus implemented simply by removing the elbow joint degree of freedom from the simulation model and performing the global trajectory optimization. These trajectories were generated using the global optimization procedure of the previous chapter with cost parameters $\alpha = 10^{-1}$, $\beta = 10$, and $\gamma = 10^{-10}$.

Globally optimal: Finally, the third method includes the previous method but also considers different paths in joint space. The same cost parameters were also used. This is the most general case in which both the joint path and speed are optimized.

In the discussion to follow, we shall refer to each of these three trajectories as “Polynomial”, “Constrained”, and “Optimal”, respectively.

We now compare the above three trajectories as determined through simulation. For Task 1, Figure 4.5 depicts each of the three paths in the horizontal plane with the path speeds superimposed in the vertical dimension.

A physical interpretation of the globally optimal path is as follows. Vibration of the flexible link is excited by the translational inertial force of the elbow motor acting on the base of the flexible link. (Recall that the model assumes elbow joint rotation alone does not excite vibration.) Thus we can analyze the disturbance to the flexible link in terms of the linear acceleration of the elbow motor, to which the inertial force is in constant proportion. This acceleration lies in the plane of the glass table and is represented by the large arrows in Figure 4.7. The short dotted lines are aligned with the axis of the virtual hinge and show the orientation of the elbow

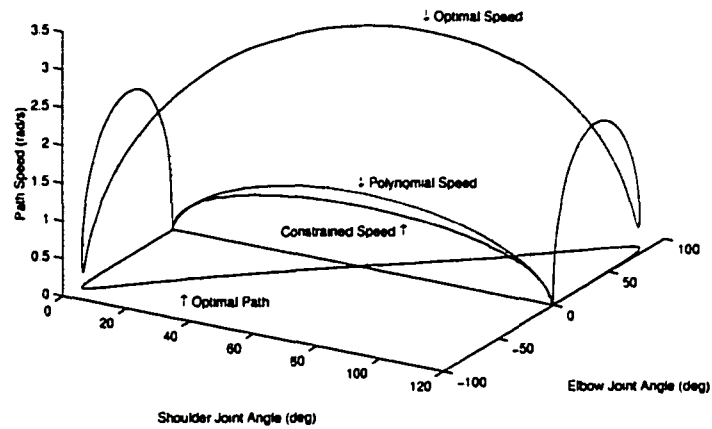


Figure 4.5: Task 1: Paths and path speeds.

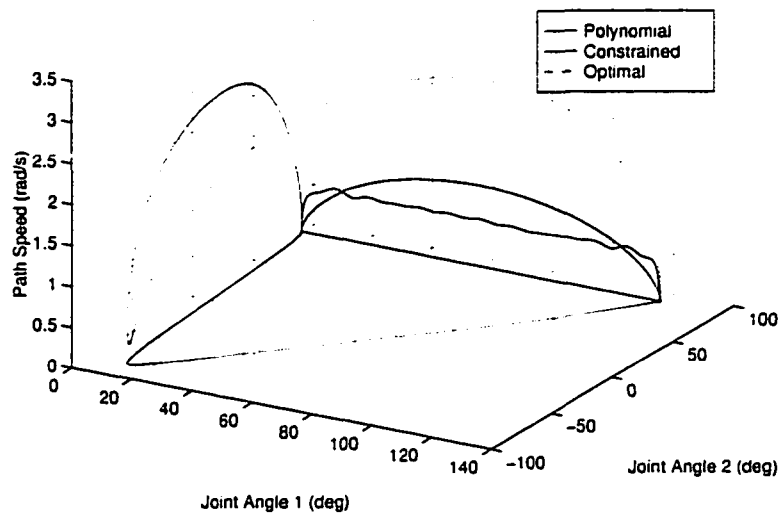


Figure 4.6: Task 2: Paths and path speeds.

joint. The bending plane of the flexible link is normal to these dotted lines, so no deflection is possible in the direction of the lines. As seen in Figure 4.5, the globally optimal path therefore begins by orienting the elbow to decouple link vibration from acceleration of the elbow motor. During the slewing of the shoulder joint, Figure 4.7 shows that the globally optimal path reduces vibration by re-orienting the elbow to maintain the bending plane perpendicular to the linear acceleration of the elbow motor. The vibration disturbance, represented by the linear acceleration, therefore has no component in the bending plane of the flexible link. A similar interpretation holds for the results of Task 2 (Figure 4.6). Except for the initial and final orientation of the elbow, the globally optimal path is identical to the globally optimal path for Task 1.

This example illustrates a fundamental shortcoming of the Coupling Map. When the globally optimal path for Task 1 is superimposed on the Coupling Map, Figure 4.8, we see that the globally optimal path does not always agree with the Coupling Map heuristics. In particular, at the midpoint of the trajectory, the globally optimal path crosses the minimum coupling lines in an area depicted as having relatively high coupling. Recall that the Coupling Map heuristics permit the joint path to cross the minimum coupling lines in areas of relatively low coupling only. The qualitative degree to which the Coupling Map predicts the optimal path can be explained by decomposing the disturbance—the linear acceleration of the elbow motor—into tangential and radial components, as shown in Figure 4.9. Comparing this figure with Figure 4.8, it can be seen that the Coupling Map agrees well with the globally optimal path when the tangential acceleration is the dominant component of the disturbance. However, the radial component is ignored by the Coupling Map in determining the direction of minimum coupling and agreement with the globally optimal path is poor when the

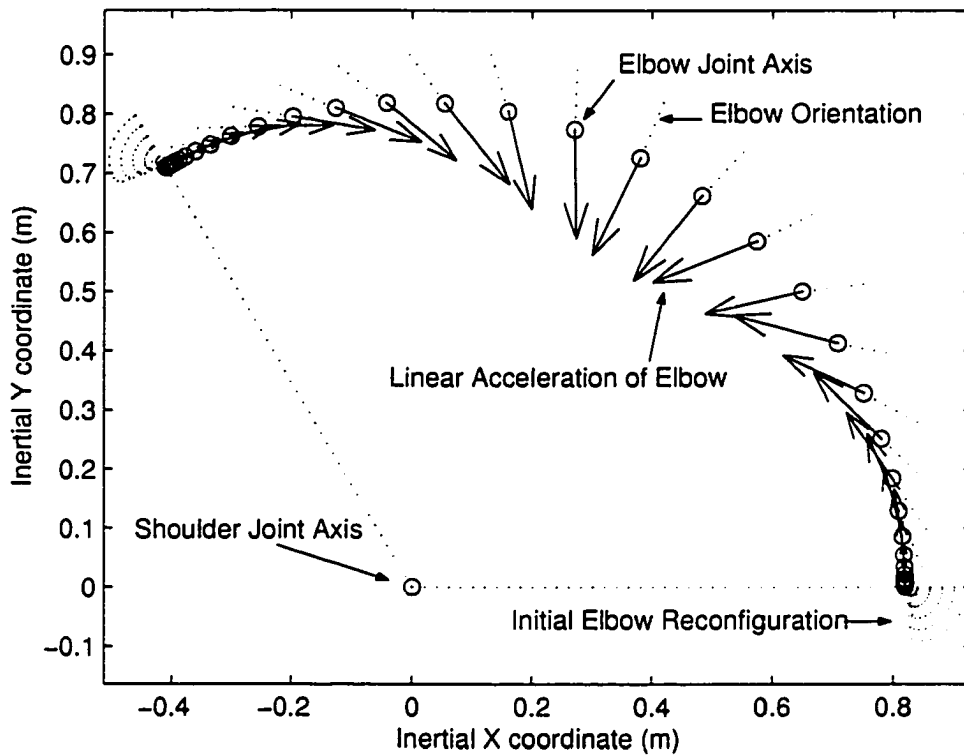


Figure 4.7: Overhead view of 120° slew of shoulder joint. The dotted line showing the orientation of the elbow joint is parallel to the virtual hinge and perpendicular to the bending plane of the flexible link. The optimal path orients the elbow joint so that the bending plane of the flexible link is perpendicular to the linear acceleration of the base of the link.

radial component dominates. Note that, for typical slew maneuvers such as this one, the tangential acceleration is always zero at the midpoint of the trajectory and the radial component will always dominate there, no matter how small. Although the Coupling Map derivation assumes that momentum is conserved and internal forces are small, the examples of §3.4 show that the Coupling Map may be successfully applied to fixed-base manipulators (for which momentum is not conserved). However, this example illustrates that the success of the Coupling Map is dependent on the relative—and not the absolute—magnitudes of the different types of internal forces.

For the straight-line paths of Task 2 (Figure 4.6), a more pronounced difference is apparent between the globally optimal straight-line speed and the polynomial speed. Since the flexible link is initially oriented to decouple vibration from shoulder joint (tangential) acceleration, a larger initial and final acceleration is possible. The result is a lower speed at the mid-point of the trajectory, which in turn reduces the radial acceleration of the base of the flexible link. It is this radial acceleration, combined with the orientation of the flexible link, which acts to bend the link. Optimization of the speed for a given path can therefore also significantly decrease vibration.

4.3.3 Trajectory Tracking Performance

Before presenting the experimental results, it is appropriate here to summarize the performance of the PD joint controller. Since the actual motion achieved experimentally is not exactly the motion that was prescribed by the simulation results, it is important to characterize the differences, particularly as they relate to vibration excitation.

The actual paths followed experimentally are indistinguishable from the com-

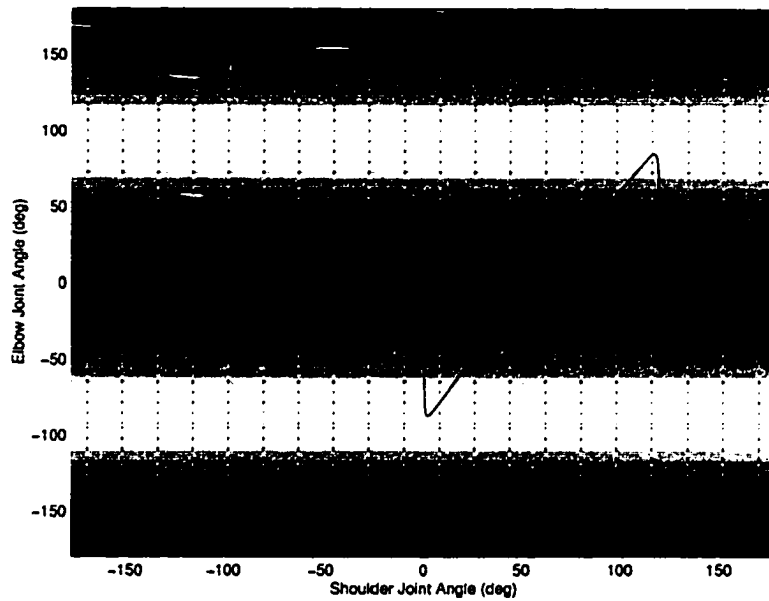


Figure 4.8: Coupling Map for Task 1: $(0^\circ, 0^\circ)$ to $(120^\circ, 0^\circ)$.

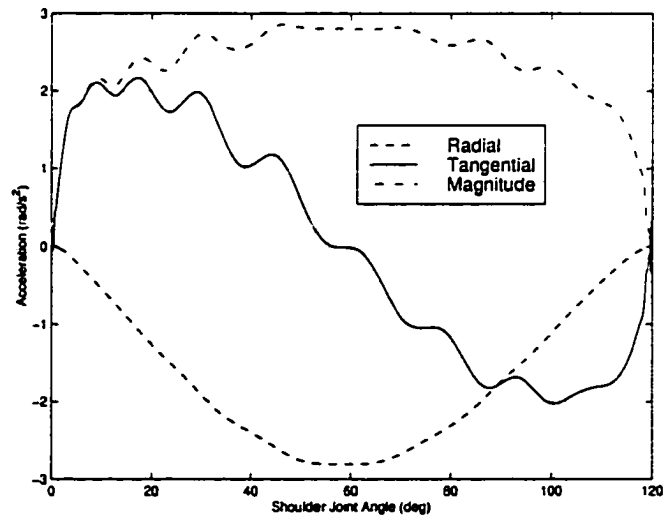


Figure 4.9: Radial and tangential components of the linear acceleration of the elbow motor, as well as total magnitude, during the slewing of the shoulder joint.

Tracking Error Trajectory		Angles (deg)		Rates (rad/s)	
		Shoulder	Elbow	Shoulder	Elbow
Task 1	Polynomial	0.2885	0.2690	0.1107	0.1100
	Constrained	0.2345	0.2150	0.0778	0.0820
	Optimal	0.7805	1.0678	0.2844	0.4546
Task 2	Polynomial	0.2890	0.5940	0.1086	0.0908
	Constrained	0.3965	0.7040	0.2009	0.1414
	Optimal	0.6983	1.0230	0.2734	0.4428

Table 4.4: Maximum absolute trajectory tracking errors.

manded paths at the scale shown earlier in Figures 4.5 and 4.6. In fact, the largest of the maximum absolute joint angle errors was about 1° for each task, as summarized in Table 4.4. Normalizing these errors by the corresponding maximum joint displacement (120° for the shoulder joint, all trajectories, or 90° for the elbow joint, optimal trajectory only) gives relative errors less than 1%.

The joint rate tracking errors were more significant. Whereas the straight-line trajectories require no elbow joint motion, the optimal trajectory is more demanding. Consequently, the joint rate tracking errors were greatest during the optimal trajectory, particularly for the elbow joint (Table 4.4). To put these errors in perspective, the joint rates predicted by the model and the rates calculated by first-order finite differences of the experimental joint angles are compared for Task 1 in Figures 4.10–4.12. Again, we see that the controller is better able to track the straight-line trajectories. Accordingly, the following section shows that the qualitative agreement between simulated and experimental strain energies is the best for these trajectories.

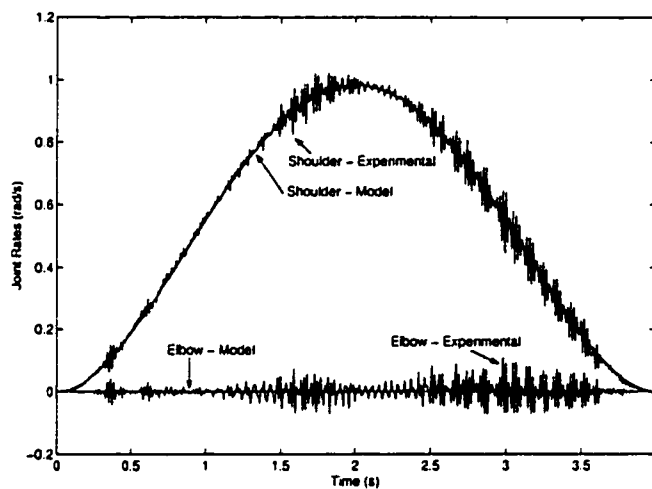


Figure 4.10: Model and experimental joint rates for Task 1: “Polynomial” trajectory.

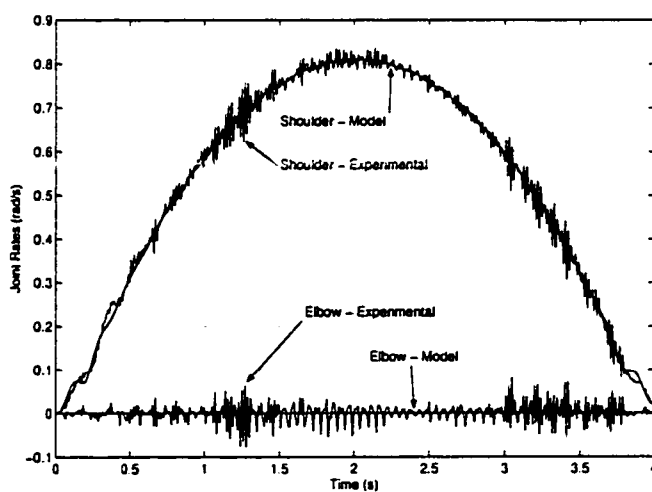


Figure 4.11: Model and experimental joint rates for Task 1: “Constrained” trajectory.

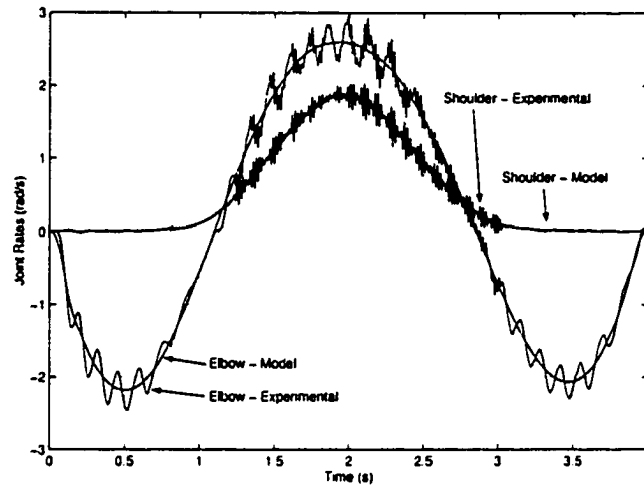


Figure 4.12: Model and experimental joint rates for Task 1: “Optimal” trajectory.

4.3.4 Strain Energy Comparison

This section presents the strain energies resulting from each experimental trajectory, comparing them to each other and to simulation. All the trajectories compared here satisfy the primary task of point-to-point joint motion, at least within a desired tolerance. Our choice of objective function, however, also reflects our secondary desire to minimize the strain energy created during the motion. Note that the objective function does not include the deflection of the virtual hinge, and neither do the simulated or experimental strain energies presented here. A quantitative cost comparison for all cases will be given in the final section.

To estimate the strain energy incurred experimentally, we employ the simulation model of the link as a cantilever beam. As for a single beam element, the link deflection is approximated by a cubic polynomial $y(x)$ in the distance x from the

clamped end. The four coefficients of the polynomial were determined from the beam flexure equation $EIy''(x) = M(x)$ and the conditions

$$y(0) = 0, \quad y'(0) = 0, \quad y''(L/3) = \frac{M_1}{EI}, \quad y''(2L/3) = \frac{M_2}{EI} \quad (4.2)$$

where E is the constant Young's modulus of the beam material, I is the constant area moment of inertia of the beam cross-section, L is the length of the beam, and M_1 and M_2 are the bending moments recorded at the positions $x = L/3$ and $x = 2L/3$, respectively. The finite element coordinates for the single element model are the tip deflection and slope, $y(L)$ and $y'(L)$ respectively. Estimates of these coordinates are

$$y(L) = \frac{L^2}{2EI}M_1 \quad (4.3)$$

$$y'(L) = \frac{L}{2EI}(M_1 + M_2), \quad (4.4)$$

which are obtained by evaluating the polynomial shape function. Finally, the experimental strain energy is approximated by

$$\tilde{V} := \frac{1}{2}\tilde{\mathbf{q}}_e^T \mathbf{K}\tilde{\mathbf{q}}_e \quad (4.5)$$

where \mathbf{K} is the stiffness matrix provided by the finite element model and $\tilde{\mathbf{q}}_e := (y(L), y'(L))^T$ is the assembled vector of estimated finite element coordinates.

The strain energy during Task 1, according to the simulation model, is plotted in Figure 4.13 for each of the three comparison trajectories. Comparing with Figure 4.14, we see excellent qualitative agreement between the predicted and the estimated experimental strain energy for each trajectory. Figures 4.15 and 4.16 give similar conclusions for Task 2.

Quantitatively, however, the estimated experimental strain energies were greater than predicted by the simulation, which may be attributed to several factors. First,

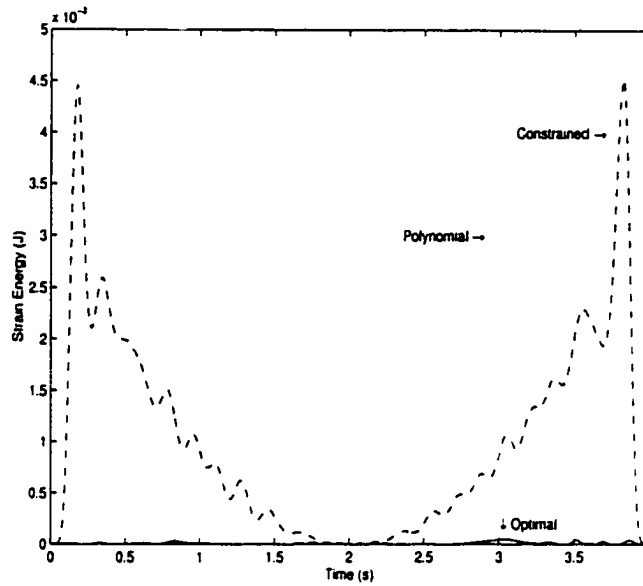


Figure 4.13: Task 1: Simulated strain energies.

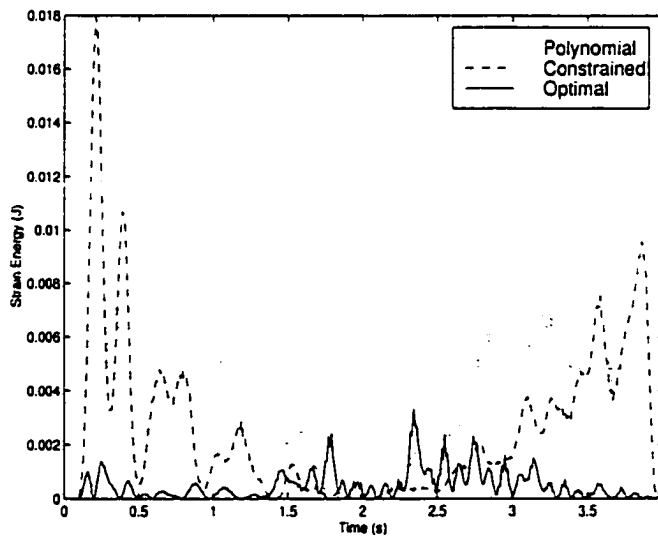


Figure 4.14: Task 1: Experimental strain energies.

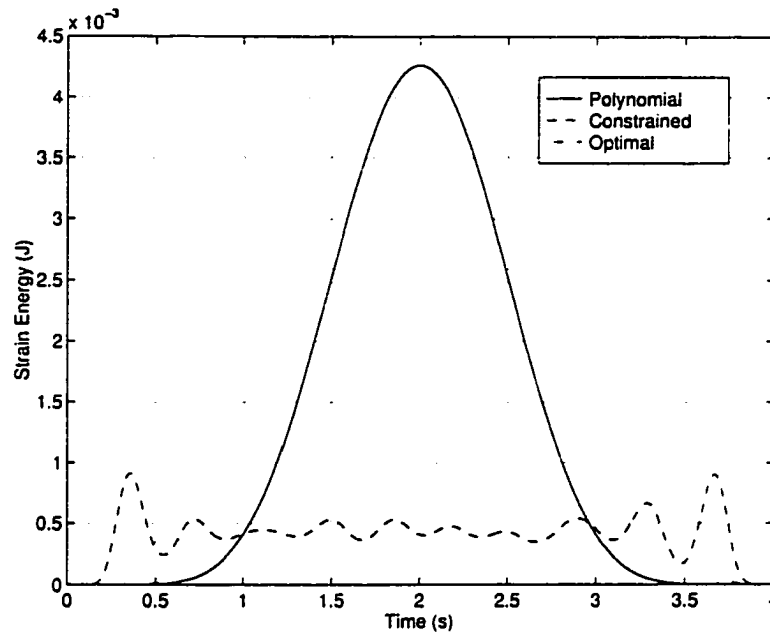


Figure 4.15: Task 2: Simulated strain energies.

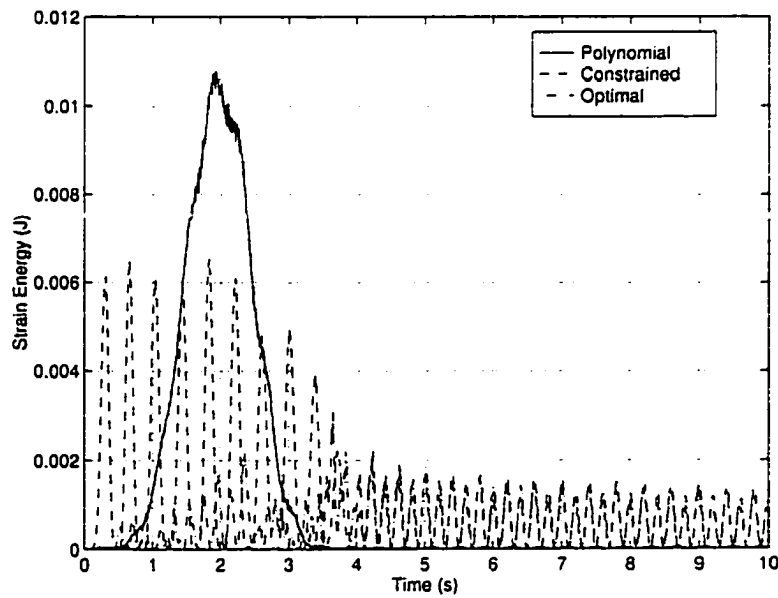


Figure 4.16: Task 2: Experimental strain energies.

only two strain measurements are made along the length of the link. Increasing the number of strain gauges would improve the experimental strain energy estimate. Second, trajectory tracking errors as discussed in the previous section create disturbances that may differ from the model in amplitude, frequency, or phase. Any such differences may increase link deflections and therefore strain energy. Finally, modelling errors tend to overestimate the structural stiffness of the link, for example by using only a single finite element. The lower actual stiffness typically leads to greater strain energy for a given applied force.

4.3.5 Cost Comparison

To give a more precise comparison of the three trajectories, in simulation and experiment, for both tasks, we tabulate the value of the terms in the objective function for each case. The three terms in the objective function are the penalty on the final cost, weighted by the \mathbf{P} matrix, the integral of the strain energy, weighted by the \mathbf{Q} matrix, and the integral of the input cost, weighted by the \mathbf{R} matrix. These three terms are labelled “Final”, “Strain”, and “Inputs” respectively.

For example, Table 4.5 gives these three costs, as well as the total cost, as predicted by the simulation for each of the three trajectories. Note that, in each case, the cost associated with the strain energy is the greatest, while the cost associated with the inputs is the least. Since these costs are summed to get the total cost, we can see that inclusion of the input cost to make the optimization problem nonsingular has not significantly affected the solution. As expected, the cost associated with the strain energy, as well as the total cost, are least for the optimal trajectory. Table 4.6 shows the same costs as measured from experimental data. The value of the integral

Task \ Cost	Final	Strain	Inputs	Total
Polynomial	1.69e-4	1.39e-3	3.04e-12	1.56e-3
Constrained	2.73e-5	1.00e-3	2.42e-12	1.03e-3
Optimal	4.97e-8	8.21e-6	1.84e-10	8.26e-6

Table 4.5: Task 1: Objective function costs for simulated trajectories.

Task \ Cost	Final	Strain	Inputs	Total
Polynomial	1.42e-4	3.21e-3	1.29e-9	3.35e-3
Constrained	7.73e-4	2.51e-3	9.18e-10	3.28e-3
Optimal	1.09e-3	6.92e-4	3.65e-8	1.78e-3

Table 4.6: Task 1: Objective function costs for experimental motions.

Task \ Cost	Final	Strain	Inputs	Total
Polynomial	1.74e-9	1.27e-3	3.04e-12	1.27e-3
Constrained	3.24e-6	4.08e-4	8.54e-12	4.11e-4
Optimal	6.58e-8	1.78e-6	1.04e-10	1.84e-6

Table 4.7: Task 2: Objective function costs for simulated trajectories.

Task \ Cost	Final	Strain	Inputs	Total
Polynomial	2.18e-4	3.16e-3	1.00e-9	3.37e-3
Constrained	1.71e-3	1.81e-3	1.06e-9	3.51e-3
Optimal	1.73e-3	5.09e-4	3.14e-8	2.24e-3

Table 4.8: Task 2: Objective function costs for experimental motions.

costs were approximated by the trapezoidal rule. Similarly, the costs for Task 2 are given in Tables 4.7 and 4.8. As discussed in §4.3.3, because of the more difficult trajectory tracking requirements of the optimal trajectory, the experimental tasks have a final cost for the optimal trajectory which is higher than for the straight-line trajectories. However, both the strain and total costs remain significantly less than the corresponding costs for the other trajectories.

4.4 Summary

The purpose of this chapter was to demonstrate experimentally the extent to which a trajectory generated by the global optimization procedure improves the performance of the manipulator with respect to other trajectories and a given objective functional. Specifically, we compared three trajectories for two point-to-point motion tasks to evaluate the reduction in strain energy achieved by varying both the joint path and speed along that path. The experimental results presented in this chapter show that a significant reduction in manipulator vibration can be achieved by employing the global optimization procedure.

Chapter 5

Conclusions

Flexible manipulators are increasingly being used in applications where reduced weight can lead to performance or cost improvements. The inherent vibration must be minimized, however, to achieve the full performance potential. For many tasks, such as point-to-point maneuvers in joint space, the joint trajectory is not completely specified, but can be optimized with respect to some objective. The goal of this dissertation was to determine minimal vibration joint trajectories for point-to-point maneuvers of flexible manipulators.

To determine the vibration resulting from a particular joint trajectory, the motion was simulated according to a given manipulator dynamics model. This work was more general than most in the literature because it considered nonlinear models of multi-link flexible manipulators. An important feature of the numerical optimization method was that the dynamics simulation could be treated as a distinct modular component. That is, no approximation of the dynamics was made outside of the simulation, and no modification of the simulation was required. Thus, previous model validation work was preserved.

Knowing the vibration resulting from a particular trajectory, some quantitative measure is required to determine the joint trajectory which excites the least vibration. The optimization objective considered in this work was the integral of the strain energy evaluated over the time interval of the motion, also known as the *global* strain energy. The optimization problem was thus one of functional minimization, which allows comparison of entire trajectories. Only two other works have tackled the global minimization of vibration [LIU AND KUJATH, 1996; YAO AND CHENG, 1995A]. The work presented here surpasses the previous works both in the optimality of the solution and in its verification.

Chapter 2 presented the mathematical definition of the functional optimization problem and conducted an extensive review of the available numerical solution methods. With this background, an in-depth literature survey highlighted the scarcity of results for global minimization of vibration in flexible manipulators. The remainder of the chapter detailed the application of the Pontryagin Maximum Principle to the global minimization of the strain energy for flexible manipulators. The computational features of this indirect method were then compared to the other available methods and were justified for the particular problem under consideration.

Further details of the numerical implementation of the proposed method were given in Chapter 3. In particular, the optimal joint accelerations were represented by a cubic spline interpolation of knot values at discrete times. These knot values became the parameters to be optimized by a nonlinear numerical optimization algorithm. An advantage of the proposed formulation is that the gradient of the cost function is available to the optimization routine, improving the speed of convergence.

To verify the numerical optimization, solution trajectories were presented for three flexible manipulators undergoing point-to-point maneuvers. The joint path solutions

in each case were nontrivial yet could be physically justified. In addition, the derivation of the Coupling Map tool for path planning was extended to general flexible manipulator architectures. The globally optimal solutions were then evaluated in the context of the Coupling Map heuristics and were found to be in excellent agreement whenever the assumptions of the Coupling Map derivation were valid. This comparison supported the validity of the optimization approach as well as brought to light deficiencies of the Coupling Map. In particular, the results for a simple manipulator slewing in the presence of gravity show that the Coupling Map may be a useful tool for path planning but must be extended for general scenarios.

Finally, the effectiveness of the trajectory optimization scheme was further evaluated experimentally in Chapter 4 for high-speed, large-angle motions of an experimental spatial flexible manipulator. Such results are lacking in the literature, but are very important for assessing the utility of trajectory optimization in the presence of modelling and tracking errors. Again for verification purposes, the physical construction of the manipulator was chosen to admit a nontrivial zero-vibration joint path. Vibration tests were conducted to characterize the natural frequencies of the assembled system and to allow identification of model parameters. Using this model, the optimal joint trajectory was generated and compared to simpler trajectories, both in simulation and experimentally. Specifically, three trajectories were compared for two point-to-point motion tasks to evaluate the reduction in strain energy achieved by varying both the joint path and speed along that path. As for the simulation results, the experiments demonstrate that significant reductions in vibration are attainable by using the global optimization approach for trajectory generation.

To summarize, the contributions of this work are as follows. First, the combination of a nonlinear dynamics model and a broad class of comparison trajectories has

resulted in optimal solutions of greater generality than any of those reviewed for trajectory generation for flexible manipulators. Second, the extensive verifications and comparisons to another method as conducted here have not been presented by others in the literature. Finally, this work presents the only known experimental evaluation of a global trajectory optimization method for flexible manipulators. The results showed that significant reductions in vibration were possible using global optimization.

As a final comment, we briefly suggest possible avenues for further research into motion planning for flexible manipulators. As shown in Chapters 3 and 4, the Coupling Map and its heuristics provide a useful tool for quickly generating joint paths to reduce vibration. However, the Coupling Map is not valid for some common tasks, such as those for which accelerations resulting from gravity or high rotational speeds are dominant over joint accelerations. New maps and related heuristics would allow planning of joint paths for such common tasks. Rather than requiring a human operator to read the Coupling Map, automated path generation would be more suitable for online implementation and allow higher dimensional joint spaces to be considered. However, ensuring that the target joint configuration is attained would be difficult using only the local information provided by the map. A globally optimal trajectory generator that is locally computable would therefore be valuable for efficient motion planning. Appendix A proposes a technique for achieving these properties but the details remain incomplete and therefore remain a topic for future research.

References

- E. Adigüzel and H. Öz. Direct optimal control of nonlinear systems via Hamilton's law of varying action. *ASME Journal of Dynamic Systems, Measurement, and Control*, 117:262–269, 1995.
- Michael Athans and Peter L. Falb. *Optimal Control*. McGraw-Hill, Toronto, 1966.
- D. J. Bell and D. H. Jacobson. *Singular Optimal Control Problems*, volume 117 of *Mathematics in Science and Engineering*. Academic Press, New York, 1975.
- S. P. Bhat and D. K. Miu. Point-to-point positioning of flexible structures using a time domain LQ smoothness constraint. *ASME Journal of Dynamic Systems, Measurement, and Control*, 114:416–421, 1992.
- Wayne J. Book. Structural flexibility of motion systems in the space environment. *IEEE Transactions on Robotics and Automation*, 9(5):524–530, October 1993.
- Arthur E. Bryson, Jr. and Yu-Chi Ho. *Applied Optimal Control*. Hemisphere, Washington, D.C., revised edition, 1975.
- Richard L. Burden and J. Douglas Faires. *Numerical Analysis*. PWS-Kent Publishing, Boston, 4th edition, 1989.
- George F. Carrier and Carl E. Pearson. *Partial Differential Equation: Theory and Technique*. Academic Press, New York, 1976.
- Scott D. Cohen and Alan C. Hindmarsh. *CVODE User Guide*. Lawrence Livermore National Laboratory, October 1994. Available at Netlib: <http://www.netlib.org>.
- C. R. Dohrmann and R. D. Robinett. Robot trajectory planning via dynamic programming. In *Proceedings of the ISRAM*, volume 4, pages 75–81, August 1994.
- G. R. Eisler, R. D. Robinett, D. J. Segalman, and J. D. Feddema. Approximate optimal trajectories for flexible-link manipulator slewing using recursive quadratic programming. *ASME Journal of Dynamic Systems, Measurement, and Control*, 115:405–410, September 1993.

- I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Selected Russian Publications in the Mathematical Sciences. Prentice-Hall, Inc., Englewood Cliffs, N. J., 1963.
- Kenneth L. Hillsley and Stephen Yurkovich. Vibration control of a two-link flexible robot arm. *Dynamics and Control*, 3(3):261–280, July 1993.
- P. C. Hughes and G. B. Sincarsin. Dynamics of an elastic multibody chain: Part B—Global dynamics. *Dynamics and Stability of Systems*, 4(3 & 4):227–244, 1989.
- D. C. Hyland, J. L. Junkins, and R. W. Longman. Active control technology for large space structures. *Journal of Guidance, Control, and Dynamics*, 16(5):801–821, September–October 1993.
- F. John. *Partial Differential Equations*. Springer-Verlag, New York. 2nd edition. 1975.
- Gadi Kaplan. Industrial electronics. *IEEE Spectrum*, 36(1):68–72, January 1999.
- Herbert B. Keller. *Numerical Solution of Two Point Boundary Value Problems*, volume 24 of *Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, 1976.
- Seon-Jae Kim and Youn-Sik Park. Self-motion utilization for reducing vibration of a structurally flexible redundant robot manipulator system. In *Proceedings of the 1997 ASME Design Engineering Technical Conference*, pages 1–8, September 1997.
- W. Kim and J. Rastegar. Robot manipulator trajectory synthesis for minimal vibrational excitation. In *Proceedings of 1997 ASME DETC*, pages 1–10, 1997.
- K. Liu and M. R. Kujath. Trajectory optimization for a two-link flexible manipulator. *International Journal of Robotics and Automation*, 11(2):56–61, 1996.
- Jack Macki and Aaron Strauss. *Introduction to Optimal Control Theory*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, 1982.
- L. Meirovitch and Y. Chen. Trajectory and control optimization for flexible space robots. *Journal of Guidance, Control, and Dynamics*, 18(3):493–502, May–June 1995.
- Leonard Meirovitch. *Principles and Techniques of Vibrations*. Prentice-Hall, New Jersey, 1997.
- Akira Mohri, Pritam Kumar Sarkar, and Motoji Yamamoto. An efficient motion planning of flexible manipulator along specified path. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 1104–1109, 1998.

- I. H. Mufti. *Computational Methods in Optimal Control Problems*, volume 27 of *Lecture Notes in Operations Research and Mathematical Systems*. Springer-Verlag, Berlin, 1970.
- M. L. Nagurka and V. Yen. Fourier-based optimal control of nonlinear dynamic systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 112: 17–26, March 1990.
- M. Nahon, C. J. Damaren, A. Bergen, and J. S. Goncalves. A test facility for multi-armed space-based manipulators. *Canadian Aeronautics and Space Journal*, 41(4), 1995.
- T. Önsay and A. Akay. Vibration reduction of a flexible arm by time-optimal open-loop control. *Journal of Sound and Vibration*, 147(2):283–300, 1991.
- Ben J. Petterson, Rush D. Robinett, and Jill C. Warner. Parameter-scheduled trajectory planning for suppression of coupled horizontal and vertical vibrations in a flexible rod. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pages 916–921, 1990.
- Enid R. Pinch. *Optimal Control and the Calculus of Variations*. Oxford University Press, Toronto, 1993.
- P. M. Prenter. *Splines and Variational Methods*. John Wiley & Sons, Toronto, 1975.
- William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, 1991.
- Sanford M. Roberts and Jerome S. Shipman. *Two-Point Boundary Value Problems: Shooting Methods*, volume 31 of *Modern Analytic and Computational Methods in Science and Mathematics*. American Elsevier, New York, 1972.
- Martin H. Schultz. *Spline Analysis*. Series in Automatic Computation. Prentice-Hall, Englewood Cliffs, N.J., 1973.
- Miguel A. Serna and Eduardo Bayo. Trajectory planning for flexible manipulators. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pages 910–915, 1990.
- N. C. Singer and W. P. Seering. Preshaping command inputs to reduce system vibration. *ASME Journal of Dynamic Systems, Measurement, and Control*, 112: 76–82, 1990.

- Jefferson Steven Graham Stanway. Validation of a dynamics simulation for a structurally flexible manipulator. M.A.Sc thesis, Department of Mechanical Engineering, University of Victoria, 1996.
- S. Sundar and Z. Shiller. Optimal obstacle avoidance based on the Hamilton-Jacobi-Bellman equation. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 2424–2429, 1994.
- R. C. Thompson, J. L. Junkins, and S. R. Vadali. Near-minimum time open-loop slewing of flexible vehicles. *Journal of Guidance, Control, and Dynamics*. 12(1): 82–88, February 1989.
- Miguel A. Torres and Steven Dubowsky. Path-planning for elastically constrained space manipulator systems. In *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, volume 1, pages 812–817, 1993.
- Miguel A. Torres, Steven Dubowsky, and Attilio C. Pisoni. Path-planning for elastically-mounted space manipulators: Experimental evaluation of the coupling map. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 2227–2233, 1994.
- Hideki Tsuji, Hiroyuki Itoh, Yasushi Ogasawara, and Shinji Mitsuta. Vibration control by optimal trajectory with feedback system (suppression of residual vibration). *JSME International Journal*, 38(4):663–669, 1995.
- Q. Tu and J. Rastegar. Manipulator trajectory synthesis for minimal vibrational excitation due to the payload. *Transactions of the CSME*. 17(4A):557–566. 1993.
- Q. Tu, J. Rastegar, and J. R. Singh. Trajectory synthesis and inverse dynamics model formulation and control of tip motion of a high performance flexible positioning system. *Mechanism and Machine Theory*, 29(7):959–968, 1994.
- Charles Vaillancourt and Clément Gosselin. Compensating for the structural flexibility of the SSRMS with the SPDM. In *Proceedings of the International Advanced Robotics Program, Second Workshop on Robotics in Space*, 1994.
- M. J. G. van de Molengraft, F. E. Veldpaus, and J. J. Kok. An optimal estimation method for nonlinear models of mechanical systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 116:805–810, December 1994.
- Michiel J. Van Nieuwstadt and Richard M. Murray. Real time trajectory generation for differentially flat systems. Technical Report CIT/CDS 96-017 (revised). California Institute of Technology, May 1997. To appear in *International Journal of Robust and Nonlinear Control*.

- C. J. Van Vliet. Predicted and observed active damping performance of macro-micro manipulators. M.A.Sc. thesis, Department of Mechanical Engineering, University of Victoria, 1998.
- J. Van Vliet and I. Sharf. Development of a planar macro-micro manipulator facility: From design through model validation. *Canadian Aeronautics and Space Journal*, 44(1):40–50, March 1998.
- D. Wang and M. Vidyasagar. Control of a class of manipulators with a single flexible link—part I: Feedback linearization. *ASME Journal of Dynamic Systems, Measurement, and Control*, 113:655–661, December 1991.
- F. Xi and R. G. Fenton. Point-to-point quasi-static motion planning for flexible-link manipulators. *IEEE Transactions on Robotics and Automation*, 11(5):770–776, October 1995.
- Chih-Min Yao and Wen-Hon Cheng. Joint space trajectory planning for flexible manipulators. *Journal of Robotic Systems*, 12(5):287–299, 1995a.
- Chih-Min Yao and Wen-Hon Cheng. Least vibration end point trajectory for a flexible arm via inverse dynamics approach. *Proceedings of the National Science Council. ROC(A)*, 19(1):55–63, 1995b.
- K. Yoshida, D. N. Nenchev, P. Vichitkulsawat, H. Kobayashi, and M. Uchiyama. Experiments on the PTP operations of a flexible structure mounted manipulator system. In *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 246–251. IEEE, 1996. Osaka, Japan.

Appendix A

A Proposed Motion Planner

To find an optimal trajectory within an infinite-dimensional class of trajectories satisfying given boundary conditions requires the solution of a functional, or global, optimization problem. In general, however, iterative numerical algorithms for solving such problems incur excessive computational costs. For example, each evaluation of the objective functional requires a complete dynamics simulation, typically the most expensive part of the algorithm. Such costs preclude online global trajectory optimization except for simple cases such as those described by low-dimensional, linear, time-invariant dynamics equations.

Only the Coupling Map has been proposed to generate trajectories for flexible manipulators online [TORRES AND DUBOWSKY, 1993]. The Coupling Map presents local information for a region of joint configuration space and relies on a human operator to choose a joint path according to heuristics. The map information and the heuristics are derived for the purpose of obtaining joint paths which reduce vibration. This scheme is not optimal nor completely automated, however, and is restricted to two- or three-dimensional joint spaces in order to view the map.

This Appendix proposes a general method for generating optimal point-to-point trajectories. Since the full derivation of the algorithm is currently incomplete, the concepts and techniques of the derivation will be outlined here as documentation for further research. Potentially, the algorithm would:

1. Be applicable to a broad class of dynamical systems that includes flexible manipulators.
2. Determine a joint trajectory which combines a given point-to-point maneuver with active damping.
3. Determine a joint trajectory which is globally optimal in some sense yet would require only local computation.
4. Require only a single dynamics simulation and would therefore be suitable for online implementation.

Although no similar algorithms currently exist, an algorithm achieving these goals would be quite useful. Furthermore, the derivation technique discussed in the sequel is entirely novel.

A.1 Background

A powerful and popular technique for joint control of rigid manipulators is the *computed-torque* method. Assume the rigid robot dynamics are given by

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} \quad (\text{A.1})$$

where $\mathbf{q}, \boldsymbol{\tau} \in \mathbb{R}^n$ are the joint angles and torques respectively, and $\mathbf{M} \in \mathbb{R}^{n \times n}$ is the invertible mass matrix. The computed-torque method selects the joint torques $\boldsymbol{\tau}$ in terms of new inputs $\mathbf{u} \in \mathbb{R}^n$ as

$$\boldsymbol{\tau} = \mathbf{M}\mathbf{u} + \mathbf{h}. \quad (\text{A.2})$$

Substituting into the dynamics, the nonlinearities are exactly cancelled and a new system of decoupled, linear, time-invariant equations

$$\ddot{\mathbf{q}} = \mathbf{u} \quad (\text{A.3})$$

is obtained. Controller design techniques for LTI systems can now be applied to each joint independently to determine the elements of the new input vector \mathbf{u} , from which the torques can be calculated from Eqn. (A.2). Note that no approximations to the original dynamics were made.

The computed-torque method is an instance of the technique called *feedback linearization*. Feedback linearization involves the choice of a feedback and possibly a coordinate transformation to render the dynamics as an LTI system. For the above rigid manipulator example, no coordinate transformation was necessary, while the feedback is given by Eqn. (A.2). Specifically, the computed-torque method achieves *input-state* feedback linearization since the resulting dynamics, Eqn. (A.3), express an LTI relationship between the new inputs and the states. For rigid manipulators, the joint angles are typically measured outputs as well, so the same feedback achieves *input-output* feedback linearization.

Unfortunately, the situation for general flexible manipulators is not as straightforward. WANG AND VIDYASAGAR [1991] prove for a class of flexible manipulators with a single flexible link that no feedback and coordinate transformation exist to achieve input-state feedback linearization. However, for particular outputs, they show that

the dynamics are input-output feedback linearizable. Thus, the new inputs can easily be chosen to drive the outputs to their desired values (say zero). The disadvantage of input-output feedback linearization is that some of the states will become unobservable. The result is that even when all of the outputs are constrained to be zero, the unobservable states will continue to evolve according to their own dynamics, termed the *zero dynamics*. For example, if the measured outputs for a flexible manipulator are the joint angles, then the manipulator may still vibrate even when the joints are clamped at the desired angles. The stability of the zero dynamics thus becomes a fundamental issue in the control of flexible manipulators.

Controller design issues and the tasks intended for the flexible manipulator determine which sensors are necessary for satisfactory performance, although this choice may not be unique. For a given set of sensors, the purpose of the output variables in control system equations is to represent the sensor measurements. However, to improve the usefulness of their input-output feedback linearization method, WANG AND VIDYASAGAR [1991] choose outputs which are a function of the sensor measurements. The question then immediately arises as to how to choose the outputs such that they can be determined from the sensor measurements and that they can express the given task.

Unlike controller design, motion planning has no physical need for output variables. A motion planner employs a model (*e.g.* geometric, kinematic, or dynamic) to determine and evaluate possible motions of the manipulator. For flexible manipulators, numerical simulation of the dynamics model determines all of the state variables and allows a proposed joint motion to be evaluated. Thus, the motion planner has access to full state information and need not distinguish between those states that are sensed directly and those that are not.

Given the aforementioned feedback linearizability properties of flexible manipulator dynamics and the work of WANG AND VIDYASAGAR [1991], the key idea proposed in this Appendix is to introduce artificial outputs in order to apply the technique of input-output feedback linearization for motion planning. A primary constraint on the choice of outputs is that the given task must be expressible in terms of the outputs alone. We emphasize that the outputs will not necessarily have any physical interpretation, but are merely a mathematical device enabling the application of a novel motion planning technique.

A.2 Dynamics

We begin the discussion of the proposed method with the dynamics model. We assume the dynamics of the physical system under consideration can be written as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \quad (\text{A.4})$$

where $\mathbf{x}, \mathbf{f}(\mathbf{x}) \in \mathbb{R}^{2n}$, $\mathbf{u} \in \mathbb{R}^m$, and $\mathbf{g} \in \mathbb{R}^{2n \times m}$. For later use, we further assume that (A.4) can be partitioned as follows:

$$\dot{\mathbf{x}}_q = \mathbf{x}_v \quad (\text{A.5})$$

$$\dot{\mathbf{x}}_v = \begin{pmatrix} \mathbf{0}_{m \times 1} \\ \mathbf{f}_v(\mathbf{x}) \end{pmatrix} + \begin{bmatrix} \mathbf{I}_{m \times m} \\ \mathbf{G}(\mathbf{x}_q) \end{bmatrix} \mathbf{u} \quad (\text{A.6})$$

where $n = m + s$, and the quantities of (A.4) have been partitioned accordingly, so that $\mathbf{x}_q, \mathbf{x}_v \in \mathbb{R}^n$, $\mathbf{f}_v \in \mathbb{R}^s$, $\mathbf{0}$ and \mathbf{I} are the zero and identity matrices respectively, of the given dimensions, and $\mathbf{G}(\mathbf{x}_q) \in \mathbb{R}^{s \times m}$. The form given by (A.6) encompasses a broad class of systems, including of course mechanical systems such as flexible

manipulators of particular interest to us. To simplify the following presentation, we consider the two input case $m = 2$ and denote the two columns of \mathbf{g} by \mathbf{g}_1 and \mathbf{g}_2 .

A.3 Choosing the Outputs

While the dynamics (A.4) are given, the purpose of this work is to choose output functions

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) \quad (\text{A.7})$$

for $\mathbf{y} \in \mathbb{R}^2$. Although $\mathbf{h}(x)$ will be referred to as the “outputs” of the system, they are not necessarily related to any physical measurements. The manipulator task we consider here is point-to-point motion, rest-to-rest, where the desired final point is the origin in state space. Optimization of this motion will be deferred to future work, and hopefully can be achieved by extension of the framework presented here.

The first output function, $h_1(x)$, is chosen to be a positive-definite function of the state \mathbf{x} . In particular, we choose

$$h_1(\mathbf{x}) := \mathbf{x}^T \mathbf{W} \mathbf{x}$$

where $\mathbf{W} \in \mathbb{R}^{2n \times 2n}$ is a constant, symmetric, positive definite weighting matrix. Thus, once the single output y_1 is zero, all of the states must also be zero and the specified motion task is satisfied. Note also that there are no zero dynamics. For the moment, we will leave the other output function $h_2(\mathbf{x})$ to be determined. We then write the total time derivative of \mathbf{y} as

$$\dot{\mathbf{y}} = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} (\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}) \mathbf{u}) \quad (\text{A.8})$$

and make the following definitions:

$$\mathbf{p}^T := \frac{\partial h_2(\mathbf{x})}{\partial \mathbf{x}}, \quad (\text{A.9})$$

$$\mathbf{A} := \begin{bmatrix} \mathbf{x}^T \mathbf{W} \mathbf{g}_1 & \mathbf{x}^T \mathbf{W} \mathbf{g}_2 \\ \mathbf{p}^T \mathbf{g}_1 & \mathbf{p}^T \mathbf{g}_2 \end{bmatrix}, \quad (\text{A.10})$$

$$\mathbf{b} := \begin{pmatrix} \mathbf{x}^T \mathbf{W} \mathbf{f} \\ \mathbf{p}^T \mathbf{f} \end{pmatrix}, \quad (\text{A.11})$$

and

$$\mathbf{v} := \dot{\mathbf{y}} \quad (\text{A.12})$$

to rewrite (A.8) as

$$\mathbf{v} = \mathbf{A}(\mathbf{x}, \mathbf{p})\mathbf{u} + \mathbf{b}(\mathbf{x}, \mathbf{p}). \quad (\text{A.13})$$

Equation (A.13) represents a feedback law in terms of the new input \mathbf{v} , the state \mathbf{x} , and the partial derivatives \mathbf{p} , which we write explicitly as

$$\mathbf{u} = \mathbf{A}^{-1}(\mathbf{x}, \mathbf{p})(\mathbf{v} - \mathbf{b}(\mathbf{x}, \mathbf{p})). \quad (\text{A.14})$$

Applying this feedback gives the linear relationship between the new input and the outputs stated in (A.12). Motion planning can then be performed for the linear system (A.12) and mapped back to the original system by determining the inputs \mathbf{u} .

Since \mathbf{A} must be invertible to obtain the desired feedback, we would like it to remain nonsingular along the determined trajectory. In addition, for numerical implementation, \mathbf{A} must also remain reasonably well-conditioned. However, we first point out that \mathbf{A} is always singular at the origin $\mathbf{x} = \mathbf{0}$ and in fact throughout a higher dimensional manifold which contains the origin. We would therefore like to avoid the singularity as long as possible, hopefully coming “sufficiently close” to the origin before \mathbf{A} becomes ill-conditioned. Near the origin, we may be able to ensure that $\mathbf{v} - \mathbf{b}$ in (A.14) remains within the range space of \mathbf{A} or goes to $\mathbf{0}$ faster than \mathbf{A}

becomes singular. To avoid the singularities of \mathbf{A} we consider both the determinant $|\mathbf{A}|$ and the reciprocal of the condition number of \mathbf{A} , given by

$$r_A^2 = \frac{B - \sqrt{B^2 - 4|\mathbf{A}|^2}}{B + \sqrt{B^2 - 4|\mathbf{A}|^2}} \quad (\text{A.15})$$

where $0 \leq r \leq 1$ and

$$B := \|\mathbf{A}\|_F^2 = \sum_{i,j} A_{ij}^2. \quad (\text{A.16})$$

Note that, although both $|\mathbf{A}|$ and r_A are zero when \mathbf{A} is singular, they are independent measures of proximity to singularity as demonstrated when $\varepsilon \rightarrow 0$ by the matrix $\text{diag}(\varepsilon, \varepsilon)$ for which $|\cdot| = \varepsilon^2$, $r \equiv 1$, and by $\text{diag}(\varepsilon, 1/\varepsilon)$ for which $|\cdot| \equiv 1$, $r = \varepsilon^2$.

To avoid the singularities of \mathbf{A} , the motion planner utilizing the linear system dynamics (A.12) must have knowledge of the above measures. We therefore would like the second output function to be zero when either of these measures are zero.

First, we simplify r_A to

$$r_A^2 = \frac{4|\mathbf{A}|^2}{\left(B + \sqrt{B^2 - 4|\mathbf{A}|^2}\right)^2} \quad (\text{A.17})$$

and derive a lower bound

$$r_A = \frac{2|\mathbf{A}|}{B + \sqrt{B^2 - 4|\mathbf{A}|^2}} \geq \frac{2|\mathbf{A}|}{2B} = \frac{|\mathbf{A}|}{B}. \quad (\text{A.18})$$

Since the first row of \mathbf{A} is bounded if the state is bounded (for suitable \mathbf{g} and using y_1), we assume that

$$|A_{21}| + |A_{22}| > |A_{11}| + |A_{12}| \quad (\text{A.19})$$

and use the fact that $\|\mathbf{A}\|_F^2 \leq 4\|\mathbf{A}\|_\infty^2$ to replace B by $\bar{B} := 2(|A_{21}| + |A_{22}|)$. Now instead of r_A we consider the new measure

$$\bar{r}_A := \frac{|\mathbf{A}|}{\bar{B}} \quad (\text{A.20})$$

where $\bar{r}_A \approx 0$ signifies ill-conditioning of \mathbf{A} . Finally, we choose the second output to be zero when either measure is zero by setting

$$y_2 = \frac{1}{\frac{1}{\bar{r}_A} + \frac{1}{|\mathbf{A}|}} = \frac{|\mathbf{A}|\bar{r}_A}{|\mathbf{A}| + \bar{r}_A} = \frac{|\mathbf{A}|}{\bar{B} + 1}. \quad (\text{A.21})$$

Expanding, we get

$$y_2 = (A_{11} - 2\text{sgn}(A_{22})y_2)A_{22} - (A_{12} + 2\text{sgn}(A_{21})y_2)A_{21} \quad (\text{A.22})$$

$$= (\mathbf{x}^T \mathbf{W} \mathbf{g}_1 - 2\text{sgn}(\mathbf{p}^T \mathbf{g}_2)y_2)\mathbf{p}^T \mathbf{g}_2 \quad (\text{A.23})$$

$$- (\mathbf{x}^T \mathbf{W} \mathbf{g}_2 + 2\text{sgn}(\mathbf{p}^T \mathbf{g}_1)y_2)\mathbf{p}^T \mathbf{g}_1 \quad (\text{A.24})$$

where sgn is the signum function and $|x| = \text{sgn}(x)x$. Notice that this definition specifies the output function $h_2(\mathbf{x})$ implicitly as the solution to a *quasi-linear* partial differential equation (PDE), that is, a partial differential equation which is linear in the derivatives \mathbf{p} , but has coefficients which are possibly functions of \mathbf{x} and y_2 . The solution to this equation provides a means to control the proximity of the input transformation matrix \mathbf{A} to singularity.

We point out here that a third output might be chosen to represent an optimization objective, such as the strain energy, but that remains a topic for future research.

A.4 Solving for the Second Output Function

To solve the quasi-linear PDE for the second output function, we employ the well-known *method of characteristics* [JOHN, 1975; CARRIER AND PEARSON, 1976]. Without explaining the method, for now we will just state that the characteristic curves, parameterized by r , are given by the following set of $2n + 1$ ODE's:

$$\frac{dx_1}{dr} = 0 \quad (\text{A.25})$$

$$\vdots \quad (\text{A.26})$$

$$\frac{dx_n}{dr} = 0 \quad (\text{A.27})$$

$$\frac{dx_{n+1}}{dr} = -\mathbf{x}^T \mathbf{W} \mathbf{g}_2 - 2 \text{sgn}(\mathbf{p}^T \mathbf{g}_1) y_2 \quad (\text{A.28})$$

$$\frac{dx_{n+2}}{dr} = \mathbf{x}^T \mathbf{W} \mathbf{g}_1 - 2 \text{sgn}(\mathbf{p}^T \mathbf{g}_2) y_2 \quad (\text{A.29})$$

$$\frac{dx_{n+3}}{dr} = G_{1,2} (\mathbf{x}^T \mathbf{W} \mathbf{g}_1 - 2 \text{sgn}(\mathbf{p}^T \mathbf{g}_2) y_2) \quad (\text{A.30})$$

$$-G_{1,1} (\mathbf{x}^T \mathbf{W} \mathbf{g}_2 + 2 \text{sgn}(\mathbf{p}^T \mathbf{g}_1) y_2) \quad (\text{A.31})$$

$$\vdots \quad (\text{A.32})$$

$$\frac{dx_{2n}}{dr} = G_{n-2,2} (\mathbf{x}^T \mathbf{W} \mathbf{g}_1 - 2 \text{sgn}(\mathbf{p}^T \mathbf{g}_2) y_2) \quad (\text{A.33})$$

$$-G_{n-2,1} (\mathbf{x}^T \mathbf{W} \mathbf{g}_2 + 2 \text{sgn}(\mathbf{p}^T \mathbf{g}_1) y_2) \quad (\text{A.34})$$

$$\frac{dy_2}{dr} = y_2 \quad (\text{A.35})$$

where $G_{i,j}$ is the i, j th element of the matrix \mathbf{G} in (A.6). We now partition some of these quantities as:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ \mathbf{W}_{12}^T & \mathbf{W}_{22} \end{bmatrix}, \quad (\text{A.36})$$

$$\mathbf{g}_i = \begin{pmatrix} \mathbf{0}_n \\ \mathbf{g}_{iv} \end{pmatrix} \quad (\text{A.37})$$

and notice that

$$\mathbf{x}^T \mathbf{W} \mathbf{g}_i = \mathbf{x}_q^T \mathbf{W}_{12} \mathbf{g}_{iv} + \mathbf{x}_v^T \mathbf{W}_{22} \mathbf{g}_{iv}. \quad (\text{A.38})$$

We can thus rewrite equations (A.25-A.34) as

$$\frac{dx_q}{dr} = 0 \quad (\text{A.39})$$

$$\frac{dx_v}{dr} = C(x_q)x_v + D(x_q)x_q - 2 \begin{bmatrix} \text{sgn}(\mathbf{p}^T \mathbf{g}_1) \\ \text{sgn}(\mathbf{p}^T \mathbf{g}_2) \\ G_{1,2} \text{sgn}(\mathbf{p}^T \mathbf{g}_2) + G_{1,1} \text{sgn}(\mathbf{p}^T \mathbf{g}_1) \\ \vdots \\ G_{n-2,2} \text{sgn}(\mathbf{p}^T \mathbf{g}_2) + G_{n-2,1} \text{sgn}(\mathbf{p}^T \mathbf{g}_1) \end{bmatrix} y_2 \quad (\text{A.40})$$

where, by choosing $\mathbf{W}_{12} = \mathbf{W}_{22}$, we get $D(x_q) = C(x_q)$, and

$$C(x_q) = \begin{bmatrix} -\mathbf{g}_{2v}^T \mathbf{W}_{22}^T \\ \mathbf{g}_{1v}^T \mathbf{W}_{22}^T \\ G_{1,2} \mathbf{g}_{1v}^T \mathbf{W}_{22}^T - G_{1,1} \mathbf{g}_{2v}^T \mathbf{W}_{22}^T \\ \vdots \\ G_{n-2,2} \mathbf{g}_{1v}^T \mathbf{W}_{22}^T - G_{n-2,1} \mathbf{g}_{2v}^T \mathbf{W}_{22}^T \end{bmatrix}. \quad (\text{A.41})$$

It is important to point out here that C always has rank 2, since all rows are linear combinations of the first two rows, which are always independent by the structure of $\mathbf{g}(\mathbf{x})$ shown in (A.6).

A.4.1 Solving the Characteristic Equations

To solve the characteristic equations we introduce a coordinate transformation. This transformation will in fact be the coordinate transformation permitted by the feedback linearization technique in addition to the input transformation already discussed. Recall that we began with $2n$ states \mathbf{x} . The new coordinates are $(\mathbf{s}^T, r)^T$ where $\mathbf{s} \in \mathbb{R}^{2n-1}$ is partitioned as

$$\mathbf{s} = (\mathbf{s}_q^T, \mathbf{s}_v^T, s)^T \quad (\text{A.42})$$

and $\mathbf{s}_q \in \mathbb{R}^n$, $\mathbf{s}_v \in \mathbb{R}^{n-2}$ and s is a scalar distinguished by lack of subscript or boldface.

The initial conditions for the characteristic ODE's are specified when $r = 0$, which we parameterize by

$$\mathbf{x}_q|_{r=0} = \mathbf{s}_q \quad (\text{A.43})$$

$$\mathbf{x}_v|_{r=0} = \beta(\mathbf{s}_q, \mathbf{s}_v, s) \quad (\text{A.44})$$

with β to be determined later.

Solving the characteristic equations (A.39) using (A.43) gives

$$\mathbf{x}_q = \mathbf{s}_q. \quad (\text{A.45})$$

We can also solve (A.35) to get

$$y_2 = c_9(\mathbf{s}_q, \mathbf{s}_v, s)e^r \quad (\text{A.46})$$

where the function $c_9(\mathbf{s}_q, \mathbf{s}_v, s)$ will be chosen later. However, solving (A.40) requires some more definitions first. Define \mathbf{P} to be the coordinate transformation matrix which renders \mathbf{C} in real, Jordan canonical form, denoted by \mathbf{J} , so that

$$\mathbf{J} = \mathbf{P}^{-1}\mathbf{C}\mathbf{P}. \quad (\text{A.47})$$

Because we know that \mathbf{C} has rank 2,¹ we can write

$$\mathbf{J} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_2 \end{bmatrix} \quad (\text{A.48})$$

where $\mathbf{J}_2 \in \mathbb{R}^{2 \times 2}$. Furthermore, we assume that the non-zero eigenvalues of \mathbf{C} are $\pm\alpha i$. This implies that

$$\mathbf{J}_2 = \begin{bmatrix} 0 & -\alpha \\ \alpha & 0 \end{bmatrix}. \quad (\text{A.49})$$

¹We also assume, for now, that the eigenvalue 0 is not deficient, that is, it has $n - 2$ independent eigenvectors.

For brevity we simply state the general solution:

$$\mathbf{x}_v = e^{C_r} \boldsymbol{\beta} + (e^{C_r} - \mathbf{I}_n) \mathbf{P} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix} \mathbf{P}^{-1} \mathbf{s}_q \quad (\text{A.50})$$

$$+ 2c_9 (e^{C_r} - e^r \mathbf{I}_n) \mathbf{P} (\mathbf{I} - \mathbf{J})^{-1} \mathbf{P}^{-1} \mathbf{G}_v \begin{bmatrix} \text{sgn}(\mathbf{p}^T \mathbf{g}_1) \\ \text{sgn}(\mathbf{p}^T \mathbf{g}_2) \end{bmatrix} \quad (\text{A.51})$$

where \mathbf{C} and \mathbf{P} are functions of \mathbf{s}_q because of (A.45).

A.5 Coordinate Transformation Conditioning

As for the input transformation, we must ensure that it is possible to invert the coordinate transformation equations given by (A.45) and (A.50), at least along the given trajectory. Then we are assured of the existence, at least locally, of an isomorphism from the parameters $(\mathbf{s}_q, \mathbf{s}_v, s, r)$ to the state \mathbf{x} and of a solution for y_2 in terms of \mathbf{x} . A necessary condition is that the Jacobian matrix

$$\frac{\partial \mathbf{x}}{\partial (s, r)} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \frac{\partial \mathbf{x}_v}{\partial \mathbf{s}_q} & \begin{bmatrix} \frac{\partial \mathbf{x}_v}{\partial (s_v, s)} & \frac{\partial \mathbf{x}_v}{\partial r} \end{bmatrix} \end{bmatrix} \quad (\text{A.52})$$

be non-singular, which implies that we must ensure that the matrix in the bottom right entry also is non-singular. It can be expanded to:

$$\begin{bmatrix} \frac{\partial \mathbf{x}_v}{\partial (s_v, s)} & \frac{\partial \mathbf{x}_v}{\partial r} \end{bmatrix} = e^{C_r} \begin{bmatrix} \frac{\partial \boldsymbol{\beta}}{\partial (s_v, s)} & \mathbf{C}(\boldsymbol{\beta} + \mathbf{x}_q) \end{bmatrix}. \quad (\text{A.53})$$

We therefore would like to choose $\boldsymbol{\beta}$ so that this matrix is non-singular. We choose, without loss of generality,

$$\boldsymbol{\beta} = -\mathbf{s}_q + \mathbf{P} \begin{bmatrix} \mathbf{s}_v \\ k_1 \\ k_2 \end{bmatrix}. \quad (\text{A.54})$$

We would now like to determine the functions $k_1(\mathbf{s})$ and $k_2(\mathbf{s})$ in such a way that both the determinant and the conditioning of the matrix $\frac{\partial \mathbf{x}}{\partial (\mathbf{s}, r)}$ are well behaved. The extent to which these properties can be arbitrarily assigned is not known, but the following analysis provides a step towards a more complete understanding.

Substituting (A.54) into (A.53) and using the fact that the columns of \mathbf{P} are independent eigenvectors² of \mathbf{C} , and that $e^{\mathbf{C}r} = \mathbf{P}e^{\mathbf{J}r}\mathbf{P}^{-1}$, we get

$$\begin{bmatrix} \frac{\partial \mathbf{x}_v}{\partial (\mathbf{s}_v, s)} & \frac{\partial \mathbf{x}_v}{\partial r} \end{bmatrix} = \mathbf{P}e^{\mathbf{J}r} \left[\mathbf{P}^{-1} \frac{\partial \boldsymbol{\beta}}{\partial (\mathbf{s}_v, s)} + 2 \frac{\partial c_9}{\partial (\mathbf{s}_v, s)} (\mathbf{I} - e^r e^{-\mathbf{J}r}) \begin{bmatrix} \mathbf{0} \\ w_1 \\ w_2 \end{bmatrix} \right] \quad (\text{A.55})$$

$$\mathbf{J}\mathbf{P}^{-1} (\boldsymbol{\beta} + \mathbf{s}_q) + 2c_9 (\mathbf{J} - e^r e^{-\mathbf{J}r}) \begin{bmatrix} \mathbf{0} \\ w_1 \\ w_2 \end{bmatrix} \quad (\text{A.56})$$

where

$$\begin{bmatrix} \mathbf{0} \\ w_1 \\ w_2 \end{bmatrix} := (\mathbf{I} - \mathbf{J})^{-1} \mathbf{P}^{-1} \mathbf{G}_v \begin{bmatrix} \text{sgn}(\mathbf{p}^T \mathbf{g}_1) \\ \text{sgn}(\mathbf{p}^T \mathbf{g}_1) \end{bmatrix} \quad (\text{A.57})$$

We continue to simplify to get

$$\begin{bmatrix} \frac{\partial \mathbf{x}_v}{\partial \mathbf{s}_v} & \frac{\partial \mathbf{x}_v}{\partial (s, r)} \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{I}_{n-2} & \mathbf{0} \\ \mathbf{0} & e^{\mathbf{J}2r} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{n-2} & \mathbf{0}_{(n-2) \times 2} \\ \frac{\partial \mathbf{k}}{\partial \mathbf{s}_v} + 2 \frac{\partial c_9}{\partial \mathbf{s}_v} (\mathbf{I} - e^r e^{-\mathbf{J}r}) \begin{bmatrix} \mathbf{0} \\ w_1 \\ w_2 \end{bmatrix} & \mathbf{K} \end{bmatrix} \quad (\text{A.58})$$

²The vectors w_i may be *generalized* eigenvectors if the multiple zero eigenvalue of \mathbf{C} is deficient.

where

$$\mathbf{K} := \begin{bmatrix} k'_1 + cc'_9 & -\alpha k_2 + ac_9 \\ k'_2 + dc'_9 & \alpha k_1 + bc_9 \end{bmatrix}, \quad (\text{A.59})$$

primes ' are with respect to s , and

$$\begin{bmatrix} \mathbf{0} \\ a \\ b \end{bmatrix} : = 2 (\mathbf{J} - e^r e^{-\mathbf{J}r}) \begin{bmatrix} \mathbf{0} \\ w_1 \\ w_2 \end{bmatrix}, \quad (\text{A.60})$$

$$\begin{bmatrix} \mathbf{0} \\ c \\ d \end{bmatrix} : = 2 (\mathbf{I} - e^r e^{-\mathbf{J}r}) \begin{bmatrix} \mathbf{0} \\ w_1 \\ w_2 \end{bmatrix}. \quad (\text{A.61})$$

We can now see that the conditioning and invertibility of the above coordinate transformation Jacobian depends critically on the conditioning and invertibility of the lower right 2×2 submatrix \mathbf{K} of the last matrix factor of (A.58). The next step in the development of the proposed algorithm is to choose the unknowns $k_i(s_q, s_v, s)$, $i = 1, 2$, and $c_9(s_q, s_v, s)$ such that singularities of the coordinate transformation can be avoided.

A.6 Summary

This Appendix has outlined the development of a motion planning scheme based on the input-output feedback linearization technique. The outputs are not physical measurements but are mathematical functions chosen to express a given motion task and to enable application of the feedback linearization technique. The first output is a positive definite function of all of the states. Driving this output to zero accomplishes the point-to-point maneuver with active damping and eliminates the zero

dynamics. The feedback linearization requires a transformation of the inputs and possibly a coordinate transformation of the states. The second output is chosen to ensure that the input transformation remains invertible during the motion. A coordinate transformation is then introduced to solve for the second output, but work remains to demonstrate that this transformation can be made nonsingular throughout the trajectory. A third output should be considered in future work to represent an optimization objective such as the strain energy. Trajectory generation using the proposed method would require only a single numerical integration of the states together with the new coordinates, resulting in an efficient method suitable for online implementation.