An Optimization-Based Approach for the Synthesis of Practical Mechanisms

by

XIAOQI FANG
B.Sc. in Engineering, Jilin University of Technology, 1982

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF APPLIED SCIENCE
in the
Department of Mechanical Engineering

We accept this thesis as conforming
to the required standard

[redacted]

Dr. R. P. Podhorodeski, Supervisor (Dept. of Mechanical Engineering)

[redacted]

Dr. C. Konzelman, Department Member (Dept. of Mechanical Engineering)

[redacted]

Dr. J. Collins, Outside Member (Dept. of Electrical and Computer Engineering)

[redacted]

Dr. C. Dayson, External Examiner (Institute for Machinery Research, NRC)

University of Victoria

Supervisor: Dr. Ron P. Podhorodeski

Abstract

This thesis presents an optimization-based approach for the synthesis of planar mechanisms. The approach is practical since it can be applied to several mechanism types and a variety of mechanism synthesis problems. The mechanisms synthesized are practical since convergence to certain categories of mechanisms within specific types is ensured and the consideration of arbitrary performance objectives in the optimization is allowed.

A least p-th algorithm with tasks specified by upper and lower limit constraints on output displacement components is used to generate initial feasible mechanisms. A sequentially unconstrained minimization technique (SUMT) is used to ensure motion feasibility during mechanism optimization for specified objectives. The concept of using relative displacements for specifying upper and lower displacement limits is introduced. Basing the motion specification on relative displacements is effective since it eliminates the global location of the mechanism from the search parameters. Furthermore it is effective since it allows initial input displacements to be considered as search parameters.

A direct search method is used with the least p-th and SUMT optimizations. While slower to converge than gradient based methods, direct search methods allow the consideration of arbitrary objective functions. Transformations of unconstrained search variables to constrained mechanism parameters are utilized to allow unconstrained searching. Transforms based on redefining link length limits, incorporating "Grashof-Criteria" considerations concurrently with transformation of unconstrained search variable set values, are introduced for link length parameters. The transforms facilitate effective optimization allowing searching and convergence within a preferred mechanism sub-type.

Two example problems, one path generation and the other motion generation, and a case study for a digger mechanism (motion generation) are presented demonstrating the effectiveness of the developed approach.

[redacted]

Dr. R. P. Podhorodeski, Supervisor (Dept. of Mechanical Engineering)

[redacted]

Dr. C. Konzelman, Department Member (Dept. of Mechanical Engineering)

[redacted]

Dr. J. Collins, Outside Member (Dept. of Electrical and Computer Engineering)

[redacted]

Dr. C. Dayson, External Examiner (Institute for Machinery Research, NRC)

Table of Contents

List of Tables

List of Figures

Acknowledgements

I wish to thank my supervisor, Dr. Ron Podhorodeski, for the many hours spent discussing kinematic theory and optimization methods which form the foundation of this work. His broad knowledge of mechanisms and mathematics were a great asset in gaining the knowledge and understanding required to carry out the research. Furthermore, his fine attitude, positive criticism, and ever-present willingness to help aided me greatly in completing this thesis.

I also wish to express my gratitude to members of the Tribology and Mechanics Laboratory of NRC in Vancouver for their assistance.

I am also very grateful to my parents and my wife whose encouragement and confidence in me have lead me to complete this and many other endeavors.

Chapter 1

Introduction

1.1 Mechanism Elements and Terminology

The terminology used throughout this discussion should be presented first. Although not consistent in all of the literature, the terminology that is introduced will be used in all of the remaining chapters.

A *mechanism* is a system of bodies designed to convert motion of, and/or force on, one or several bodies into constrained motions of, and/or force on, other bodies.

A *linkage* is a kinematic chain whose joints are equivalent to lower pairs only.

A *coupler* is a link that is not connected directly to the frame.

A *input/driving link* is a link whereby motion and force are imparted to a mechanism.

A *output/driven link* is a link from which required forces and motions are obtained.

A *bar* is a link that carries only revolute joints.

Structural error is the difference between desired and generated motion of a synthesized mechanism.

1.2 Processes of the Synthesis for Mechanisms

Synthesis involves the systematic design of mechanisms for a given performance. In other words, a mechanism is to be found that achieves desired motion and/or force outputs and desired performance characteristics. This is the essence of mechanism synthesis.

In order to present the objectives of the thesis, processes used for the synthesis of mechanism must be first reviewed. The basic stages of mechanisms synthesis include: type synthesis; dimensional synthesis; and analysis. Type synthesis is concerned with determining the suitable mechanism type or "topology" without regard to the dimensions of the links [29]. Examples of mechanism types include linkages, cams, belts and pulleys, gears, and servo-mechanisms. Type synthesis is to determine the elements, mechanisms or configuration that in some combination has the potential of resulting in a design that best satisfies the need. Type synthesis is a key step involving inventiveness and creativity.

Once a potential mechanism type is chosen, dimensional synthesis must be performed. Dimensional synthesis is the process of determining the physical data, such as the dimensions of the bodies, the positions of revolute joints, and the orientation of translational joints. Dimensions must be determined to satisfy the prescribed desired kinematic behaviour/task of the mechanism. Traditionally, one of the following categories of kinematic behaviour/task is required [39]:

(1) Function Generation: The motion of a body (the output link) is a specified function of the motion of another body (the input link). The flow-rate-indicator

mechanism illustrated in Figure 1.1 is an example of a function generator. Note that for this particular mechanism a four-bar mechanism and a slider mechanism are joined in series. The objective of this linkage is to provide a measure of the flow rate through the weir. The input is provided by the height of the water level.

(2) Path Generation: A point on a body is to follow a specified path. An example of a path generator is the film-advance mechanism of a movie camera as shown in Figure 1.2. The rotations of the input link A-A0 in the diagram are coordinated to locations of the path traced by the point C.

(3) Motion Generation: A body is to pass through a series of prescribed positions and orientations. An example of a motion generator is the scoop mechanism for a front loader illustrated in Figure 1.3. Both the location and angular orientation of the scoop are important in this design.

In addition to the tasks mentioned above, other tasks that are achieved through a synthesis process may include kinematic and dynamic characteristics such as velocity and/or acceleration components, and transmission angle characteristics associated with the synthesized mechanism.

The final step in the synthesis process described here is to assess how well the designed mechanism satisfies the final requirements. For example, the designed mechanism must be capable of motion through all of the design positions. This is referred to as the ability of assembly of the mechanism. A mechanism that can be assembled at each of the design positions may still have undesirable mechanism properties and for this reason must be analyzed and evaluated to check if further synthesis is required to achieve an appropriate mechanism.

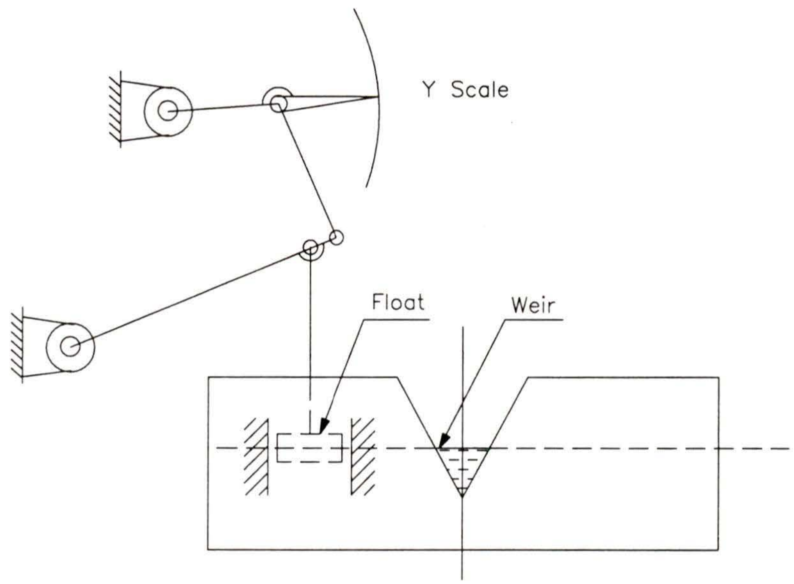

Figure 1.1: Flow-rate-indicator Mechanism

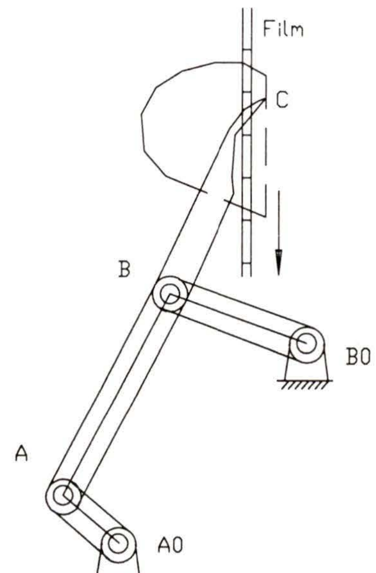

Figure 1.2: Film-advance Mechanism of a Movie Camera or Projector

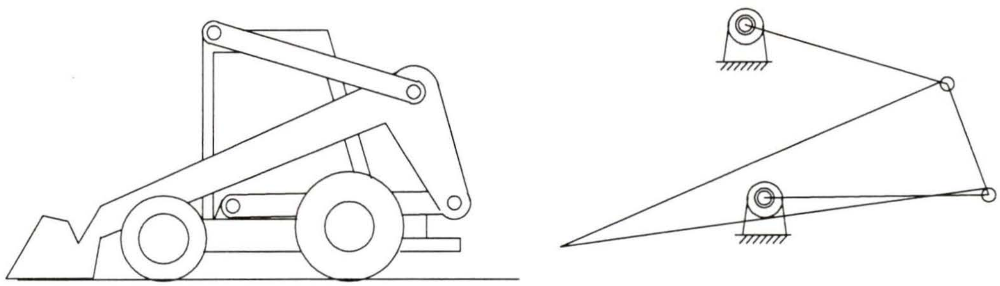

Figure 1.3: Front-loader Four-bar Mechanism

1.3 Previous Approaches to Synthesis – Literature Review

In the past, a number of different techniques have been employed for the synthesis of mechanisms. In traditional approaches, the solution method is based on graphical/geometrical and/or analytical methods. Later with the proliferation of high-speed computers and their integration into mechanism analysis and synthesis, a wide variety of numerical optimization methods have been developed for the synthesis of mechanisms.

Many graphical techniques have been developed for specific mechanism and task classes. For example, using graphical techniques, Hrones and Nelson(1951) [18] and Zhang et al.(1984) [46] have developed atlases of coupler-point motions for four-bar and geared five-bar linkage mechanisms. Summaries of graphical techniques are provided in easily available mechanism textbooks, e.g. [10]. These graphical techniques can provide the designer with a straightforward method of design. However these approaches are limited to simple planar mechanisms and have accuracy limitations due to drawing error. Also, to achieve suitable results, the graphical approach may have to be repeated many times.

Burmeister was the first to develop analytical methods for the synthesis of mechanisms (1888) [6]. Extension and application of Burmeister's theories can be found in the works by Beyer(1963) [5], and Primrose et al.(1964) [33]. The analytical methods in practice today are mostly based on either algebraic methods, i.e., Hanson(1964) [16], displacement matrix methods, i.e., Rigelman(1978) [37], or complex-number methods, i.e., Seireg(1984) [40]. In these analytical approaches, the mechanism synthesis problem is solved by defining a set of precision positions (precision points) for the desired mechanism output and equating analytical expressions to render dimensions that satisfy the precision points exactly.

Analytical methods of synthesis are suitable for automatic computation and have the advantages of accuracy and repeatability. Once a mechanism is modeled mathematically and coded for a computer, mechanism parameters can be easily manipulated to create new solutions without further programming. However, a major drawback in using the analytical methods is that the number of prescribed precision points is limited by the type of mechanism selected. Consequently, while the designed mechanism may exactly satisfy these precision points, there can be significant errors in the overall output between the precision points. Furthermore, using the analytical methods, there can be other problems such as: branching (the synthesized linkage cannot be moved through all of the design positions without changing the assembly branch mode); order (the order in which the output link traverses the design positions is different from the desired order); and Grashof related (the resulting linkage is non-Grashof, i.e., not capable of full rotation of the input link). In addition, the consideration of further performance objectives is not easily accommodated.

An alternative approach is approximate linkage synthesis, which employs numerical optimization techniques allowing any number of design points to be included and allowing the consideration of additional design objectives. The subject of approximate linkage synthesis has been studied by several researchers. Many of the optimization techniques treat mechanism synthesis as a problem in nonlinear programming. In these methods, approximate optimal solutions are obtained by minimizing an objective function, which is defined as a weighted norm of the errors between the desired and generated positions and orientations of the coupler link. It appears that Levitskii and Shakvazian in 1954 were the first to propose "error-norm"-based objectives in the realm of linkage optimization, although their original paper in Russian was not translated until 1960 [23]. Independently, Lewis and Gyorgy(1967) [25] applied least-square error objectives to solve problems of path generation with planar four-bar linkages. Fox and Willmert (1967) [12] optimized a four-bar path-generating linkage using a least-square error objective by means of a penalty function method. Nolle (1967) [26] introduced the concept of a multi-dimensional solution surface, representing the relationship between equality of solution and independent parameters, and developed an algorithm for solving optimum synthesis problems of four-bar mechanisms as function generators.

Modified methods based on the concept of the minimized error-norm criterion were later developed by Tomas (1968) [44] and Nolle and Hunt (1971) [27] to determine optimal dimensions of various path-generating and function generating mechanisms. An optimization method for synthesis of path-generating planar mechanisms based on a method of superposition has been presented by Bagci and Lee (1975) [4]. A review of some of the general-purpose optimization techniques can be found in the works by

Seireg (1973) [40], Root and Ragsdell (1976) [38], and Fox and Gupta (1973) [11].

Recently, a number of researchers have looked at the application of several other more efficient optimization techniques for the synthesis and design of mechanisms. A selective-precision synthesis approach that is suitable for function, path and motion generation was presented by Kramer and Sandor (1975) [22]. In the work of [22], the kinematic chain is considered to be composed of dyads, "accuracy neighbourhoods" are constructed around the precision points, and a mechanism that goes through all of the accuracy neighbourhoods is found with a nonlinear optimization technique. Rao (1979) [35] applied geometric programming to the synthesis of planar four-bar function-generating mechanisms. The objective function in the work of Rao was taken as the sum of the squares of the structural errors at a number of design positions. Ion and Cezar (1979) [19] proposed a general method based on matrix algebra for synthesis of path-generating mechanisms. Paradis and Willmert (1983) [30] used the Gauss constrained method for solving problems with linear constraints and objective functions which are the sum of squared quantities. Hammond and Johnson (1988) [15] have presented a method of alternative formulations for mechanism design problems with monotonicity and with a small number of design variables. Angeles et al. (1988) [1] developed a method of optimum synthesis of planar RRRR linkages for path generation where the method modeled the problem as one of an unconstrained nonlinear least-square optimization. Akhras and Angeles (1990) [2] combined continuation methods with the unconstrained nonlinear least-square technique of [1] for motion generation by four-bar mechanisms. In this later work, inequality constraints, such as space limitation, were avoided by using the Cartesian coordinates of the joint centers as design variables, and the difference between desired and generated point coordinates was minimized by applying the Newton-Gauss method.

In all of the above optimization techniques, the optimization process was carried out based on a single objective, such as guiding the mechanism through a set of prescribed design points, and the goal was to best solve for that one objective subject to a set of constraints. However, in many real problems, there could be several objectives and the designer may find it difficult to select one single objective. It would be advantageous if all the objectives could be introduced directly in the optimization process. Rao and Hati (1979) [36] introduced one such multiple-objective optimization technique using a game-theory approach. More recently, Krishnamurty and Turcic (1992) [21] introduced a multi-objective optimization technique based on nonlinear goal programming for determining the optimal solutions to general planar mechanism synthesis problems with priority levels for each of the objectives.

Optimization-based methods proposed for mechanism synthesis generally fall into one of the following three categories: 1) direct methods, which do not require any derivatives and easy to implement, but they are slow to converge; 2) gradient methods, which require the computations of first derivatives; and 3) Hessian methods, which require the second derivatives and converge quickly (at least theoretically), but increase the algebraic and computational complexity of the problem dramatically. The gradient and Hessian matrix of a function $f(\mathbf{x})$ of n variables x_i, i=1, $\cdots$, n at a point $\mathbf{x}^*$ are a column vector and an n $\times$ n matrix, respectively, i.e., $\Delta f(\mathbf{x}^*) = [\partial f(\mathbf{x}^*)/\partial x_i]^T$ and $\mathbf{H} = [\partial^2 f(\mathbf{x}^*)/\partial x_i \partial x_j]$, where i,j = 1 to n, and $\partial f(\mathbf{x}^*)/\partial x_i$ and $\partial^2 f(\mathbf{x}^*)/\partial x_i \partial x_j$ are the first and second partial derivatives of the function $f(\mathbf{x})$ evaluated at the given point $\mathbf{x}^*$. Generation of algebraic expressions for the gradient vector and Hessian matrix of the desired design objective can be very difficult. This poses a serious disadvantage for some objective functions and hampers consideration of arbitrary objectives. This is a disadvantage for achieving a general mechanism design tool.

1.4 Research Objective

Several features are common to all the previous approximate synthesis methods. First, the linkage error is defined as a weighted norm of the differences between the generated and prescribed variables defining position and orientation of the coupler link. Secondly, a set of simultaneous nonlinear equations is formed that represents the relationships between dependent and independent parameters and variables. Finally, an optimization method is applied to find the solution iteratively. Generally, the methods have been applied to four-bar mechanisms.

However, some mechanism design associated problems have not been solved very well. The most critical part in the aforementioned optimization procedure comes about when bounds on certain linkage parameters or linkage variables must be introduced. Such bounds can arise due to space limitations or due to desired requirements of maximum ratio of link length (dimensional balance). Furthermore, when designing a mechanism for an application, a specific mechanism type may be desired. Often it is desirable to have mechanisms capable of full-input link rotation (a Grashof mechanism). For example, in many applications requiring the guidance of a point through a particular path (path generation) or requiring the guidance of a rigid body through a path (motion generation), it is preferable to have a single-direction continuous input.

In Tandirci et al. (1991) [43], discriminants were derived from consideration of a displacement closure and Freudenstein's equation, and a mechanism was shown to be a Grashof mechanism if the associated discriminants remained positive for all configurations. The magnitude of these discriminants could be utilized to form a penalty function for non-Grashof mechanisms. However, this would not ensure sub-type specific searching and convergence within the family of Grashof mechanisms. Generally, with the exception of the work of Krishnamurty and Turcic (1992) [21], the above works have not been concerned with the issue of guaranteeing convergence to specific sub-types. In the work of Krishnamurty and Turcic (1992) [21], a high priority was placed on a sub-objective, formed from Grashof's criteria for four-bars, to ensure convergence to a Grashof mechanism. Adding a further sub-type specific objective with appropriate weighting priority could ensure sub-type specific convergence. Unless extreme care was taken, searching outside of the specific sub-type would occur. A technique allowing direct sub-type specific searching would be an asset.

The research of this thesis is aimed at development of an optimization-based approach for the synthesis of practical mechanisms. Specifically, the objectives of this thesis are to develop an optimization-based synthesis methodology allowing:

1) the consideration of several practical planar mechanism types;

2) the consideration of arbitrary performance objectives in the optimization;

3) searching and convergence within specific categories of mechanisms within specific sub-types.

1.5 Summary of Contents of Thesis

In the work presented in this thesis, a method is presented which ensures both the searching within and convergence to a specific mechanism sub-type. Transforms incorporating "Grashof-criteria" considerations are developed to allow unconstrained searching within specific four-bar and six-bar mechanism sub-types in Chapter 3. The transformations facilitate the simple, yet versatile, direct-search optimization-based synthesis approach for mechanisms described in Chapter 2. Some example applications are included in Chapter 2 to demonstrate the methodology effectiveness. A case study (a deep digging implement) is investigated in Chapter 4 and the thesis ends with conclusions in Chapter 5.

Chapter 2

Mechanism Optimization Approach

2.1 Basic Optimization Problem

A general optimization problem can be expressed as:

$$
\begin{aligned}
\text{Minimize}: f_{obj} &= f(\mathbf{X}, \mathbf{Y}) \text{ with respect to } \mathbf{X} \\
\text{Subject to}: g_i(\mathbf{X}, \mathbf{Y}) &= 0;\ i = 1, m \\
h_j(\mathbf{X}, \mathbf{Y}) &\leq 0;\ j = 1, n
\end{aligned}
\tag{2.1}
$$

For mechanism synthesis based on optimization, the objective function f_{obj} is a measure that quantifies the performance of the mechanism and may represent specified motion requirements, dynamic properties, mechanism characteristics (e.g., size, transmission angle, etc.), and so on. $\mathbf{X}$ of eqn. (2.1) represents the set of design variables. For a mechanism, $\mathbf{X}$ includes variables such as associated lengths and gear ratios, and initial conditions such as initial input displacement(s). $\mathbf{Y}$ of eqn. (2.1) represents fixed (non-design) parameters of the mechanism. The equality and inequality constraints of eqn. (2.1) may exist on the design variables or may be related to mechanism performance characteristics.

2.2 Mechanism Optimization Method

2.2.1 Associated Mechanism Design Parameters

The design parameters for the synthesis of a mechanism are the mechanism parameters associated with the input-output relationship of the mechanism. As such, the parameters are dependent on the type of mechanism and the task under consideration. Mechanism tasks as mentioned in Chapter 1 can be: i) function generation; ii) path generation; iii) motion generation (rigid body guidance). The design variables for a four-bar mechanism will be identified and presented for different tasks in section 2.4 and similarly the design variables can be assigned for other types of mechanisms. Appendix A presents such design variables for six-bar and geared five-bar mechanisms.

2.2.2 Motion Specification

A synthesized mechanism must provide the required motion or complete a certain task. In analytical synthesis methods, the required motion is usually specified by a set of precision positions (precision points) and the maximum number of precision points depends on the type of task and the mechanism type. On the other hand, for linkage synthesis based on optimization, the required motion can be specified with a large number of design points. One optimization goal becomes the best, or an adequate depending on application, satisfaction of the many design points.

Upper and lower limits at specific input increments from an unknown initial input displacement will be used in this thesis to approximate desired displacements as a function over a range of the input values. For a four-bar mechanism the crank angle is generally the input. With the member labelling of Figure 2.1 the four-bar input is angle θ_2. Modelling of a desired motion displacement component using upper and lower limits at specific $\Delta\theta_2$ increments is illustrated in Fig. 2.2 for a desired range of θ_2 values ($\theta_{2_0} \leq \theta_2 \leq \theta_{2_f}$). The solid line represents a desired displacement and solid triangles represent the acceptable range of resulting output displacement for the design requirements. In the case of the four-bar mechanism, this type of motion specification can be applied to any combination of the displacement terms X_P, Y_P, θ_3 and θ_4, where individual upper and lower constraint approximations would be utilized for each displacement component. This method of motion specification is discussed in further detail in Section 2.6.

2.2.3 Constraints

Methods of Handling Constraints

There are various means and combinations of means which can be used to impose constraints. There may be constraints regarding the size or proportion of link lengths, limits on acceptable transmission angles, assurance of mechanism assembly capability related constraints, etc. These constraints lead directly to inequalities which complicate the optimization process. However, there are some optimization methods that incorporate constraints directly in such a way as to effectively allow unconstrained optimization. One method of handling constraints is to include them in a penalty function with terms that increase when the constraints are violated. The penalty function is adjoined to the original objective function to produce a new objective

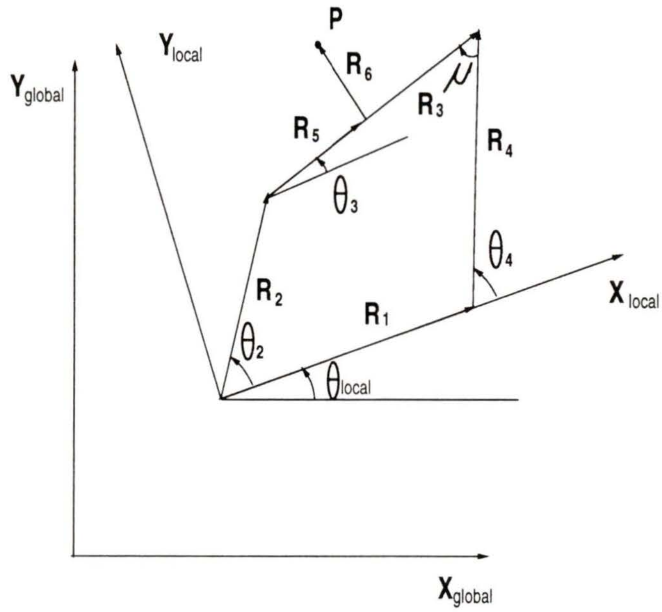

Figure 2.1: Four-bar Mechanism Vector Representation

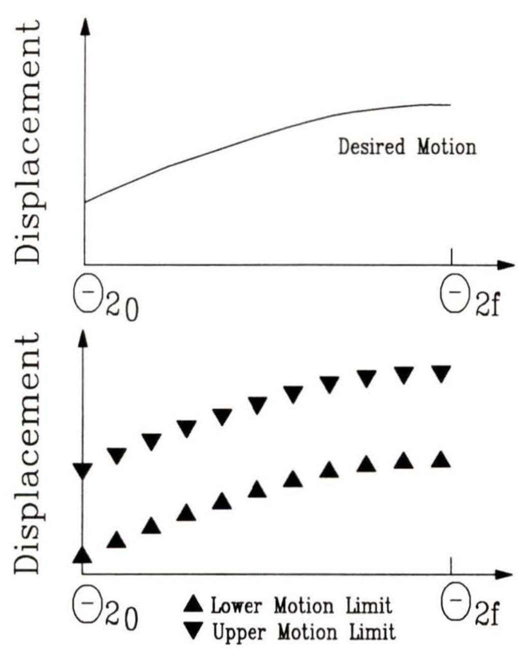

Figure 2.2: Motion Specification Using Upper and Lower Constraints

function. Another technique to handle the constraints is to develop algebraic transformations to allow a mapping of unconstrained variable values to constrained variables. The ability to develop such transforms is dependent on the characteristics of the constraint. Using a combination of transformation and penalty function techniques can facilitate the use of unconstrained minimization techniques.

Motion Constraints

Motion requirements will become constraints when other objectives are to be optimized. In this work, motion constraints will be ensured by using a sequentially unconstrained minimization technique (SUMT)(1989) [3]. When the constraints are approached, a penalty value is assigned to force the searching process away from the boundary and to keep it within the feasible area. A sequence of unconstrained searches where the penalty values are successively scaled is utilized to allow convergence to the actual objective function value. Use of SUMT requires initial feasible mechanisms. Section 2.6.2 presents a least pth technique for finding feasible mechanisms. Section 2.6.3 further discusses SUMT.

Maximum and Minimum Design Parameter Limits

Constraints on the maximum and minimum values for the design parameters often exist. In performing searches for optimization purposes, it is preferred to search over a set of unconstrained search variables. A transformation is developed to transform unconstrained search variables $\mathbf{X}$ into constrained search variables $\mathbf{r}$ in section 2.5.

Grashof-based Mechanisms

It is often desirable to have a mechanism that allows complete rotations of at least one link relative to an adjacent link. It is also often desirable to search within specific mechanism sub-types. In this work, transformations are developed based on Grashof-criterion considerations to allow unconstrained searching within specific mechanism sub-types. Details are contained in Chapter 3.

2.2.4 Design Objectives

The goal of mechanism synthesis is to synthesize a mechanism that satisfies certain motion requirements, and is optimal for desired mechanism characteristics objectives. Several objectives may be relevant in mechanism design. These objectives may be concerned with specific kinematic and dynamic related performance criteria (e.g., norm or maximum values of velocities and accelerations, minimum values of transmission angles, mechanical advantages, etc.) or may be concerned with implementation or application concerns (e.g., overall size of the device, relative link lengths, base pin placements, etc.). A composite objective of weighted objectives

$$f_{obj} = \sum w_i f_{obj_i}, \; i = 1, m \tag{2.2}$$

representative of desired performance can be minimized in a search for an optimal mechanism.

2.2.5 Basic Optimization Methodology

The terms of the objective function of eqn. (2.2) typically evolve during a mechanism synthesis process. That is, a specific objective may be the initial goal, but upon simulation and evaluation of the performance of the resulting mechanism, further objectives will typically be considered. Several such iterations will generally occur, often considering several mechanism types, before a suitable mechanism is synthesized.

As mentioned in Section 1.3, there are several search techniques available. Some of them include gradient-based techniques, Hessian-based methods, and direct-search-based methods. Gradient-based and Hessian methods require the computations of first derivatives and first & second derivatives, respectively. Gradient and Hessian methods can have the advantage of rapid convergence, but can also be very sensitive to the shape of objective function and will increase the algebraic and computational complexity of the optimization dramatically, in comparison to direct-search methods.

To facilitate the optimization of practical mechanisms, where many objectives may be considered and may be discontinuous in. nature, a direct-search method will be used in this research. While slower to converge than gradient-based methods, direct-search methods allow the consideration of arbitrary objective functions. Specifically, the flexible polyhedron search technique (Press, et al. 1989 [32]) is utilized.

For direct searching, the ability to search over an unconstrained domain of search variables is an asset. In this thesis, transformations of unconstrained search variables to constrained mechanism parameters are utilized to allow unconstrained searching. In addition, a least pth algorithm with specific tasks approximated by the specification of upper and lower limit constraints will be used to generate an initial feasible mechanism and sequentially unconstrained minimization techniques are used to ensure motion feasibility during mechanism optimization for the specified objectives.

2.3 Kinematic Modeling

For the optimization process, appropriate models allowing the determination of mechanism performance characteristics are required. Critical to kinematic synthesis is having complete models of displacement, velocity and acceleration characteristics as well as related indices such as transmission angles. In this work, explicit models are utilized . A four-bar linkage is considered in this section as an example of kinematic modelling.

2.3.1 Motion Analysis for a Four-bar Mechanism

A loop-closure of vectors (summing the vectors around the kinematic loop) representing the linkage members of the four-bar as illustrated in Fig. 2.1 yields the vector equation,

$$r_2 e^{i\theta_2} + r_3 e^{i\theta_3} - r_4 e^{i\theta_4} - r_1 = \mathbf{0}. \tag{2.3}$$

Breaking into real X and imaginary Y components gives:

$$\begin{aligned} r_2\cos(\theta_2) + r_3\cos(\theta_3) - r_4\cos(\theta_4) - r_1 &= 0 \\ r_2\sin(\theta_2) + r_3\sin(\theta_3) - r_4\sin(\theta_4) &= 0. \end{aligned} \tag{2.4}$$

Solving the above equations for unknowns θ_3 and θ_4 yields,

$$\begin{aligned} \theta_3 &= 2\ \arctan((-B \pm \sqrt{B^2 - 4\,AC})/(2\,A)) \\ \theta_4 &= 2\ \arctan((-B \pm \sqrt{B^2 - 4\,DE})/(2\,D)) \end{aligned} \tag{2.5}$$

where,

$$\begin{aligned} A &= (r_1/r_3)C_2 + C_2 + \text{Term}_1 - r_1/r_2 \\ B &= -2S_2 \end{aligned}$$

$$
\begin{aligned}
C &= (r_1/r_3)C_2 - C_2 + \text{Term}_1 + r_1/r_2 \\
D &= C_2 + \text{Term}_2 - r_1/r_2 - (r_1/r_4)C_2 \\
E &= r_1/r_2 + \text{Term}_2 - (1 + r_1/r_4)C_2
\end{aligned}
$$

with

$$C_2 = \cos(\theta_2) \text{ and } S_2 = \sin(\theta_2)$$

and

$$
\begin{aligned}
\text{Term}_1 &= (r_4^2 - r_1^2 - r_2^2 - r_3^2)/(2r_2r_3) \\
\text{Term}_2 &= (r_1^2 + r_2^2 - r_3^2 + r_4^2)/(2r_2r_4).
\end{aligned}
$$

In the equations (2.5), the positive sign corresponds to the cross link case, and the negative sign corresponds to the non-cross link case.

Taking the derivative of eqn. (2.4) with respect to time yields

$$
\begin{aligned}
-r_2\dot{\theta}_2 \sin(\theta_2) - r_3\dot{\theta}_3 \sin(\theta_3) + r_4\dot{\theta}_4 \sin(\theta_4) &= 0 \\
r_2\dot{\theta}_2 \cos(\theta_2) + r_3\dot{\theta}_3 \cos(\theta_3) - r_4\dot{\theta}_4 \cos(\theta_4) &= 0.
\end{aligned} \tag{2.6}
$$

Solving $\dot{\theta}_3$ and $\dot{\theta}_4$ yields the velocity solutions

$$
\begin{aligned}
\dot{\theta}_3 &= (r_2\omega_2 S_{24})/(r_3 S_{43}) \\
\dot{\theta}_4 &= (r_2\omega_2 S_{23})/(r_4 S_{43})
\end{aligned} \tag{2.7}
$$

where $S_{ij} = \sin(\theta_i - \theta_j)$, and $\omega_2 = \dot{\theta}_2$ is the input rotational velocity. Taking the second derivative of eqns. (2.4) with respect to time yields

$$
\begin{aligned}
-r_2\dot{\theta}_2^{\,2} \cos(\theta_2) - r_3\dot{\theta}_3^{\,2} \cos(\theta_3) - r_3\ddot{\theta}_3 \sin(\theta_3) + r_4\dot{\theta}_4^{\,2} \sin(\theta_4) + r_4\ddot{\theta}_4 \sin(\theta_4) &= 0 \\
-r_2\dot{\theta}_2^{\,2} \sin(\theta_2) - r_3\dot{\theta}_3^{\,2} \sin(\theta_3) + r_3\ddot{\theta}_3 \cos(\theta_3) + r_4\dot{\theta}_4^{\,2} sin(\theta_4) - r_4\ddot{\theta}_4 \cos(\theta_4) &= 0
\end{aligned} \tag{2.8}
$$

Solving for $\ddot{\theta}_3$ and $\ddot{\theta}_4$ yields the acceleration solutions

$$\ddot{\theta}_3 = (r_4\dot{\theta}_4^{\,2} - r_2\omega_2^2 C_{24} - r_3\dot{\theta}_3^{\,2} C_{34})/(r_3 S_{34})$$
$$\ddot{\theta}_4 = (r_3\dot{\theta}_3^{\,2} + r_2\omega_2^2 C_{23} - r_4\dot{\theta}_4^{\,2} C_{34})/(r_4 S_{43}) \qquad (2.9)$$

where $C_{ij} = \cos(\theta_i - \theta_j)$.

For the kinematic equations describing the motion of the coupler point P, consider an arbitrary orientation of a four-bar mechanism in a plane as shown in Fig. 2.1. The X and Y displacements of the coupler point with respect to the local coordinate system are given by:

$$^{l}X_P = r_2\cos(\theta_2) + r_5\cos(\theta_3) - r_6\sin(\theta_3)$$
$$^{l}Y_P = r_2\sin(\theta_2) + r_5\sin(\theta_3) + r_6\cos(\theta_3) \qquad (2.10)$$

Differentiating the eqn. (2.10) with respect to time yields the velocity and acceleration components:

$$^{l}\dot{X}_P = -r_2\omega_2\sin(\theta_2) - r_5\dot{\theta}_3\sin(\theta_3) - r_6\dot{\theta}_3\cos(\theta_3)$$
$$^{l}\dot{Y}_P = r_2\omega_2\cos(\theta_2) + r_5\dot{\theta}_3\cos(\theta_3) - r_6\dot{\theta}_3\sin(\theta_3)$$
$$^{l}\ddot{X}_P = -r_2\omega_2\cos(\theta_2) - r_5(\ddot{\theta}_3\sin(\theta_3) + \dot{\theta}_3^{\,2}\cos(\theta_3)) - r_6(\ddot{\theta}_3\cos(\theta_3)) - \dot{\theta}_3^{\,2}\sin(\theta_3))$$
$$^{l}\ddot{Y}_P = -r_2\omega_2\sin(\theta_2) + r_5(\ddot{\theta}_3\cos(\theta_3) - \dot{\theta}_3^{\,2}\sin(\theta_3)) - r_6(\ddot{\theta}_3\sin(\theta_3)) + \dot{\theta}_3^{\,2}\cos(\theta_3)).$$

With respect to the global coordinates, the equations for the kinematics associated with the coupler point become:

$$^{g}X_P = {}^{l}X_P C_{local} - {}^{l}Y_P S_{local} + X$$
$$^{g}Y_P = {}^{l}X_P S_{local} + {}^{l}Y_P C_{local} + Y$$
$$^{g}\dot{X}_P = {}^{l}\dot{X}_P C_{local} - {}^{l}\dot{Y}_P S_{local}$$
$$^{g}\dot{Y}_P = {}^{l}\dot{X}_P S_{local} + {}^{l}\dot{Y}_P C_{local} \qquad (2.11)$$

$$
\begin{aligned}
{}^{g}\ddot{X}_P &= {}^{l}\ddot{X}_P C_{local} - {}^{l}\ddot{Y}_P S_{local} \\
{}^{g}\ddot{Y}_P &= {}^{l}\ddot{X}_P S_{local} + {}^{l}\ddot{Y}_P C_{local}
\end{aligned}
$$

where θ_{local} is the angular orientation of the local coordinate system; X is the x displacement of the local coordinate system; Y is the y displacement of the local coordinate system and $S_{local} = \sin(\theta_{local})$ and $C_{local} = \cos(\theta_{local})$.

The above equations describe the kinematics of a four-bar mechanism by forming a mathematical model of the linkage. Other types of mechanisms can be described with similar mathematical models. Models are developed for six-bar and geared five-bar mechanisms in Appendix A.

2.3.2 Transmission Angle Analysis for a Four-bar Mechanism

In designing mechanisms, it is often desirable to ensure the quality of force transmission. The minimum value of transmission angle can be a measure of mechanism force transmission adequacy. The transmission angle, μ, of the four-bar linkage of Fig. 2.3, is defined as the smaller (acute) angle between the direction of the velocity-difference vector of the coupler link and the direction of the absolute velocity of the output link, both taken at the point of connection. The best transmission occurs when $\mu = 90°$. Minimum transmission angles ($\mu = 30°$ is often the recommended minimum) should be assured for mechanisms driving substantial loads. The value of $\sin\mu$ is a direct indicator of what percentage of the force applied by the coupling link r_3 actually performs useful work on the load applied at the output link r_4. Clearly, the higher the absolute value of $\sin\mu$, the better the transmission properties of the linkage. Hence, the absolute value of $\cos\mu$ can be used as a suitable means to penalize a poor performance of a linkage from the viewpoint of force transmission. An

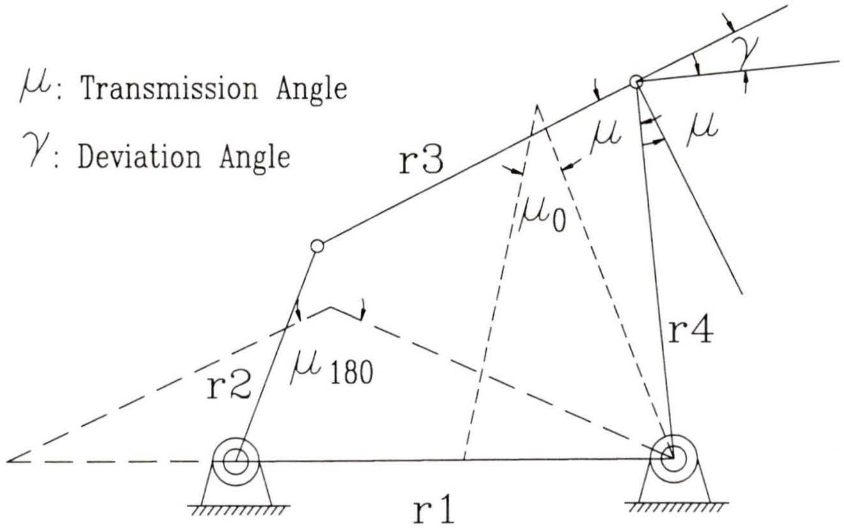

Figure 2.3: Transmission Angles of Four-bar Mechanisms

expression for $\cos\mu$ is

$$\cos\mu = (r_3^2 + r_4^2 - r_1^2 - r_2^2 + 2r_1r_2cos(\theta_2))/(2r_3r_4) \tag{2.12}$$

Note that the transmission angle is a function of the input displacement. The transmission angle can be at its minimum value when $\theta_2 = 0°$ or when $\theta_2 = 180°$, i.e., at μ_0 and μ_{180} as illustrated in Fig. 2.3. The expressions of μ_0 and μ_{180} become:

$$\cos\mu_0 = (r_3^2 + r_4^2 - (r_1 - r_2)^2)/(2r_3r_4) \tag{2.13}$$
$$\cos\mu_{180} = (r_3^2 + r_4^2 - (r_1 + r_2)^2)/(2r_3r_4)$$

If the values generated by eqn. (2.13) exceed 90°, the transmission angle is 180° minus the generated value. An objective function to maximize the transmission angle can be formed. That is,

$$f_{obj} = 1/\min\{\mu_{min1}, \mu_{min2}\} \tag{2.14}$$

where $\mu_{min1} = \mu_0$, $\mu_{min2} = \mu_{180}$.

2.3.3 Grashof Criteria and Sub-types of Four-bars

Grashof mobility criteria for RRRR planar linkages establish the conditions on the relative lengths of the links for the existence of Grashof four-bar mechanisms. Grashof's law states that the sum of the shortest and longest links of a planar four-bar linkage cannot be greater than the sum of the remaining two links if there is to be continuous relative rotation between two links. This criterion would be considered if a linkage is driven by a continuously rotating motor. By considering geometry, the criteria for a four-bar to be a Grashof mechanism can be seen to be [39]:

$$r_{short} + r_{long} \leq r_a + r_b \tag{2.15}$$

where r_{short} and r_{long} are respectively the shortest and the longest lengths among the frame, crank, coupler and follower, and r_a and r_b are the other two lengths.

The Grashof mechanisms may be further classified into sub-types capable of complete rotation of a link(s) according to the position of the shortest link. The possible conditions are summarized in Table 2.1.

$r_{short} + r_{long}$ $(<, =)$ $r_a + r_b$	Shortest link	Type
$<$	Frame	Drag-links(Double-Crank)
$<$	Crank	Crank-Rocker
$<$	Coupler	Rocker-Rocker
$=$	Any	Change-point

Table 2.1: Sub-types of Grashof Four-bar Mechanisms

For non-Grashof four-bars,

$$r_{short} + r_{long} > r_a + r_b \tag{2.16}$$

All non-Grashof four-bars are triple-rockers.

Grashof-type criteria are utilized in developing transformations to ensure searching over specific mechanism sub-types in Chapter 3.

2.4 Mechanism Search Variables

2.4.1 Review of Potential Mechanism Tasks

The design variables for the synthesis of a mechanism are the parameters associated with the input-output relationship of the mechanism. As such, the design variables are dependent on the type of mechanism and the task under consideration. Mechanism tasks as mentioned in Chapter 1 can be classified as: i) function generation; ii) path generation; iii) motion generation (rigid body guidance). Tasks may involve displacement relationships and may also deal with the achieval of higher-order kinematic and/or force relationships.

2.4.2 Search Variables for Absolute Task Specification

Consider the four-bar mechanism vector-representation as illustrated in Fig. 2.1. Function generation would desire a relationship between changes in the angular displacement of the output link $\mathbf{R}_4$ (follower) as a function of changes in the input link $\mathbf{R}_2$ (crank) angle. The follower angle $\theta_4 = f(\theta_2, r_2^{norm}, r_3^{norm}, r_4^{norm})$ where $r_i^{norm} = r_i/r_1$ is the length of ith link normalized with respect to $r_1 = ||\mathbf{R_1}||$ (the length of the base link). This normalization is done since scaling of all of the link lengths does not affect the angular output. The location, X and Y, and orientation, θ_{local}, in space of the four-bar also are not variables for an angular function generation since the task is not related to the location of a point and since the angles are always measured with respect to the direction of the base link. The base link can be located and oriented arbitrarily and the task will still be satisfied. Therefore, the design variables for synthesis of a four-bar function generator total four and consist of the initial crank angle (θ_{2_0}) and the normalized lengths of the crank (r_2^{norm}), the coupler (r_3^{norm}) and the follower (r_4^{norm}).

Path generation involves achieving a relationship between the locations { X_P, Y_P } of a coupler point P and the input angle θ_2. Noting that ${}^g\{X_P, Y_P\} = f(\theta_2, r_i, X_{local}, Y_{local}, \theta_{local})$ where θ_2 is the crank angle, $r_i, i = 1,6$, are the link lengths (r_5 and r_6 defining the specific point on the coupler), and X_{local}, Y_{local}, and θ_{local} are coordinates defining the location and orientation of the mechanism with respect to a global frame F_{global}. Hence, synthesis for path generation can be considered to have up to ten design variables for absolute task specification.

Motion generation involves achieving relationships for both the coupler's angle θ_3 and the location ${}^g\{X_P, Y_P\}$ of a coupler point P with respect to the input angle θ_2. Since ${}^g\{X_P, Y_P\} = f(\theta_2, r_i, X_{local}, Y_{local}, \theta_{local})$ and ${}^g\theta_3 = g(\theta_2, r_i^{norm}, \theta_{local})$ the design parameters for rigid body guidance are the same as for path generation. Similarly, the parameters for other mechanism types can be easily found.

2.4.3 Search Variables for Relative Task Specification

The search variables for path and motion generation were specified above with respect to the global frame F_{global}. However, the task can be specified in terms of required changes in the location of the point P, noting that the mechanism base can be relocated with respect to F_{global}. This results in a reduction to eight searching parameters for many path and motion generation. This is, ${}^{local_g}\{X_P, Y_P\} = f(\theta_2, r_i, \theta_{local})$, where $i = 1,6$, and F_{local_g} refers to a F_{local} located, globally-oriented reference. However, X_{local} and Y_{local} can not be reduced if the required task has to be specified with the global frame of reference.

2.5 Constraints on Design Parameters

Constraints on the maximum and minimum values for the design parameters often exist. The transform of eqn. (2.17) allows a mapping from an unconstrained variable, x_i, to a parameter, r_i, constrained to lie between $r_{lower_i} \leq r_i \leq r_{upper_i}$ where r_{lower_i} and r_{upper_i} are lower and upper limits of the ith design parameter, respectively.

$$r_i = r_{lower_i} + (r_{upper_i} - r_{lower_i})/(1 + e^{-x_i}) \tag{2.17}$$

The mapping provided by the transform of eqn. (2.17) is illustrated in Fig. 2.4.

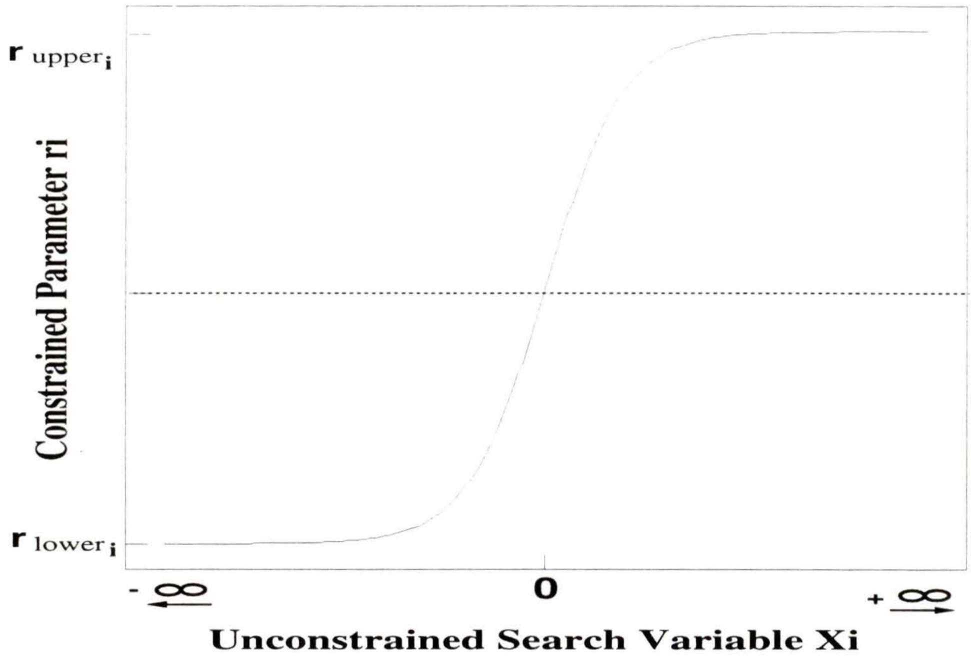

Figure 2.4: The Unconstrained to Constrained Mapping of eqn. (2.17)

While accommodating upper and lower limits on link lengths, the transformation of eqn. (2.17) will not ensure that the search is confined to a specific mechanism sub-type. As was previously mentioned in the section 2.3.3 on the analysis of four-bar mechanisms, only certain relative link lengths allow mechanisms to be capable of complete crank rotation.

A method is developed in the work of this thesis to ensure searching within specific mechanism sub-types. The basis of this new method is the redefinition of the lower and upper link length limits $\mathbf{r}_{lower}$ and $\mathbf{r}_{upper}$, concerned with the transformation of unconstrained search variables, into new limits $\mathbf{r}^*_{lower}$ and $\mathbf{r}^*_{upper}$. That is, $\mathbf{r}^*_{lower} = f(\mathbf{r}_{lower}, \mathbf{X})$ and $\mathbf{r}^*_{upper} = g(\mathbf{r}_{upper}, \mathbf{X})$ are such that the redefined limits ensure that the unconstrained to constrained link length mapping of eqn. (2.17) results in a specific mechanism sub-type. The redefinition of limits is based on consideration of Grashof-type criteria. The method and its application to specific mechanism types is presented in detail in Chapter 3.

2.6 Specifying Task Motion Requirements and Ensuring its Feasibility

2.6.1 Task Specification

To be feasible the synthesized mechanism must provide the required motion. For some tasks close adherence to a specific desired output may be desired. In these cases, a motion objective can be appended to the objective function, i.e.,

$$f_{obj_{append}} = f_{obj} + w_{motion} \left(\sum_{j,k} w_{jk} (| f_{error_{jk}} |)^p \right)^{1/p} \tag{2.18}$$

where $f_{error_{jk}}$ is the displacement error of the jth displacement component at the kth evaluation point, w_{jk} and w_{motion} are weightings on the individual error components and the total of the weighted errors, respectively, and the term p allows a p-norm weighting of the motion errors (e.g., $p = 2$ yields a Euclidean (2) norm, while a large p emphasizes the maximum error). Selection of appropriate weightings allows desired emphasis on motion errors.

In many applications, strict adherence to a particular output is not a primary goal. That is, output within a certain range will accomplish the task, and the goal is to ensure that the mechanism output is within the feasible motion range and that the best mechanism within the feasible range is found. The optimization process can then proceed by first finding an initial feasible mechanism (e.g., by a least pth search as discussed in Section 2.6.2) and then constraining the mechanism to the feasible region while optimizing for the objective (e.g., by SUMT as discussed in Section 2.6.3).

2.6.2 Finding an Initial Feasible Mechanism

A feasible mechanism can be found through the optimization of a least pth objective based on motion-constraint violation (or degree of satisfaction, if a feasible mechanism furthest from the constraints is desired), in the following fashion.

Let

$$\begin{aligned} e_{upper_{jk}} &= w_{upper_{jk}}(d_{jk} - c_{upper_{jk}}) \\ e_{lower_{jk}} &= w_{lower_{jk}}(c_{lower_{jk}} - d_{jk}) \end{aligned} \tag{2.19}$$

for all j and k and

$$\begin{aligned} \mathbf{k}_{upper} &= \{jk, e_{upper_{jk}} > 0\} \\ \mathbf{k}_{lower} &= \{jk, e_{lower_{jk}} > 0\} \end{aligned} \tag{2.20}$$

or

$$\mathbf{k}_{upper_{jk}} = \mathbf{k}_{lower_{jk}} = \{jk, \text{for all } jk \text{ if } e_{max} \leq 0\} \tag{2.21}$$

with

$$e_{max} = \max_{jk}(e_{upper_{jk}}, e_{lower_{jk}}). \tag{2.22}$$

In equations (2.19) - (2.22), $e_{upper_{jk}}$ and $e_{lower_{jk}}$ represent the error (amount of satisfaction) of the actual displacement d_{jk} of the jth displacement component value at the kth task constraints $c_{upper_{jk}}$ and $c_{lower_{jk}}$; $w_{upper_{jk}}$ and $w_{lower_{jk}}$ represent weighting factors for the particular upper and lower constraints. If $e_{max} > 0$ all of the task displacement constraints are not satisfied. If $e_{max} \leq 0$ then all of the motion constraints are satisfied. An objective representing the degree of task constraint violation (satisfaction) can be written as

$$f_{task} = \left(\sum_{ij \in \mathbf{k}_{upper}} (e_{upper_{jk}}/e_{max})^q + \sum_{ij \in \mathbf{k}_{lower}} (e_{lower_{jk}}/e_{max})^q \right)^{1/q} \tag{2.23}$$

where $q = (e_{max}/|e_{max}|)p$.

Minimization of the objective of eqn. (2.23) results in the search forcing an initially infeasible (when $e_{max} > 0$) mechanism into the feasible region and then continuing to maximize the distance of the feasible (when $e_{max} < 0$) mechanism's output from the constraints. The specification of different p values allows a different emphasis to be placed upon the violation values. For example, a large p value will cause the f_{task} value to emphasize the maximum violation (or least satisfaction) of the constraints. A p value of two will cause a 2-norm (Euclidean norm) of the errors (or degree of satisfaction) to be considered. The change in f_{task} value is utilized as the exit criteria for the least pth search.

2.6.3 Ensuring a Feasible Mechanism During Mechanism Optimization

Once a feasible mechanism is found it is desired to find the optimal feasible mechanism for the design objectives. Sequential unconstrained minimization techniques (SUMT) can be utilized to facilitate confining the mechanism to the feasible region. In SUMT an appended objective function is considered, i.e.,

$$f_{SUMT} = f_{obj} + w_{SUMT} f_{task} \tag{2.24}$$

where f_{obj} is the design objective, and f_{task} is the constraint function developed in equations (2.19) - (2.23) and w_{SUMT} determines the weight on the constraint function. In the SUMT the value of w_{SUMT} is decreased (generally by a factor of 5 to 10) after each convergence and the search is restarted. As the minimization continues, the value of f_{SUMT} converges within a tolerance to the value of f_{obj}, at which point the search is terminated. Properly applied SUMT-based optimization will confine the search to mechanisms satisfying the task motion constraints and will allow convergence to the optimal mechanisms for the design objectives.

2.7 A Flow Chart for the Developed Synthesis Approach

See Figure 2.5.

2.8 Example Optimizations

2.8.1 A Rigid-Body Guidance Problem

A crank-rocker four-bar mechanism is desired to guide a payload from a one conveyor to another as shown schematically in Fig. 2.6. To achieve the task, a lift (Δy) and a translation (Δx) of approximately 100 cm and a change in angle ($\Delta\theta_3$) of 90° are required. Constraints on the lengths of the links of $\mathbf{r}_{upper} = \{175, 100, 160, 170, 230, 100\}^T$

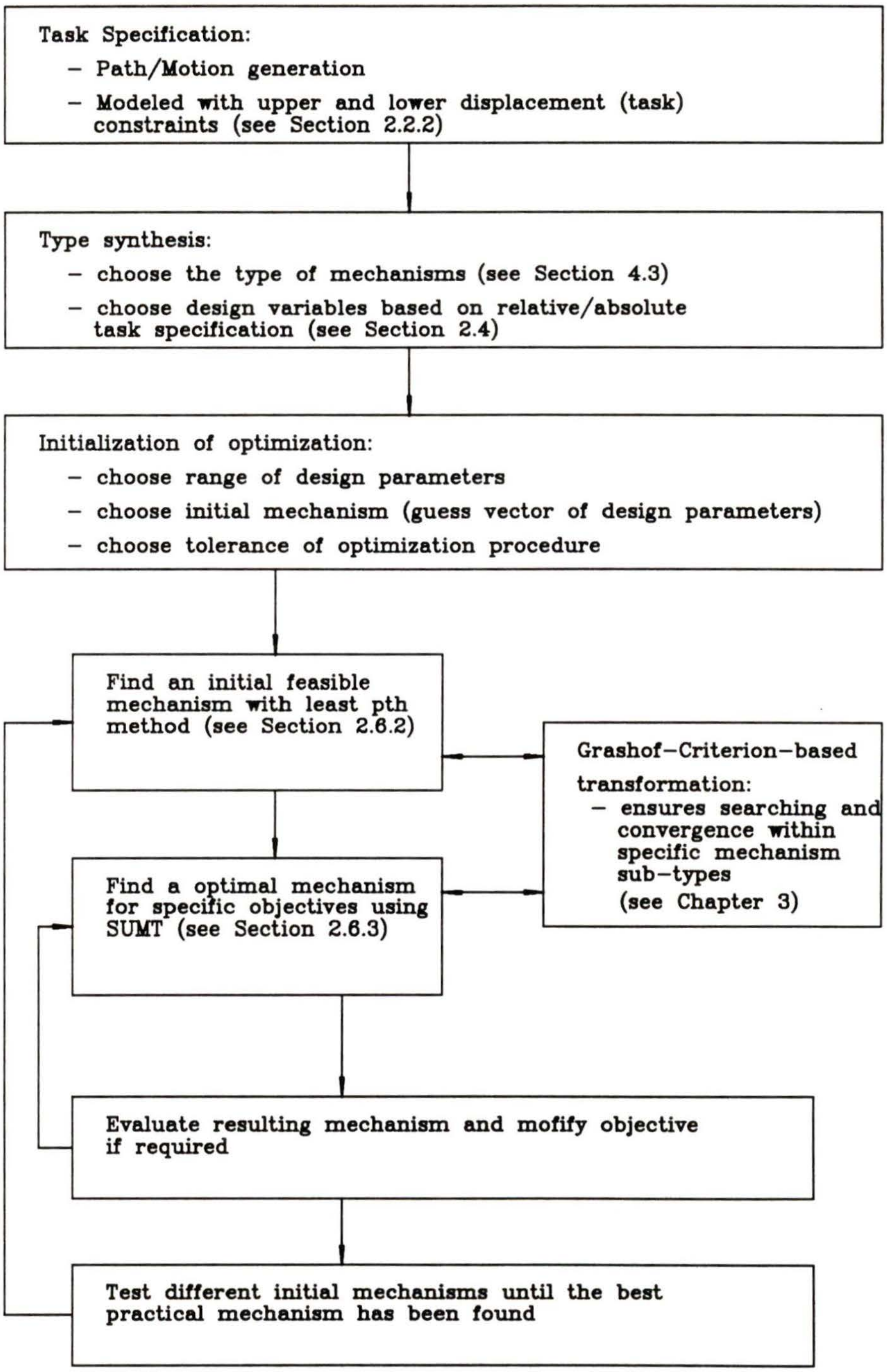

Figure 2.5: A Flow Chart for the Developed Synthesis Approach

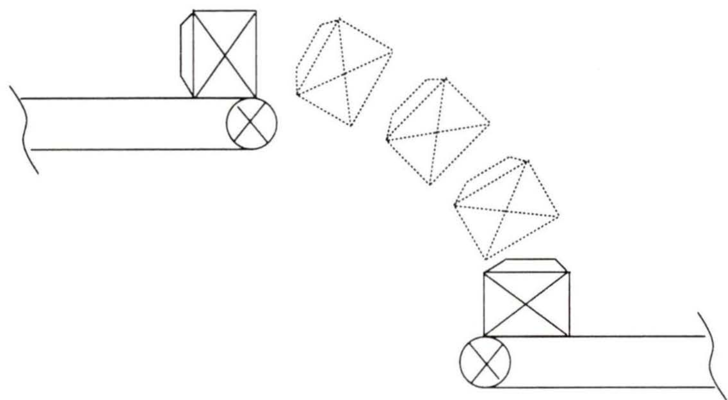

Figure 2.6: Payload Transfer Task (Rigid-Body Guidance)

cm and $\mathbf{r}_{lower} = \{25, 10, 25, 25, 5, -100\}^T$ cm exist. The initial value of the displacement of the crank (θ_{2_0}) is unconstrained (0-360°) and the values of θ_{local} are constrained to $\pm 90°$.

The task was modeled with upper and lower displacement component (task) constraints as illustrated in Figs. 2.7 to 2.9. An initial mechanism $\mathbf{r}_{initial} = \{80, 28, 55, 76, 26, 75\}^T$ cm with $\theta_{2_0} = 25°$ and $\theta_{local} = -25°$ resulted in the motion indicated by the solid lines on the graphs of Figs. 2.7 to 2.9. Note that the initial mechanism is infeasible in terms of the required task.

A least pth search (as discussed in Section 2.6.2) combined with the sub-type specific transforms (as discussed in Chapter 3) rendered a feasible crank-rocker with $\mathbf{r}_{feasible} = \{33.96, 18.27, 25.60, 28.17, 8.97, 95.37\}^T$ cm with $\theta_{2_0} = 23.4°$ and $\theta_{local} = -0.4°$ The displacement output for this feasible mechanism is indicated by the dashed lines in Figs. 2.7 to 2.9. Least pth searches for different initial mechanisms were performed and their results are included in Table 2.2 where n-cross link refers to a

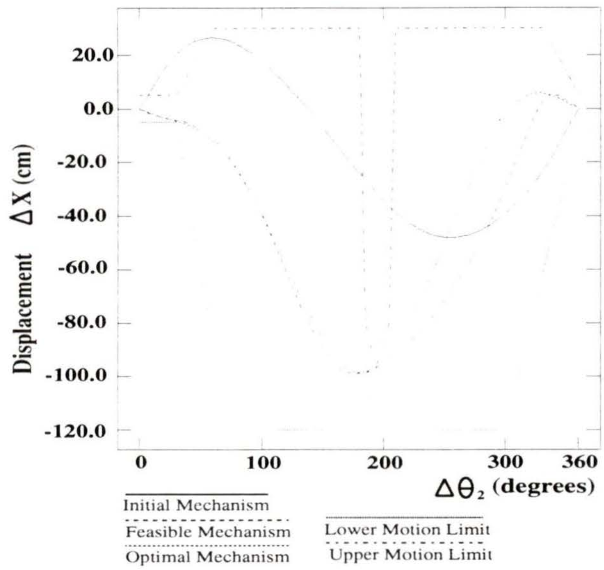

Figure 2.7: Task Specification and Resulting Mechanism Motions - ΔX

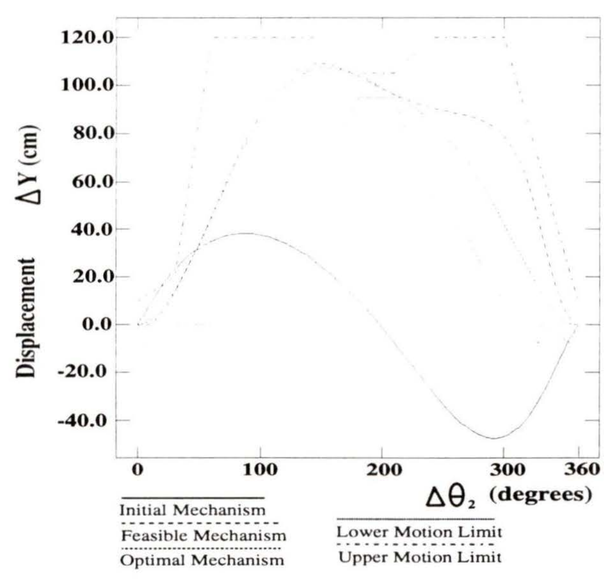

Figure 2.8: Task Specification and Resulting Mechanism Motions - ΔY

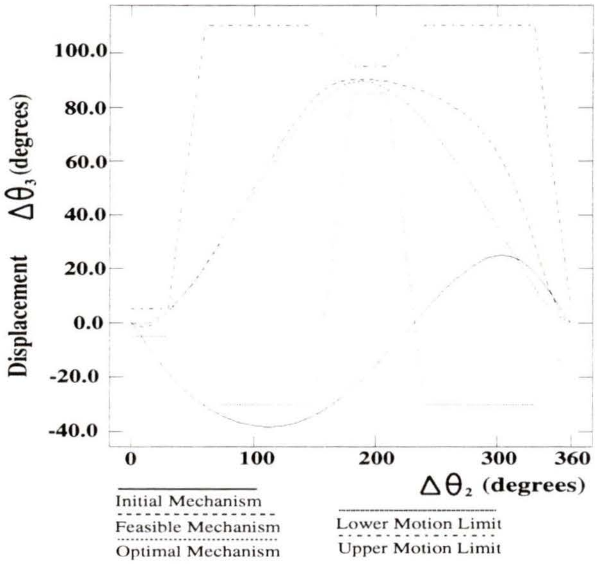

Figure 2.9: Task Specification and Resulting Mechanism Motions - $\Delta\theta_3$

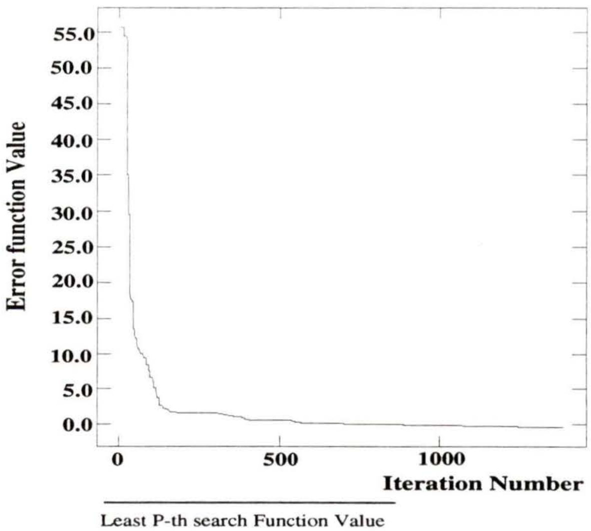

Figure 2.10: Example Least pth Search Results

ini. mechanism $\mathbf{r}_{initial}$	$f_{initial}$ value	feasible mechanism $\mathbf{r}_{feasible}$	No. of iteration	f_{final} value
50,15,39,45, 28,23,20,0	144.315	32.65,19.28,26.89,25.42, 11.19,94.73,27.94,-4.77	965	-0.226
50,15,39,45, 28,23,20,0	411.927 n-cross link	25.03,16.93,25.03,25.00, 98.92,-73.19,119.45,18.80	516	-0.191
50,18,35,46, 26,65,25,-20	380.338	41.89,22.02,29.83,35.72, 17.50,92.36,22.05,4.22	1280	-0.035
50,15,45,46, 26,65,10,-20	175.778	30.42,17.92,25.00,25.00, 5.10,95.30,25.72,-3.04	1521	-0.361
50,15,45,46, 26,65,10,-20	390.607 n-cross link	25.13,16.94,25.13,25.00, 85.40,-85.52,120.88,29.40	1475	-0.274
80,18,45,66, 36,85,5,-5	34.738	116.40,20.25,27.51,113.3, 19.97,93.16,3.02,25.35	1004	-0.0314
80,28,55,76, 26,75,25,-25	55.61	33.96,18.27,25.60,28.17, 8.97,95.37,23.43,-0.43	1386	-0.307
80,28,55,76, 26,75,25,-25	383.709 n-cross link	32.67,17.09,25.04,32.67, 73.46,-89.60,140.36,30.02	2173	-0.172
40,20,30,35, 15,80,20,-30	133.408	32.54,19.21,26.76,26.68, 7.90,95.13,26.34,-2.74	1118	-0.296
40,20,30,35, 15,80,20,30	41.2464	30.88,17.94,25,25.33.00, 6.78,95.37,25.56,-2.18	2233	-0.349

Table 2.2: Least pth Search Results for the Lifting Mechanism

non-cross four-bar configuration. The function value results for one least pth search are shown on Fig. 2.10.

For the task of payload transfer the required input torque will be a concern. The input torque and mechanism pin forces are dependent upon the inertia and acceleration of the mechanism members and the externally applied forces. In this case the exterior force applied to the four-bar mechanism is caused by the resistance of the payload to motion and gravity. The resistance to motion is in part dependent upon the acceleration of the payload and therefore upon the norm of acceleration ${}^g\ddot{X}_P$ and ${}^g\ddot{Y}_P$ for $\Delta\theta_2 = 0 - 200°$ (the lift portion of the motion). Simulation of the least pth rendered mechanism found it to have a maximum acceleration norm 194.78

m/s^2 at the payload point P during the transfer portion ($\Delta\theta_2 = 0 - 200°$) as shown by the solid line on Fig. 2.11. This was deemed higher than desired. An objective of minimizing the maximum norm of the acceleration of P, i.e.,

$$f_{objj_1} = \max_{0° \leq \Delta\theta_2 \leq 200°} (({}^g\ddot{x}_P)^2 + ({}^g\ddot{y}_P)^2)^{1/2} \tag{2.25}$$

over the transfer motion portion of the task was considered. Figures 2.12 and 2.13 show the simulation results of ${}^g\ddot{x}_P$ and ${}^g\ddot{y}_P$ with the solid lines.

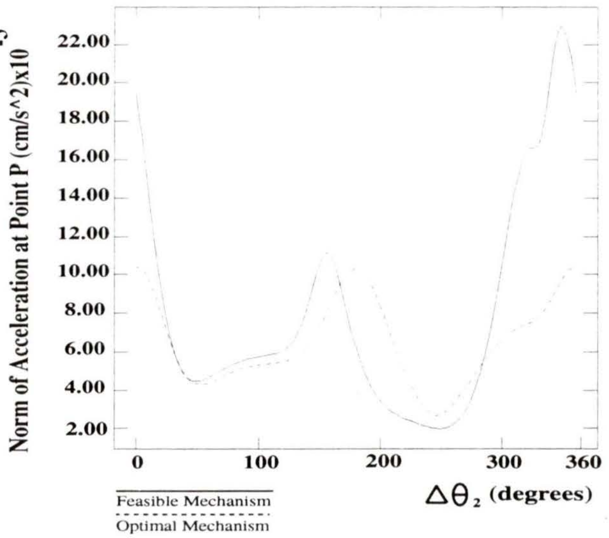

Figure 2.11: Norm of the Acceleration - Feasible and Optimal Mechanisms

The sequential unconstrained minimization technique discussed in Section 2.6.3, in combination with the sub-type specific transformation method, was applied, resolving the mechanism as shown in the Obj. 1 row in Table 2.3. The results from the search process are shown in Figure 2.14. From Table 2.3, the optimized mechanism has shown a maximum acceleration norm of 104 m/s^2 which is approximately 52 percent of the maximum norm of acceleration value for the starting feasible mechanism.

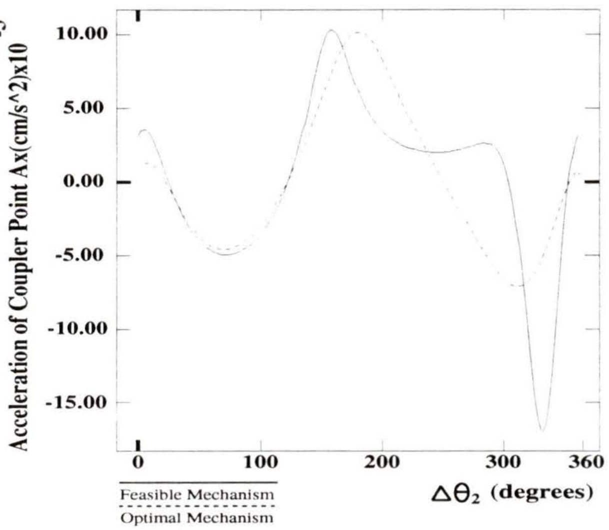

Figure 2.12: Acceleration in x Direction

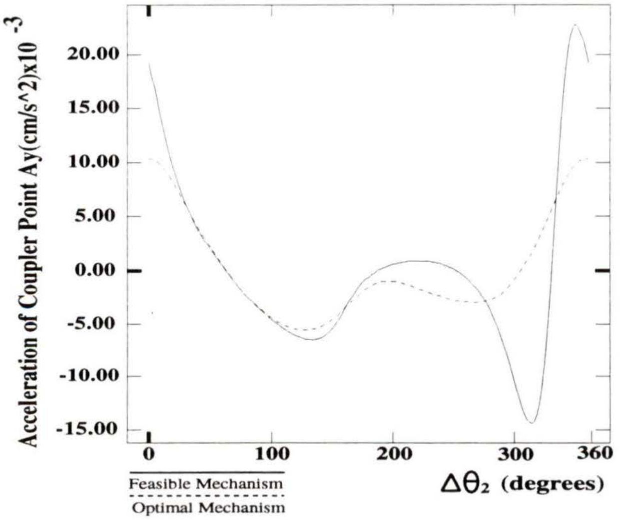

Figure 2.13: Acceleration in y Direction

Objective	f_{obj} value before opt.	f_{obj} value after opt.	No. of iterations	Optimized mechanism (cm)
Obj. 1	194.8 m/s^2	104 m/s^2	859	141.5,17.7,25.1,140.4 5.0,100.0,0.0,26.7
Obj. 2	27.4 degrees	41.0 degrees	923	90.6,17.6,25.0,90.1 11.8,100.0,3.2,28.2
Obj. 3	1960 Nm	1750.6 Nm	985	80.1,18.1,25.6,78.0 5.7,97.7,4.8,19.5
Comb. Obj.	4980	3845	966	92.0,18.0,25.3,90.2 7.3,96.5,2.7,21.9
Obj. 2	27.4(2047) degrees	43.3(1294) degrees		
Obj. 3	1960(2934) Nm	1703.1(2550) Nm		

Table 2.3: Optimization Results for the Lifting Mechanism

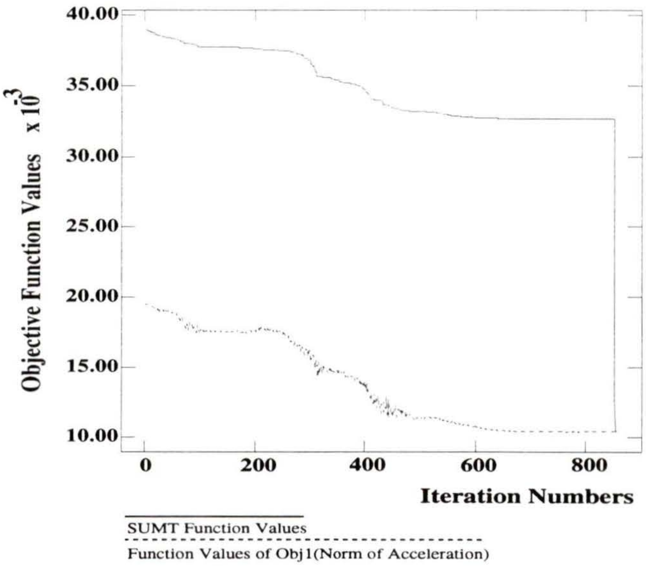

Figure 2.14: Optimization Results ($f_{obj_1} = \max_{0^\circ \leq \Delta\theta_2 \leq 200^\circ}((^g\ddot{x}_P)^2 + (^g\ddot{y}_P)^2)^{1/2}$)

The acceleration norm for the optimal mechanism is shown by the dashed line on Fig. 2.11. The corresponding results for ${}^g\ddot{y}_P$ and ${}^g\ddot{y}_P$ of the optimal mechanism are shown by the dotted lines on the Figures 2.12 and 2.13.

Since the designed mechanism will be used to lift a heavy payload, the transmission angle will be of concern. Simulation of the least pth rendered mechanism found it to have a minimum transmission angle $\mu_{min} = 27.4°$ as shown by solid line on Fig. 2.15. Therefore, the second objective to be considered was maximizing the minimum of the transmission angle over the transfer-motion portion of the task, in other words, minimizing the inverse of the minimum of the transmission angle, i.e.,

$$f_{obj_2} = \min_{0° \leq \Delta\theta_2 \leq 200°} [1/\min\{\mu_{min1}, \mu_{min2}\}] \tag{2.26}$$

where $\mu_{min1} = \mu_0$ and $\mu_{min2} = \mu_{180}$ are calculated from equations (2.14) (see Section 2.3.2).

The SUMT was applied again resolving the mechanism shown in the Obj. 2 row in Table 2.3 and the searching process shown in Figure 2.16. From Table 2.3, the optimized mechanism for the objective of maximizing the minimum transmission angle has $\mu_{min} = 41.0°$, corresponding to a 50 percent increase in the minimum transmission angle over that of the starting feasible mechanism. The transmission angle for the optimal mechanism is indicated by the dashed line on Fig. 2.15.

The required input torque should also be considered from the point of view of dynamic analysis of the system. An inertial model is shown in Fig. 2.17. The links are modelled as slender rods having masses of 5 kg/m and the package is modelled as a point mass of 50 kg. Simulation of the least pth rendered mechanism gave the maximum torque of 1960 Nm over the payload transferring process as shown by the dashed line on Fig. 2.18. The effect of the payload on the required input torque is seen substantial ($T_{max} = 1960.4$ Nm at $\theta_2 = 24°$). When unloaded, as shown by the

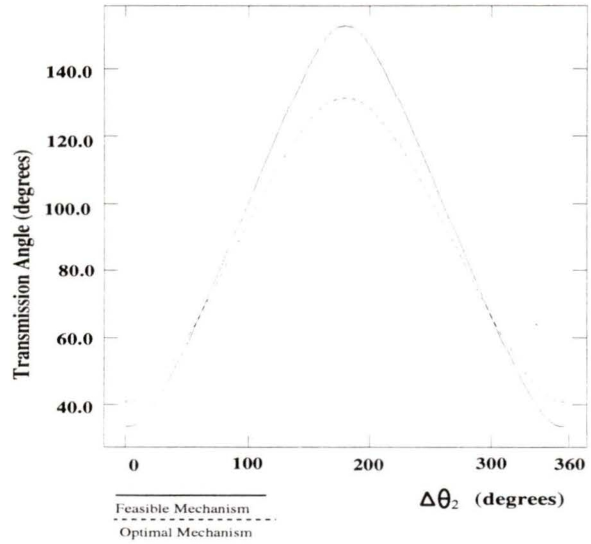

Figure 2.15: Transmission Angle - Feasible and Optimal Mechanisms

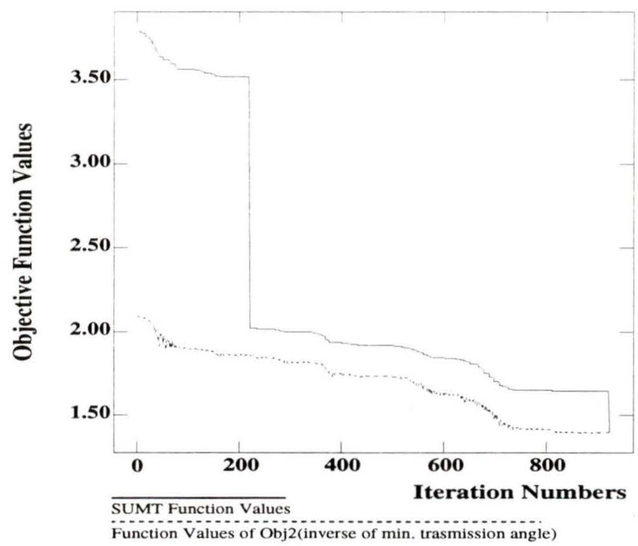

Figure 2.16: Optimization Results ($f_{obj_2} = \min_{0° \leq \Delta\theta_2 \leq 200°} [1/\min\{\mu_{min1}, \mu_{min2}\}]$)

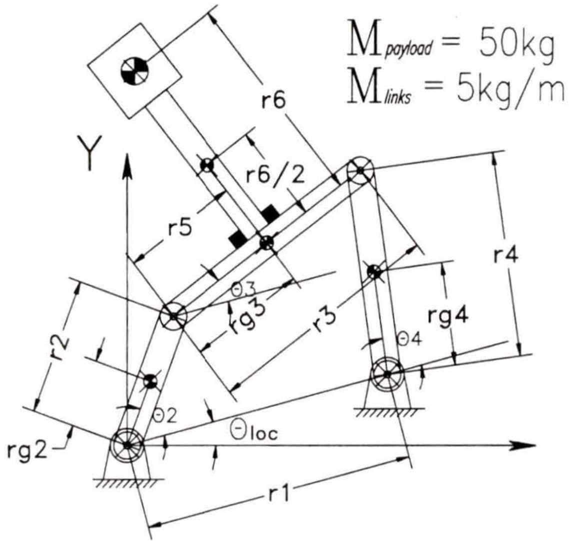

Figure 2.17: Inertial Model During Package Transfer

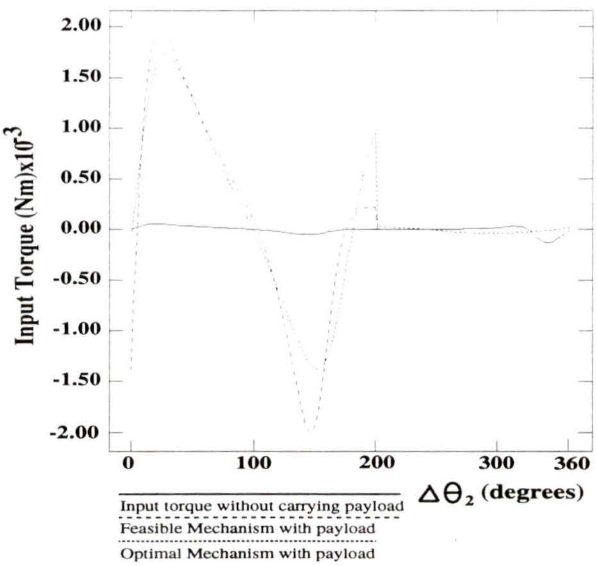

Figure 2.18: Required Input Torque - Feasible and Optimal Mechanisms

solid line on Fig. 2.18, the required input torque is very small ($T_{max} = 16.7$ Nm at $\theta_2 = 326°$). Thus, the third objective of minimizing the maximum of the required input torque T_{input} during the lift, i.e.,

$$f_{obj_3} = \max_{0° \leq \Delta\theta_2 \leq 200°} T_{input} \tag{2.27}$$

was considered.

The SUMT was applied, resolving the mechanism shown in the Obj. 3 row in Table 2.3 and the searching process shown in Figure 2.19. From Table 2.3, the optimized mechanism for the required input torque has a maximum torque of 1751 Nm which is about 89 percent of that of the feasible mechanism. The required input torque for this optimal mechanism is shown by the dotted line on Fig. 2.18.

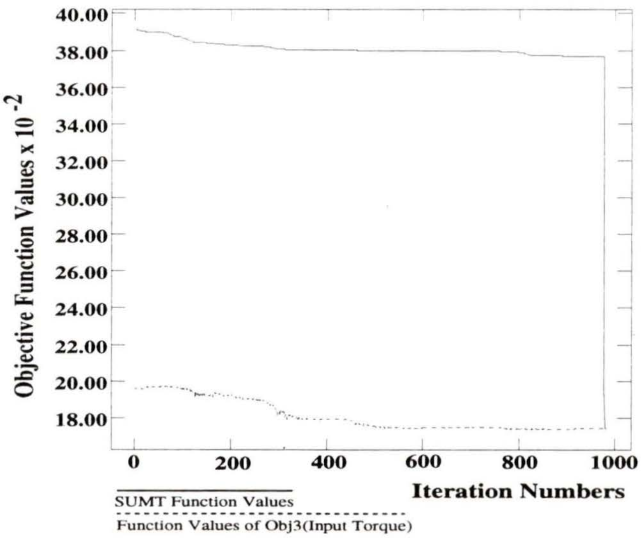

Figure 2.19: Optimization Results ($f_{obj_3} = \max_{0° \leq \Delta\theta_2 \leq 200°} T_{input}$)

A final objective to be considered was a weighted combination of second and third objectives, i.e.,

$$f_{obj_{combined}} = w_2 f_{obj_2} + w_3 f_{obj_3} \tag{2.28}$$

The optimal mechanism for the combined objective of equation (2.28) has shown about a 23 percent of improvement in terms of the weighted function. Specifically, minimum transmission angle increases about 58 percent of the original value and maximum torque is approximately 87 percent of the starting feasible mechanism.

The displacements for the combined-objective optimize mechanism are shown in the graphs of Figs. (2.7) - (2.9) as dotted lines. Notice that they have moved right up to some of the constraint limits. This demonstrates that the motion constraints are limiting factors in the search. The search for the optimum mechanism required 966 iterations for the combined objective, and resulted in the f_{SUMT} and f_{obj} values during searching as illustrated in Fig. 2.20. The input torque requirement is illustrated in Figure 2.21. The transmission angle results are shown in Figure 2.22 and the final mechanism and coupler curve are depicted in Figure 2.23.

2.8.2 A Path Generation Problem

It is desired to drive a conveyor with the coupler point of a four-bar mechanism as shown in Figure 2.24. The conveyor is to move during crank angles of $\Delta\theta_2 = 30°$ to $135°$, and remain stationary for the remainder of the crank revolution. To achieve this task, a task model was made with upper and lower displacement component constraints as illustrated in Figs. 2.25 and 2.26. An initial mechanism $\mathbf{r}_{initial} = \{120, 30, 90, 75, 70, 30\}^T$ cm with $\theta_{2_0} = 10°$ and $\theta_{local} = 0°$ resulted in the motion indicated with a solid line on the graphs of Figs. 2.25 and 2.26. Constraints on the lengths of the links of $\mathbf{r}_{upper} = \{175, 100, 165, 170, 230, 150\}^T$ cm and $\mathbf{r}_{lower} = \{25, 10, 25, 25, 5, 0\}^T$ cm exist. The initial value of the displacement of the crank (θ_{2_0}) is allowed to range from $0°$ to $360°$ and the values of θ_{local} are constrained to $\pm 90°$.

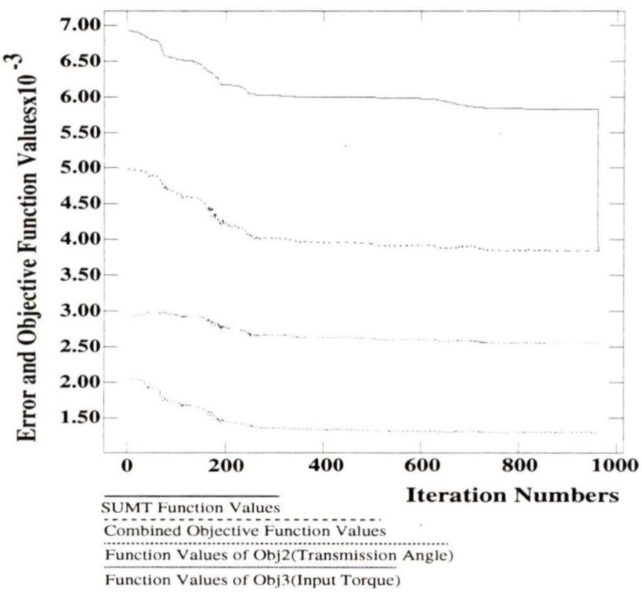

Figure 2.20: Optimization Results ($f_{obj_{combined}} = w_2 f_{obj_2} + w_3 f_{obj_3}$)

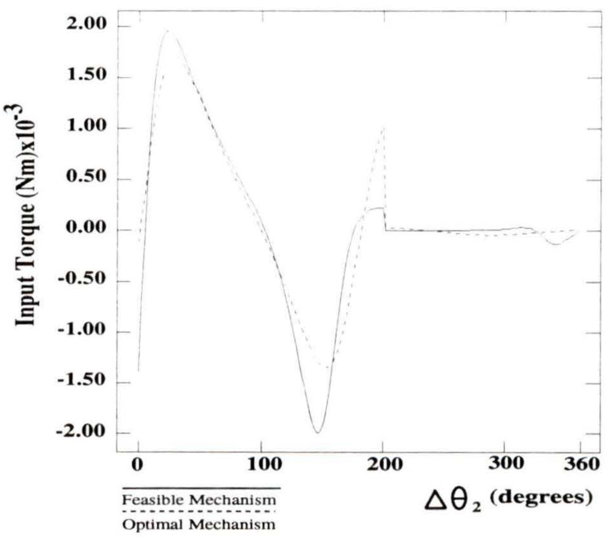

Figure 2.21: Input Torque Requirement (combined objective)

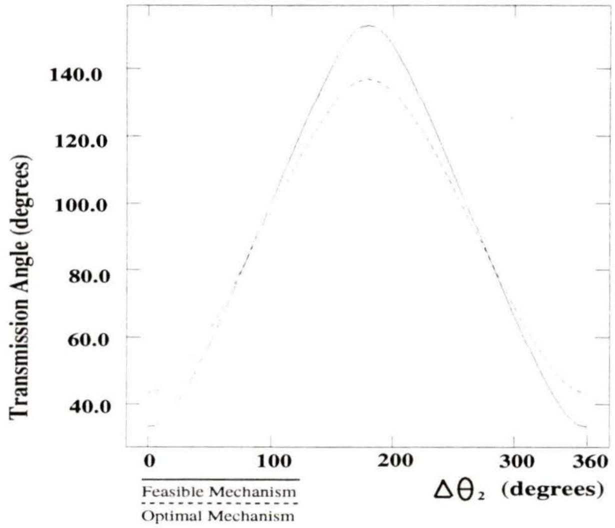

Figure 2.22: Transmission Angle Results (combined objective)

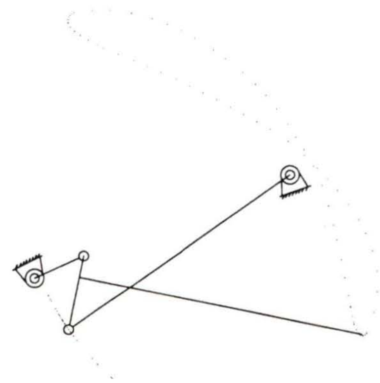

Figure 2.23: Lifting Mechanism and its Coupler Curve

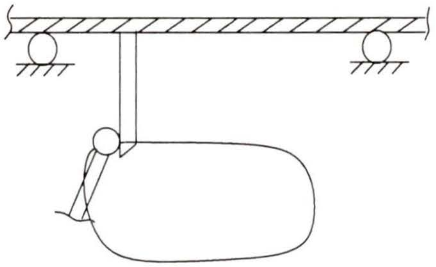

Figure 2.24: Conveyor (Path Generator)

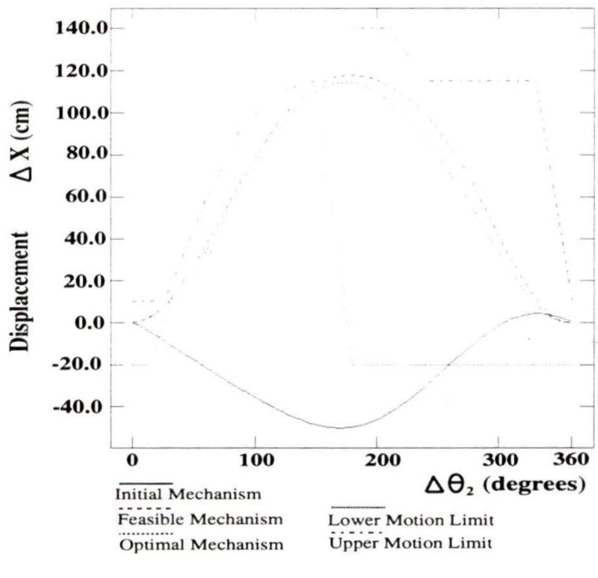

Figure 2.25: Task Specification and Resulting Mechanism Motion – ${}^{g}\Delta X_P$

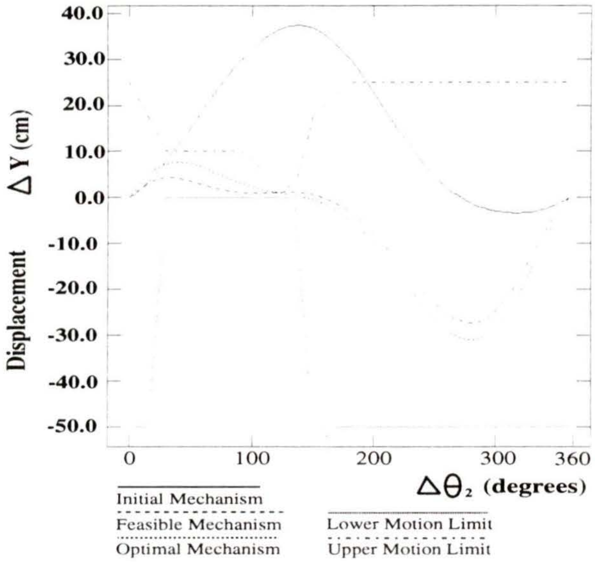

Figure 2.26: Task Specification and Resulting Mechanism Motion – ${}^{g}\Delta Y_P$

A least pth search (as discussed in Section 2.6.2) combined with the sub-type specific transforms (as discussed in Chapter 3) rendered a feasible crank-rocker with $\mathbf{r}_{feasible} = \{71.96, 16.5, 62.85, 49.8, 230, 149.89\}^T$ cm with $\theta_{2_0} = 48.1°$ and $\theta_{local} = -90°$ The least pth search process is illustrated in Fig. 2.27. The displacement output for this feasible mechanism is indicated by the dashed lines on Figs. 2.25 and 2.26. Further least pth searches based on varying the initial mechanism have also been performed, resulting in the feasible mechanisms included in Table 2.4.

The crank angle of $\Delta\theta_2$ =30° corresponds approximately to the crank angle at which the coupler point would initially make contact with the conveyor. Simulation of the least pth rendered mechanism found it to have a velocity value of 386 cm/s in x direction at $\Delta\theta_2 = 30°$ as illustrated by the solid line on Fig. 2.28. To minimize the shock it would be appropriate to minimize the ${}^{g}X_P$ component of velocity at this

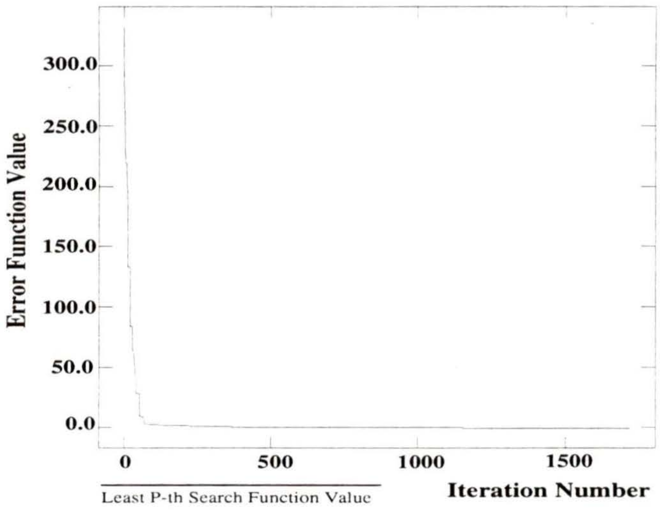

Figure 2.27: Least pth Searching Process

ini. mechanism $\mathbf{r}_{initial}$	f_{1st} value	feasible mechanism $\mathbf{r}_{feasible}$	No. of iterations	f_{final} value
120,30,90,75, 70,30,10,0	309.915	174.9,25.17,83.13,127.6, 230,0.0067,26.58,-62.86	1623	-0.449
100,25,90,45, 30,30,20,0	297.827	175,10,75.21,109.79, 230,13.95,34,81.02	2753	-0.294
100,25,45,90, 30,30,20,0	173.624	175,10,66.36,119.21, 230,0.26,26.36,79.1	1111	-0.378
120,30,90,75, 70,30,10,0	331.872	71.96,16.5,62.84,49.81, 230,149.89,48.06,-89.99	1720	-0.343

Table 2.4: Least pth Search Results for the Conveyor Mechanism

crank angle. The objective

$$f_{obj_1} = {}^{g}\dot{X}_{P\,\Delta\theta_2=30^\circ} \tag{2.29}$$

was considered.

The input torque and mechanism pin forces are dependent upon the inertia and acceleration of the mechanism members and upon the externally applied forces. In

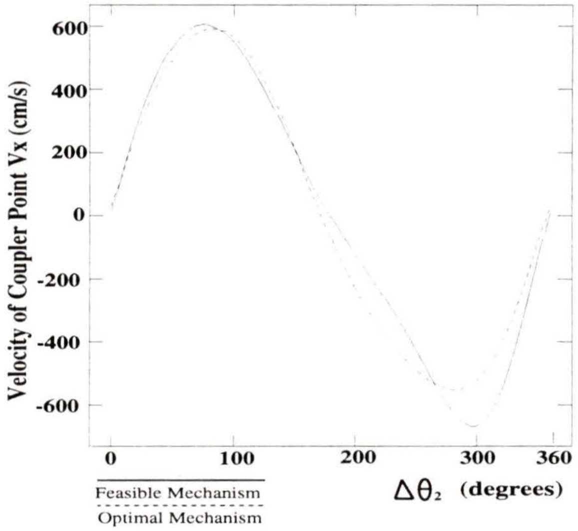

Figure 2.28: Coupler Point Velocity ${}^{g}\dot{X}_P$ - Feasible and Optimal Mechanisms for f_{obj_1}

this case, the exterior force applied to the four-bar mechanism is caused by the resistance of the conveyor to motion. This resistance is in part dependent upon the acceleration of the conveyor and therefore upon the acceleration of ${}^{g}\ddot{X}_P$ for $\Delta\theta_2 = 30 - 135°$. Simulation of the least pth rendered mechanism found it to have a maximum acceleration value of 54.98 m/s^2 in x direction at the coupler point P during the push phase ($\Delta\theta_2 = 30 - 135°$). The x component of the coupler point acceleration is shown by the solid line on Fig. 2.29. It would be desirable to minimize this acceleration value. An objective of minimizing the maximum of the acceleration of P, i.e.,

$$f_{obj_2} = \max_{30° \leq \Delta\theta_2 \leq 135°} {}^{g}\ddot{X}_P \tag{2.30}$$

was thus considered.

The SUMT discussed in Section 2.6.3, in combination with the sub-type specific transformation method, was applied, resolving optimal mechanisms as listed in Table 2.5. From the results shown in Table 2.5, the optimized mechanism for f_{obj_1} has a velocity of 351 cm/s in the X direction, which is approximately 90.8 percent of the velocity value for the initial feasible mechanism (see the dashed line on Fig. 2.28). The optimized mechanism for f_{obj_2} has a maximum acceleration value of 48.04 m/s^2 and shows a 13 percent reduction in the acceleration value of the initial mechanism (see the dashed line on Fig. 2.29).

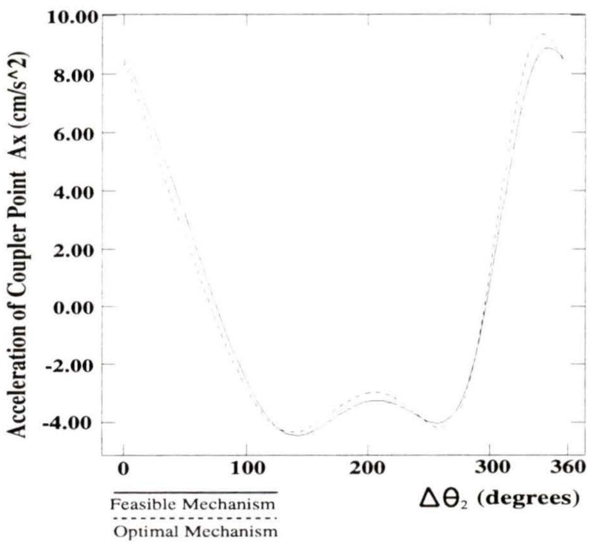

Figure 2.29: Coupler Point Acceleration ${}^{g}\ddot{X}_P$ - Feasible and Optimal Mechanisms for f_{obj_2}

Objective	f_{obj}value before opt.	f_{obj}value after opt.	No. of iterations	Optimized mechanism (cm)
Obj. 1	386 cm/s	351 m/s	919	136.2,24.0,88.3,108.7, 225.3,149.99,24.74,-90
Obj. 2	5498 cm/s^2	4804 cm/s^2	209	97.95,24.48,87.3,75.2, 229.97,149.9,47.21,-90
Comb.Obj.	10990.0	10153.04	862	120.2,23.1,85.6,94, 230.0,149.8,29.74,-90
Obj. 1	386 cm/s	350.1 cm/s		
Obj. 2	5498 cm/s^2	5167.79cm/s^2		

Table 2.5: Optimization Results for the Conveyor Mechanism

In addition, a third objective, formed from a weighted combination of the objectives in equations (2.29) and (2.30), was considered, i.e.,

$$f_{obj_{combined}} = w_1 f_{obj_1} + w_2 f_{obj_2} \tag{2.31}$$

The optimized mechanism for this combined objective showed about 7.6 percent reduction in combined objective function value, with the velocity value decreasing about 9.3 percent of the original value and the maximum acceleration value decreasing about 6 percent of the original value (see the dashed line on Figs. 2.31 and 2.32). The case of the combined objective also showed that improvement of contact velocity went with the sacrifice of the improvement in maximum acceleration.

The displacements for the "combined objective optimum" mechanism are shown in the graphs of Figs. 2.25 and 2.26 as dotted lines. Notice that they have moved right up to some of the constraint limits. This demonstrates that the motion constraints are limiting factors in the search. The search for the optimum mechanism required 862 iterations for the combined objective, and resulted in the f_{SUMT} and f_{obj} values during searching illustrated in Fig. 2.30. The corresponding velocity, acceleration and transmission angle are given by the dashed lines on Figs. 2.31, 2.32 and 2.33, respectively. The transmission angle of the optimized mechanism was checked for acceptability. As seen in Fig. 2.33, the minimum transmission angle for the optimized mechanism is approximately 65°. This is a very acceptable value, so no further optimization considering transmission angle is necessary. The final mechanism and its coupler curve are depicted in Fig. 2.34.

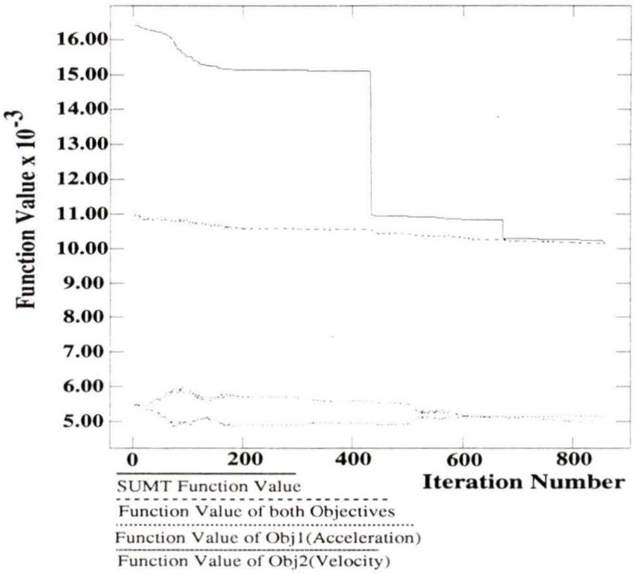

Figure 2.30: Optimization Results ($f_{obj} = w_1({}^g\dot{X}_{P\,\Delta\theta_2=30^\circ}) + w_2(\max_{30^\circ \leq \Delta\theta_2 \leq 135^\circ}{}^g\ddot{X}_P)$)

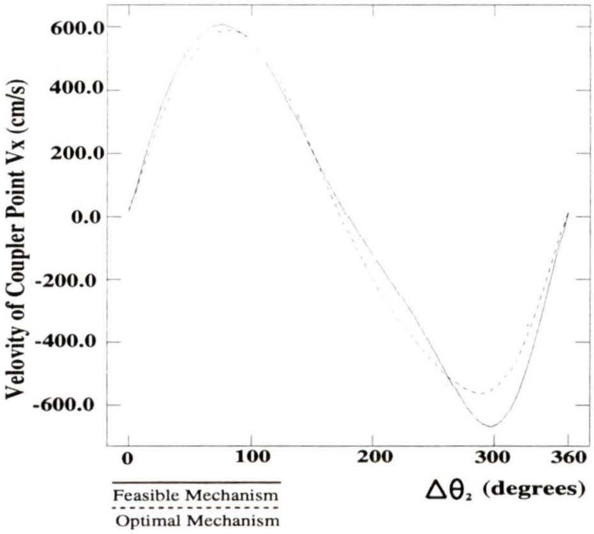

Figure 2.31: Coupler Point Velocity ${}^g\dot{X}_P$ - Feasible and Optimal Mechanisms for $f_{obj_{combined}}$

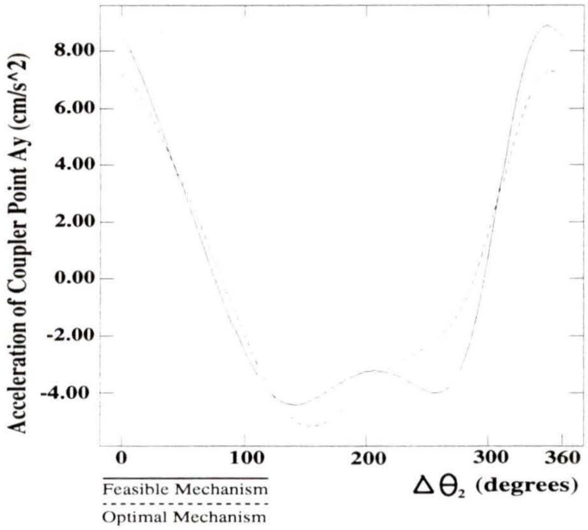

Figure 2.32: Coupler Point Acceleration ${}^{g}\ddot{X}_{P}$ - Feasible and Optimal Mechanisms for $f_{obj_{combined}}$

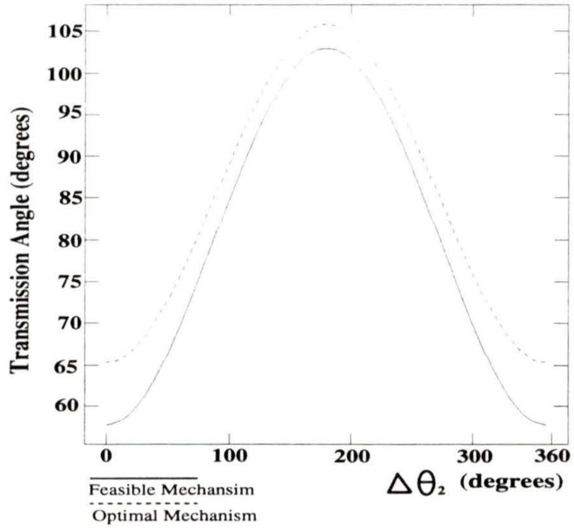

Figure 2.33: Transmission Angle - Feasible and Optimal Mechanisms

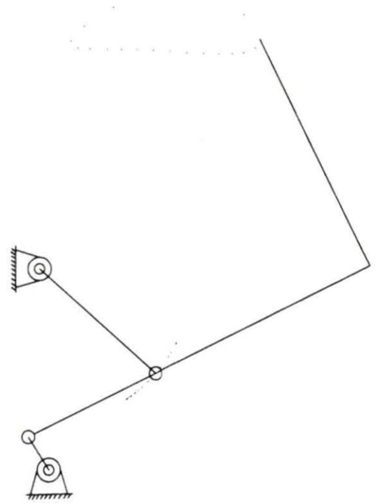

Figure 2.34: Mechanism of Conveyor and its Coupler Curve

2.9 Discussion

In this chapter, several issues related to mechanism optimization were discussed. These issues included associated mechanism design parameters, motion specification, relevant constraints and design objectives.

In previous work, required task motions have generally been specified with respect to an absolute/global coordinate system. As introduced in this chapter, it is very convenient to specify a task in terms of required changes in the location of the point of interest. The advantage of doing so is that the number of search variables can be reduced. Thus, the effort of searching optima will be decreased. However, it is important to note that in certain cases, task specification in terms of required changes will not be possible. One example, where such specification is not possible, is the case where desired base pin locations are known relative to the global frame of reference.

Direct search methods that rely upon only objective and constraint function values to guide the search were considered in the optimization process. The reason for considering such methods is that in engineering applications, such as mechanism synthesis, it is frequently necessary to solve problems whose functions are discontinuous or non-differentiable. In such cases, function-value-based approaches are most appropriate.

Even the best direct methods can require a large number of function evaluations to locate the optimal solution. However, it is important to note that mechanism synthesis is not a "real-time" problem. The numerical results of the examples in Section 2.8 and later in Chapter 4 showed that function evaluations (typically around 500-2500 depending on task objective, convergence tolerances and search routine parameters) and computing time (typically 15 seconds to four minutes on a Sun Sparc Workstation) are not unreasonable. "Tuning" search routine parameters for particular mechanism types and tasks would lead to faster convergence. Given the reasonable performance in convergence, no major effort was made in "tuning" for faster convergence.

It may be of interest to examine the implementation of gradient/Hessian-based search methods for specific tasks in future research. It is important to note that the unconstrained to constrained search parameter transformation discussed in this chapter, and the Grashof-criteria based transforms (discussed in detail in Chapter 3), will complicate the process of finding derivatives. Implementing derivative-based search routines will probably require the use of numerical methods for finding the required derivatives.

A SUMT was utilized in the mechanism optimization approach. The basic idea of a SUMT is to construct a composite function using the objective and constraint functions. It also contains certain parameters - called the penalty parameters - that weight the function for violation of constraints. Unconstrained minimization techniques consist of two basically different types of penalty functions: a barrier(interior) function, which adds a term that favors points interior to the feasible region over those near the boundary; and/or a penalty(exterior) function, which adds a term that produces a high cost for violation of the constraints. The SUMT barrier method that is used in this thesis always iterates within the feasible region, so the final design will always be feasible and hence usable. However, the starting design point must be feasible. Exterior penalty methods can have an arbitrary starting design point and can iterate through infeasible regions where the problem functions may be undefined. However, if the iterative process terminates prematurely, the final design may not be feasible and hence not usable.

Two example optimization problems were discussed. The optimized mechanisms obtained represent only local optima for the specified motion constraints and objectives. This is evident in the examples where combined objectives produced better results for specific objective terms than those found by the specific objective searches. It is difficult to find global optima because of the non-linearity of the mechanism synthesis problems. More searches starting from different initial mechanisms and search parameters would be necessary to instill confidence in the found optima. Different initial mechanisms can be generated, (and were in this work), by building new initial mechanisms around previous local minima with varied scale factors and restarting the least pth search.

Chapter 3

Grashof-Criterion-Based Transformations Allowing Unconstrained Searching

3.1 Grashof and Non-Grashof Mechanisms

A Grashof mechanism allows complete rotation of at least one link relative to an adjacent link. Conversely a non-Grashof mechanism has no links with the ability of complete rotation and hence will not be suitable for continuous single-input-direction application. Geometric considerations allow the classification of mechanism sub-types based on relative link lengths. As an example, by considering geometry, the criteria for a four-bar to be a Grashof mechanism can be seen to be [39]:

$$r_{short} + r_{long} \leq r_a + r_b \tag{3.1}$$

where r_{short} and r_{long} are respectively the shortest and the longest lengths among the frame, crank, coupler and follower, and r_a and r_b are the remaining two lengths.

Further examination of the geometry and resulting motions leads to Grashof's division of four-bars into sub-types capable of complete rotation of a link(s). Specifically, these further sub-types are:

⟶ $r_{short} = r_2$: crank-rockers, (crank is the shortest link, output "rocks" between two extreme positions, Fig. 3.1-A;

⟶ $r_{short} = r_1$: drag-links, (frame is the shortest link, follower is "dragged" through a full rotation, Fig. 3.1-B;

⟶ $r_{short} = r_3$: rocker-rockers, (coupler is the shortest link and is capable of full rotation, the "crank" and the follower rock between respective extreme displacements, Fig. 3.1-C;

For non-Grashof four-bars,

$$r_{short} + r_{long} > r_a + r_b \tag{3.2}$$

All non-Grashof mechanisms result in triple-rockers, where no link is capable of a complete rotation relative to an adjacent link.

3.2 Transformation Based on Redefining Link Length Limits (4-bar Crank-Rocker)

Equation 2.17 presented a transform allowing a mapping of an unconstrained set of search variables, $\mathbf{x}$, to parameters, $\mathbf{r}$, constrained by upper and lower length limits, $\mathbf{r}_{upper}$ and $\mathbf{r}_{lower}$. For link lengths, however, this mapping does not ensure that the rendered link lengths belong to a specific Grashof sub-type of the mechanism or even that the designed mechanism will be Grashof mechanism. Mapping to specific mechanism sub-types can be achieved by considering Grashof-type criteria to generate

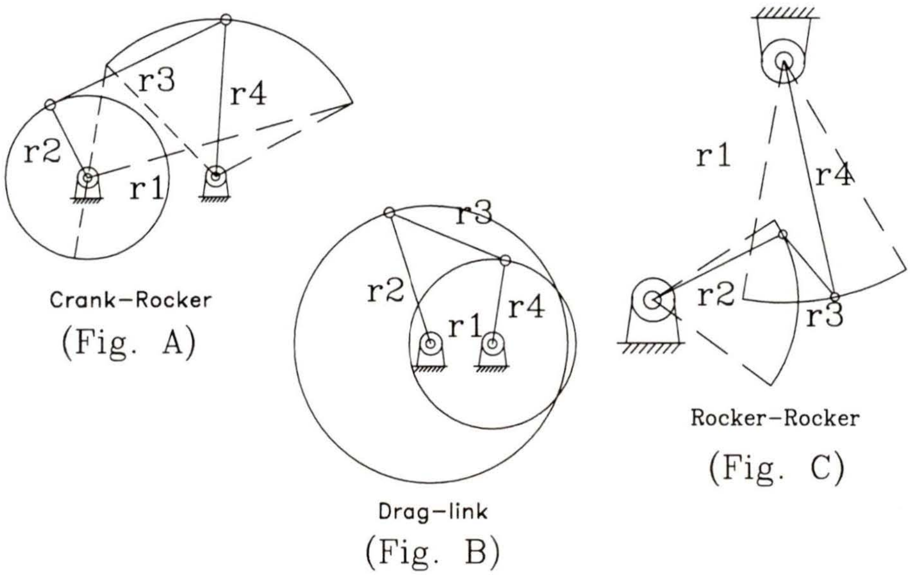

Figure 3.1: Planar Four-bar Mechanism Sub-types

appropriate rule sets for the redefinition of $\mathbf{r}_{upper}$ and $\mathbf{r}_{lower}$ into $\mathbf{r}^*_{upper}$ and $\mathbf{r}^*_{lower}$ which ensure transformation to constrained lengths belonging to the desired mechanism sub-type. That is, eqn.(2.17) becomes

$$\mathbf{r}_i \;=\; \mathbf{r}^*_{lower_i} + \frac{\mathbf{r}^*_{upper_i} - \mathbf{r}^*_{lower_i}}{e^{-\mathbf{x}_i} + 1} \tag{3.3}$$

As an example, consider a crank-rocker mechanism ($r_{short} = r_2$) having the frame as the longest ($r_{long} = r_1$) and associated link length limits $\mathbf{r}_{upper}$ and $\mathbf{r}_{lower}$. The lengths of an acceptable four-bar within this sub-type must satisfy:

$$\begin{gathered} r_{short} = r_2; \quad r_{long} = r_1; \quad r_2 + r_1 \;\leq\; r_3 + r_4 \\ r_i \;\leq\; r_{upper_i}, \; i = 1,4; \quad r_i \;\geq\; r_{lower_i}, \; i = 1,4 \end{gathered} \tag{3.4}$$

Transformation of the unconstrained search variables x_i, i=1,4 to link lengths r_i, i=1,4 satisfying the constraints of eqn. (3.4) is desired. Assumptions of $\mathbf{r}_{upper}$ and $\mathbf{r}_{lower}$ compatibility with the desired four-bar sub-type must be made. For example, for this mechanism sub-type

$$\begin{aligned} r_{upper_i} &> r_{upper_2},\ i = 1,3,4 \\ r_{upper_i} &< r_{upper_1},\ i = 2,3,4 \\ r_{upper_2} + r_{upper_1} &< r_{upper_3} + r_{upper_4} \\ r_{lower_i} &> r_{lower_2},\ i = 1,3,4 \end{aligned} \tag{3.5}$$

The transformation from **x** to **r**, based on redefined upper and lower link length limits for a crank-rocker having the base link the longest, is as follows:

For $x_2 \longrightarrow r_2$:

(CRa) $r^*_{lower_2} = r_{lower_2}$

(CRb) $r^*_{upper_2} = r_{upper_2}$

(CRc) Apply equation 3.3 to find r_2

For $x_1 \longrightarrow r_1$:

(CRd) If: $r_2 > r_{lower_1}$; then $r^*_{lower_1} = r_2$; else: $r^*_{lower_1} = r_{lower_1}$

(CRe) $r^*_{upper_1} = r_{upper_1}$

(CRf) Apply equation 3.3 to find r_1

For $x_3 \longrightarrow r_3$:

(CRg) If: $r_2 > r_{lower_3}$; then $r^*_{lower_3} = r_2$; else: $r^*_{lower_3} = r_{lower_3}$

(CRh) If: $r_1 < r_{upper_3}$; then $r^*_{upper_3} = r_1$; else: $r^*_{upper_3} = r_{upper_3}$

(CRi) Apply equation 3.3 to find r_3

For $x_4 \longrightarrow r_4$:

(CRj) If: $r_2 > r_{lower_4}$; then $r^*_{lower_4} = r_2$; else: $r^*_{lower_4} = r_{lower_4}$

(CRk) If: $r_1 < r_{upper_4}$; then $r^*_{upper_4} = r_1$; else: $r^*_{upper_4} = r_{upper_4}$

(CRl) If: $r^*_{lower_4} + r_3 < r_1 + r_2$; then $r^*_{lower_4} = r_1 + r_2 - r_3$; else: $r^*_{lower_4} = r^*_{lower_4}$

(CRm) If: $r^*_{upper_4} > r^*_{lower_4}$; then goto (CRn); else: redefine the lower limits of $r^*_{lower_3}$ and $r^*_{lower_4}$ as follows (see Appendix B):

$$r^*_{lower_3} = \frac{Or_3(r_1+r_2-r^*_{upper_4})+Or_4 r^*_{upper_3}}{Or_3+Or_4}$$
$$r^*_{lower_4} = \frac{Or_4(r_1+r_2-r^*_{upper_3})+Or_3 r^*_{upper_4}}{Or_3+Or_4}$$

where $Or_3 = r_{upper_3} - r_{lower_3}$, and $Or_4 = r_{upper_4} - r_{lower_4}$

Apply equation 3.3 to find r_3

(CRn) Apply equation 3.3 to find r_4

The above redefinitions for r_{lower_i} and r_{upper_i} ensure that the transformation of equation 3.3 will return a crank-rocker mechanism having $r_{long} = r_1$. Specifically,

- CRa, d, g and j ensure that $r_{short} = r_2$; CRb, e, h and k ensure that $r_{long} = r_1$;
- CRl and m ensures that $r_1 + r_2 \leq r_3 + r_4$;
- CRc, f, i and n map the unconstrained $\mathbf{x}$ to $\mathbf{r}$ corresponding to a constrained crank-rocker ($r_{long} = r_1$) mechanism.

The steps (CRa) to (CRl) evolve directly from the crank-rocker Grashof-criteria being considered during a sequential mapping of $\mathbf{x}$ elements to $\mathbf{r}$ values. Step (CRm) is a check to ensure compatibility of the redefined limits. The step becomes necessary when the r_3 rendered by (CRg) to (CRi) is too short to allow r_4 to remain within its limits. The redefinitions for $r^*_{lower_3}$ and $r^*_{lower_4}$ applied in (CRm) effectively ensure sufficiency of the r_3 and r_4 values. The redefinition is based on ensuring that the relative ranges available for r_3 and r_4 are the same as the original ranges Or_3 and Or_4, i.e., $\frac{r^*_{upper_4}-r^*_{lower_4}}{Or_4} = \frac{r^*_{upper_3}-r^*_{lower_3}}{Or_3}$ and the minimal satisfaction of the Grashof-criteria, i.e., $r^*_{lower_3} + r^*_{lower_4} = r_1 + r_2$. In Appendix B the redefinition of $r^*_{lower_3}$ and $r^*_{lower_4}$ in CRm is shown to meet the applicable compatibility requirements of equation (3.5).

The CR transformations ensure that lengths $\mathbf{r}$ corresponding to a crank-rocker mechanism, within the required length limits, will be generated for unconstrained search variables $\mathbf{x}$. The least-pth and optimization searches for the examples of section 2.7 exploited these transformations.

Transformation rule sets for four-bar crank-rockers having other links as the longest link (i.e., $r_{long} = r_4$ or $r_{long} = r_3$) result from the exchange of the other link for r_1 in the above rule set. Rule sets can also be easily generated for other single-loop and multiple-loop planar mechanisms. Section 3.3 and 3.4 give further examples.

3.3 Transformations for Other Four-bar Mechanisms

3.3.1 Transformation for Drag-Link Mechanisms

The transformation from $\mathbf{x}$ to $\mathbf{r}$, based on redefined upper and lower link length limits, for a drag-link ($r_{short} = r_1$) having link 4 as the longest link ($r_{long} = r_4$) is as follows:

For $x_1 \longrightarrow r_1$:

(DLa) $r^*_{lower_1} = r_{lower_1}$

(DLb) $r^*_{upper_1} = r_{upper_1}$

(DLc) Apply equation 3.3 to find r_1

For $x_4 \longrightarrow r_4$:

(DLd) If: $r_1 > r_{lower_4}$; then $r^*_{lower_4} = r_1$; else: $r^*_{lower_4} = r_{lower_4}$

(DLe) $r^*_{upper_4} = r_{upper_4}$

(DLf) Apply equation 3.3 to find r_4

For $x_2 \longrightarrow r_2$:

(DLg) If: $r_1 > r_{lower_2}$; then $r^*_{lower_2} = r_1$; else: $r^*_{lower_2} = r_{lower_2}$

(DLh) If: $r_4 < r_{upper_2}$; then $r^*_{upper_2} = r_4$; else: $r^*_{upper_2} = r_{upper_2}$

(DLi) Apply equation 3.3 to find r_2

For $x_3 \longrightarrow r_3$:

(DLj) If: $r_1 > r_{lower_3}$; then $r^*_{lower_3} = r_1$; else: $r^*_{lower_3} = r_{lower_3}$

(DLk) If: $r_4 < r_{upper_3}$; then $r^*_{upper_3} = r_4$; else: $r^*_{upper_3} = r_{upper_3}$

(DLl) If: $r^*_{lower_3} + r_2 < r_1 + r_4$; then $r^*_{lower_3} = r_1 + r_4 - r_2$; else: $r^*_{lower_3} = r^*_{lower_3}$

(DLm) If: $r^*_{upper_3} > r^*_{lower_3}$; then goto (DLn); else: redefine the lower limits of $r^*_{lower_3}$ and $r^*_{lower_2}$ as follows (see Appendix B):

$$r^*_{lower_2} = \frac{Or_2(r_1+r_4-r^*_{upper_3})+Or_3 r^*_{upper_2}}{Or_2+Or_3}$$

$$r^*_{lower_3} = \frac{Or_3(r_1+r_4-r^*_{upper_2})+Or_2 r^*_{upper_3}}{Or_2+Or_3}$$

where $Or_3 = r_{upper_3} - r_{lower_3}$, and $Or_2 = r_{upper_2} - r_{lower_2}$

Apply equation 3.3 to find r_2

(DLn) Apply equation 3.3 to find r_3

The above redefinitions for r_{lower_i} and r_{upper_i} ensure that the transformation of equation 3.3 will return a drag-link mechanism having $r_{long} = r_4$. Specifically,

- DLa, d, g and j ensure that $r_{short} = r_1$; DLb, e, h and k ensure that $r_{long} = r_4$;
- DLl and m ensure that $r_1 + r_2 \leq r_4 + r_3$;
- DLc, f, i and n map the unconstrained $\mathbf{x}$ to $\mathbf{r}$ corresponding to a constrained drag-link ($r_{long} = r_4$) mechanism.

Transformation rule sets for four-bar drag-links having other links as the longest link (i.e. $r_{long} = r_2$ or $r_{long} = r_3$) result from the exchange of the other link for r_4 in the above rule set.

3.3.2 Transformation for Double-Rocker Mechanisms

The transformation from **x** to **r**, based on redefined upper and lower link length limits, for a double-rocker ($r_{short} = r_3$) having frame 1 as the longest link ($r_{long} = r_1$) is as follows:

For $x_3 \longrightarrow r_3$:

(DRa) $r^*_{lower_3} = r_{lower_3}$

(DRb) $r^*_{upper_3} = r_{upper_3}$

(DRc) Apply equation 3.3 to find r_3

For $x_1 \longrightarrow r_1$:

(DRd) If: $r_3 > r_{lower_1}$; then $r^*_{lower_1} = r_3$; else: $r^*_{lower_1} = r_{lower_1}$

(DRe) $r^*_{upper_1} = r_{upper_1}$

(DRf) Apply equation 3.3 to find r_1

For $x_2 \longrightarrow r_2$:

(DRg) If: $r_3 > r_{lower_2}$; then $r^*_{lower_2} = r_3$; else: $r^*_{lower_2} = r_{lower_2}$

(DRh) If: $r_1 < r_{upper_2}$; then $r^*_{upper_2} = r_1$; else: $r^*_{upper_2} = r_{upper_2}$

(DRi) Apply equation 3.3 to find r_2

For $x_4 \longrightarrow r_4$:

(DRj) If: $r_3 > r_{lower_4}$; then $r^*_{lower_4} = r_3$; else: $r^*_{lower_4} = r_{lower_4}$

(DRk) If: $r_1 < r_{upper_4}$; then $r^*_{upper_4} = r_1$; else: $r^*_{upper_4} = r_{upper_4}$

(DRl) If: $r_{lower_4} + r_2 < r_1 + r_3$; then $r^*_{lower_4} = r_1 + r_3 - r_2$; else $r^*_{lower_4} = r_{lower_4}$

(DRm) If: $r^*_{upper_4} > r^*_{lower_4}$; then goto (DRn); else: redefine the lower limits of $r^*_{lower_4}$ and $r^*_{lower_2}$ as follows (see Appendix B):

$$r^*_{lower_2} = \frac{Or_2(r_1+r_3-r^*_{upper_4})+Or_4 r^*_{upper_2}}{Or_2+Or_4}$$

$$r^*_{lower_4} = \frac{Or_4(r_1+r_3-r^*_{upper_2})+Or_2 r^*_{upper_4}}{Or_2+Or_4}$$

where $Or_4 = r_{upper_4} - r_{lower_4}$, and $Or_2 = r_{upper_2} - r_{lower_2}$

Apply equation 3.3 to find r_2

(DRn) Apply equation 3.3 to find r_4

The above redefinitions for r_{lower_i} and r_{upper_i} ensure that the transformation of equation 3.3 will return a double-rocker mechanism having $r_{long} = r_1$. Specifically,

- DRa, d, g and j ensure that $r_{short} = r_3$; DRb, e, h and k ensure that $r_{long} = r_1$;
- DRl and m ensure that $r_1 + r_2 \leq r_4 + r_3$;
- DRc, f, i and n map the unconstrained $\mathbf{x}$ to $\mathbf{r}$ corresponding to a constrained double-rocker ($r_{long} = r_1$) mechanism.

Transformation rule sets for four-bar double-rocker having other links as the longest link (i.e. $r_{long} = r_2$ or $r_{long} = r_4$) result from the exchange of the other link for r_1 in the above rule set.

3.4 Transformations for Six-bar Mechanisms

3.4.1 Transformation for Watt I Six-bar Mechanisms

The transformation from $\mathbf{x}$ to $\mathbf{r}$, based on redefined upper and lower link length limits, for a Watt I six-bar mechanism as shown in Fig. 3.2 having link 2 as the shortest link ($r_{short} = r_2$) and having the frame 1 as the longest link ($r_{long} = r_1$) for the four-bar base loop, is as follows:

For $x_2 \longrightarrow r_2$:

(WIa) $r^*_{lower_2} = r_{lower_2}$

(WIb) $r^*_{upper_2} = r_{upper_2}$

(WIc) Apply equation 3.3 to find r_2

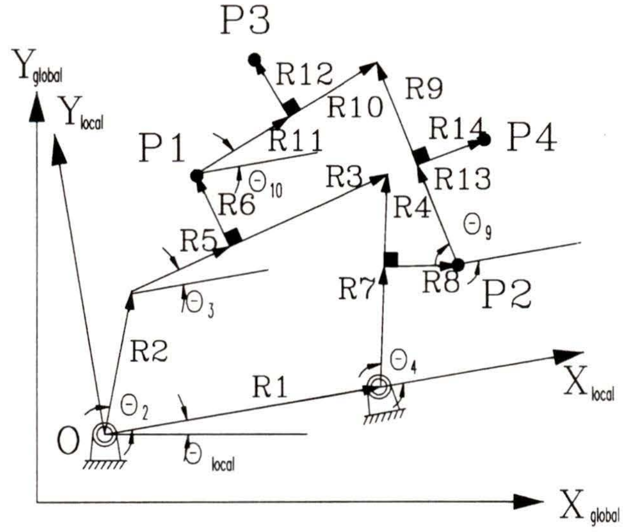

Figure 3.2: Six-Bar (Watt I) Vector Representation

For $x_1 \longrightarrow r_1$:

(WId) If: $r_2 > r_{lower_1}$; then $r^*_{lower_1} = r_2$; else: $r^*_{lower_1} = r_{lower_1}$

(WIe) $r^*_{upper_1} = r_{upper_1}$

(WIf) Apply equation 3.3 to find r_1

For $x_3 \longrightarrow r_3$:

(WIg) If: $r_2 > r_{lower_3}$; then $r^*_{lower_3} = r_2$; else: $r^*_{lower_3} = r_{lower_3}$

(WIh) If: $r_1 < r_{upper_3}$; then $r^*_{upper_3} = r_1$; else: $r^*_{upper_3} = r_{upper_3}$

(WIi) Apply equation 3.3 to find r_3

For $x_4 \longrightarrow r_4$:

(WIj) If: $r_2 > r_{lower_4}$; then $r^*_{lower_4} = r_2$; else: $r^*_{lower_4} = r_{lower_4}$

(WIk) If: $r_1 < r_{upper_4}$; then $r^*_{upper_4} = r_1$; else: $r^*_{upper_4} = r_{upper_4}$

(WIl) If: $r_{lower_4} + r_3 < r_1 + r_2$; then $r^*_{lower_4} = r_1 + r_2 - r_3$; else: $r^*_{lower_4} = r_{lower_4}$

(WIm) If: $r^*_{upper_4} > r^*_{lower_4}$; then goto (WIn); else: redefine the lower limits of $r^*_{lower_3}$ and $r^*_{lower_4}$ as follows (see Appendix B):

$$r^*_{lower_3} = \frac{Or_3(r_1+r_2-r^*_{upper_4})+Or_4 r^*_{upper_3}}{Or_3+Or_4}$$

$$r^*_{lower_4} = \frac{Or_4(r_1+r_2-r^*_{upper_3})+Or_3 r^*_{upper_4}}{Or_3+Or_4}$$

where $Or_3 = r_{upper_3} - r_{lower_3}$, and $Or_4 = r_{upper_4} - r_{lower_4}$.

Apply equation 3.3 to find r_3

(WIn) Apply equation 3.3 to find r_4

For $x_i \longrightarrow r_i$, i= 5 to 9, 11 & 12:

(WIn) $r^*_{lower_i} = r_{lower_i}$, i= 5 to 9, 11 & 12

(WIo) $r^*_{upper_i} = r_{upper_i}$, i= 5 to 9, 11 & 12

(WIp) Apply equation 3.3 to find r_i

For $x_{10} \longrightarrow r_{10}$:

(WIq) $d_{max} = max(\sqrt{(x_{P1}-x_{P2})^2+(y_{P1}-y_{P2})^2})$;

$d_{min} = min(\sqrt{(x_{P1}-x_{P2})^2+(y_{P1}-y_{P2})^2})$. [1]

(WIr) If: $r_{lower_{10}} < |d_{max} - r_9|$; then $r^*_{lower_{10}} = |d_{max} - r_9|$;

else: $r^*_{lower_{10}} = r_{lower_{10}}$

(WIs) If: $r^*_{lower_{10}} < |r_9 - d_{min}|$; then $r^*_{lower_{10}} = |r_9 - d_{min}|$;

(WIt) If: $r_{upper_{10}} > (d_{min} + r_9)$; then $r^*_{upper_{10}} = (d_{min} + r_9)$; else: $r^*_{upper_{10}} = r_{upper_{10}}$

(WIu) If: $r^*_{upper_{10}} > r^*_{lower_{10}}$; then goto (WIv); else, redefine the lower and upper limits of $r^*_{lower_9}, r^*_{upper_9}, r^*_{lower_{10}}$ and $r^*_{upper_{10}}$ as follows (see Appendix B):

$$r^*_{lower_9} = \frac{Or_9(dmax-dmin)+Or_{10}(dmax+dmin)}{2(Or_9+Or_{10})}$$

$$r^*_{upper_9} = \frac{(Or_9(dmax+3dmin)+Or_{10}(dmax+dmin)}{2(Or_9+Or_{10})}$$

$$r^*_{lower_{10}} = \frac{Or_9(dmax+dmin)+Or_{10}(dmax-dmin)}{2(Or_9+Or_{10})}$$

$$r^*_{upper_{10}} = \frac{Or_9(dmax+dmin)+Or_{10}(dmax+3dmin)}{2(Or_9+Or_{10})}$$

where $Or_9 = r_{upper_9} - r_{lower_9}$, and $Or_{10} = r_{upper_{10}} - r_{lower_{10}}$.

(WIv) Apply equation 3.3 to find r_{10}

[1] x_{Pi} and y_{Pi} $(i = 1, 2)$ can be obtained from a kinematic analysis of a Watt I six-bar mechanism; d_{min} and d_{max} are the minimum and maximum distances, respectively, between the pins at $P1$ and $P2$ throughout the motion cycle.

The above redefinitions for r_{lower_i} and r_{upper_i} ensure that the transformation of equation 3.3 will return a Watt I six-bar mechanism having $r_{long} = r_1$ of the four-bar base. Specifically,

- WIa, d, g and j ensure that $r_{short} = r_2$; WIb, e, h and k ensure that $r_{long} = r_1$;
- WIl ensures that $r_1 + r_2 \leq r_4 + r_3$, i.e., Grashof condition;
- WIr ensures that $r_{10} + r_9 > d_{max}$ and WIs ensures that $|r_9 - r_{10}| < d_{min}$;
- WIt ensures that $|r_{10} - r_9| < d_{min}$ and WIu ensures that $r^*_{upper_{10}} > r^*_{lower_{10}}$;
- WIc, f, i, n, p and v map the unconstrained $\mathbf{x}$ elements to $\mathbf{r}$ lengths corresponding to a constrained Watt I six-bar mechanism.

3.4.2 Transformation for Stephenson III Six-bar Mechanisms

The transformation from $\mathbf{x}$ to $\mathbf{r}$, based on redefined upper and lower link length limits, for a Stephenson III six-bar mechanism (Fig. 3.3) having the link 2 as the shortest link ($r_{short} = r_2$) and having the frame 1 as the longest link ($r_{long} = r_1$), is as follows:

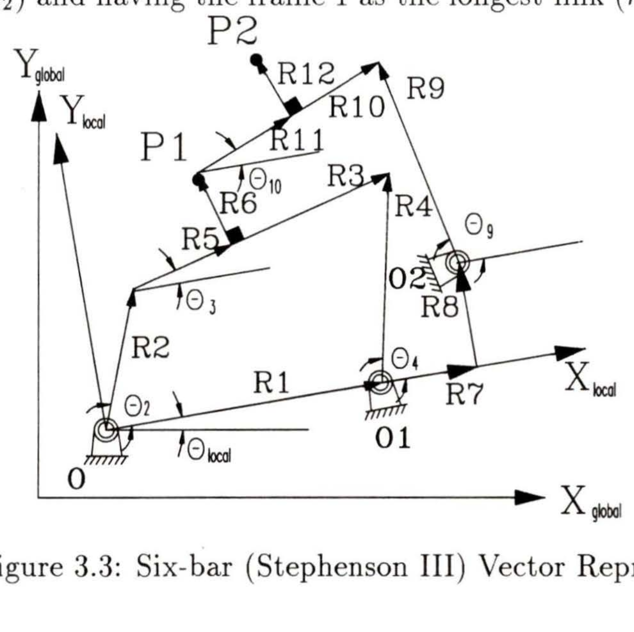

Figure 3.3: Six-bar (Stephenson III) Vector Representation

For $x_2 \longrightarrow r_2$:

(STa) $r^*_{lower_2} = r_{lower_2}$

(STb) $r^*_{upper_2} = r_{upper_2}$

(STc) Apply equation 3.3 to find r_2

For $x_1 \longrightarrow r_1$:

(STd) If: $r_2 > r_{lower_1}$; then $r^*_{lower_1} = r_2$; else: $r^*_{lower_1} = r_{lower_1}$

(STe) $r^*_{upper_1} = r_{upper_1}$

(STf) Apply equation 3.3 to find r_1

For $x_3 \longrightarrow r_3$:

(STg) If: $r_2 > r_{lower_3}$; then $r^*_{lower_3} = r_2$; else: $r^*_{lower_3} = r_{lower_3}$

(STh) If: $r_1 < r_{upper_3}$; then $r^*_{upper_3} = r_1$; else: $r^*_{upper_3} = r_{upper_3}$

(STi) Apply equation 3.3 to find r_3

For $x_4 \longrightarrow r_4$:

(STj) If: $r_2 > r_{lower_4}$; then $r^*_{lower_4} = r_2$; else: $r^*_{lower_4} = r_{lower_4}$

(STk) If: $r_1 < r_{upper_4}$; then $r^*_{upper_4} = r_1$; else: $r^*_{upper_4} = r_{upper_4}$

(STl) If: $r_{lower_4} + r_3 < r_1 + r_2$; then $r^*_{lower_4} = r_1 + r_2 - r_3$; else: $r^*_{lower_4} = r_{lower_4}$

(STm) If: $r^*_{upper_4} > r^*_{lower_4}$; then goto (STn); else: redefine the lower limits of $r^*_{lower_3}$ and $r^*_{lower_4}$ as follows (see Appendix B):

$$r^*_{lower_3} = \frac{Or_3(r_1+r_2-r^*_{upper_4})+Or_4 r^*_{upper_3}}{Or_3+Or_4}$$

$$r^*_{lower_4} = \frac{Or_4(r_1+r_2-r^*_{upper_3})+Or_3 r^*_{upper_4}}{Or_3+Or_4}$$

where $Or_3 = r_{upper_3} - r_{lower_3}$, and $Or_4 = r_{upper_4} - r_{lower_4}$

Apply equation 3.3 to find r_3

(STn) Apply equation 3.3 to find r_4

For $x_i \longrightarrow r_i$, i= 5 to 9, 11 & 12:

(STn) $r^*_{lower_i} = r_{lower_i}$, i= 5 to 9, 11 & 12

(STo) $r^*_{upper_i} = r_{upper_i}$, i= 5 to 9, 11 & 12

(STp) Apply equation 3.3 to find r_i

For $x_{10} \longrightarrow r_{10}$:

(STq) $d_{max} = max(\sqrt{(x_{P1} - x_{O2})^2 + (y_{P1} - y_{O2})^2})$;
$d_{min} = min(\sqrt{(x_{P1} - x_{O2})^2 + (y_{P1} - y_{O2})^2})$. [2]

(STr) If: $r_{lower_{10}} < |d_{max} - r_9|$; then $r^*_{lower_{10}} = |d_{max} - r_9|$;
else: $r^*_{lower_{10}} = r_{lower_{10}}$

(STs) If: $r^*_{lower_{10}} < |r_9 - d_{min}|$; then $r^*_{lower_{10}} = |r_9 - d_{min}|$;

(STt) If: $r_{upper_{10}} > (d_{min} + r_9)$; then $r^*_{upper_{10}} = (d_{min} + r_9)$; else: $r^*_{upper_{10}} = r_{upper_{10}}$

(STu) If: $r^*_{upper_{10}} > r^*_{lower_{10}}$; then goto (STv); else, redefine the lower and upper limits of $r^*_{lower_9}, r^*_{upper_9}, r^*_{lower_{10}}$ and $r^*_{upper_{10}}$ as shown in (WIu).

(STv) Apply equation 3.3 to find r_{10}

The above redefinitions for r_{lower_i} and r_{upper_i} ensure that the transformation of equation 3.3 will return a Stephenson III six-bar mechanism having $r_{long} = r_1$ of the four-bar base loop. Specifically,

- STa, d, g and j ensure that $r_{short} = r_2$; STb, e, h and k ensure that $r_{long} = r_1$;
- STl ensures that $r_1 + r_2 \leq r_4 + r_3$, i.e., Grashof condition;
- SIr ensures that $r_{10} + r_9 > d_{max}$ and SIs ensure that $|r_9 - r_{10}| < d_{min}$;
- SIt ensures that $|r_{10} - r_9| < d_{min}$ and SIu ensure that $r^*_{upper_{10}} > r^*_{lower_{10}}$;
- STc, f, i, n, p and v map the unconstrained $\mathbf{x}$ to $\mathbf{r}$ corresponding to a constrained Stephenson III six-bar mechanism.

[2] x_{P1}, y_{P1}, x_{O2} and y_{O2} can be obtained from kinematic analysis of Stephenson III six-bar mechanism; d_{min} and d_{max} is the minimum and maximum distance between the pins at P_1 and O_2 throughout the motion cycle.

3.5 Discussion

In this chapter, transforms were developed to allow mapping from an unconstrained n-dimensional search variable set $\mathbf{x}$ to n link parameter values $\mathbf{r}$ describing a constrained linkage within specific mechanism sub-types. The transforms were based on redefining link length limits, incorporating "Grashof-criteria" considerations, concurrently with the transformation of the unconstrained variable set values. The technique facilitates effective optimization allowing searching and convergence within a preferred mechanism sub-type.

The four-bar transform guarantees convergence within user-specified limits if conditions on the relative magnitudes of the specified link length ranges are satisfied (see Appendix B).

The presented six-bar transform does not guarantee convergence within user-specified limits and it is possible that the redefined length limits of link 9 and 10 may go beyond the original length limits of the link. It is recommended that an avenue of future research be the investigation of further constraints to completely ensure user-specified limits for six-bar mechanisms (see Section 5.2).

Transforms were presented for Grashof four-bar (crank-rockers, drag-links and double-rockers) mechanisms and for six-bar (Watt I and Stephenson III) mechanisms. The technique can be applied to other type of mechanisms such as geared five-bar mechanisms. In future research, it may be of interest to attempt to develop similar transformations for spatial mechanism sub-types. The developed transforms are demonstrated to be effective through application in the example optimizations of Chapter 2 and in the case study of Chapter 4.

Chapter 4

Case Study - A Deep Digging Implement

4.1 Case Study Problem

4.1.1 Background

In recent years there has been a growing awareness of the damage done to soil structures by modern farming practices. The trend towards larger and heavier agricultural machinery exacerbates the problem of soil compaction, which impairs both plant root growth and soil water movement.

There are also more and more concerns on increased cost and reduced availability of fossil fuels, on improved human nutrition, and on environment degradation. These concerns have encouraged agronomists to re-evaluate our current conventional farming practice and have caused many scientists and farmers to search for an alternative crop-production system. As a result, organic farming is offered as an alternative for conventional agriculture.

In organic farming and in particular with organic vegetable growing, it is important to optimize the soil condition in order to maximize crop yields. Important aspects of soil condition are good aeration and drainage, and adequate fertility. To obtain good yields with vegetables, these conditions should exist to a soil depth of about 20 inches. Preparing the soil by deep-digging or double-digging has been a major aspect of the bio-intensive gardening technique brought to the US by Alan Chadwick. The research on this technique by Jeavons (1982) [20] was shown that loosening the soil to a maximum of 24 inches has increased yields by a factor of 2 to 30 times. The reason for this boost is the increased air space, which allows for an increase in the diffusion of oxygen (which the roots and microbes depend on) into the soil, and the diffusion of carbon dioxide (which the leaves depend on) out of the soil. Generally speaking, the required soil conditions can be achieved by loosening the soil to that depth and subsequently working the top soil to form a bed for seeds or transplants.

In order to achieve the soil condition mentioned above, a deep-loosening implement needs to be developed to thoroughly loosen the soil. There have been some improvements to sub-soiling implements in recent years, especially the development of a spading machine in Europe which uses a basic four-bar mechanism. This device uses paddles to till the top 6 to 8 inches of soil, but very aggressively mixes soil horizons at deeper depth. Experience has indicated that such a machine in improper moisture conditions can cause severe damage to the structure of the soil. Another implement is the Para-plow (hard-pan breaker) that uses unique angled legs mounted on a horizontal shaft to break up the soil compaction and pans. Studies under field conditions show it to be very effective at disturbing the hardpan layer. The effect of this form of sub-soiling diminishes with time and must be repeated every few years depending upon soil type and condition. In addition, the Para-plow requires a minimum power input of 35 to 40 hp per shank, which are placed 26 inches apart.

One of the organic farmers at Fraser Valley in B.C. has experimented with a modified Bobcat having forks attached to the bucket to allow deep-digging with good result. No controlled testing has yet been undertaken. The Bobcat was observed to be overly aggressive in loosening the soil and mixing of the topsoil and subsoil did occur.

In addition, the present practice of ploughing and rotor tilling is deficient in two ways - the deep soil is not loosened to improve aeration and drainage, and topsoil is treated too vigorously by rotor tilling (killing worms, etc., and pulverising the soil). Presently, there is no implement on the market that can effectively loosen soil to the depth of 20 inches. Therefore, such an implement needs to be developed.

4.1.2 Task Specification

The design objective is to develop a deep-loosening implement that can loosen soil to a depth of about 20 inches, without mixing of topsoil and subsoil, for improved aeration, hair root growth, water infiltration, and nutrient utilization.

The developed implement will be a fork-based mechanism. It will have a series of long forks, spanning the width of the bed or that of a tractor, approximately separated by 6 inches. During the digging action, each fork would pierce the soil at a penetration angle of β and would leave at a withdrawal angle of γ as shown in Figure 4.1. The forks should move up and down driven by means of a continuously-rotating crank-driven mechanism.

After each fork leaves the ground, it moves forward a "bite" before entering the soil again. The mechanism that generates this motion of the fork will be a mechanism which is mounted on a frame supported by a three-point hitch. The mechanism

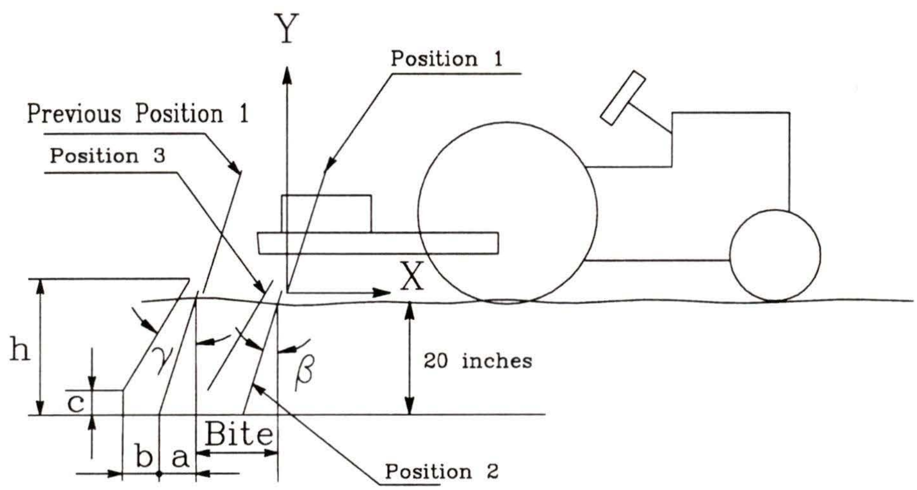

Figure 4.1: Required Motion w.r.t. the Ground

will be powered from the power-take-off(PTO) shaft on the tractor. Working speed requirements are estimated to be 20 to 30 feet per minute.

The task specification above can summarized as shown in Table 4.1. The motion of the implement with respect to the moving tractor is illustrated in the Fig. 4.2.

4.2 Problem Definition

4.2.1 Design Objective

The design objective is to develop a deep-loosening implement that can loosen soil to a depth of about 20 inches without mixing of topsoil and subsoil. The significance of the objective has been discussed in Section 4.1.1.

Parameter	Parameter Description	Range of Value	Value Selected
h	required digging depth (20 inches) plus clearance (2 inches)	22 inches	22 inches(55 cm)
β	penetrating angle	15 °	15 °
γ	withdrawing angle	30 °	30 °
Bite	horizontal distance moved between successive fork entries	4 - 8 inches	6 inches (15.24 cm)
b		3 - 5 inches	4 inches (10 cm)
c		3 - 4 inches	3 inches (7.6 cm)
v_{tr}	tractor speed	0.4 mph	0.4 mph(0.6 km/hr.)
n	Crank rotation speed	60 rpm	60 rpm (1 cycle/sec.)

Table 4.1: Summary of Deep Digging Task Specification

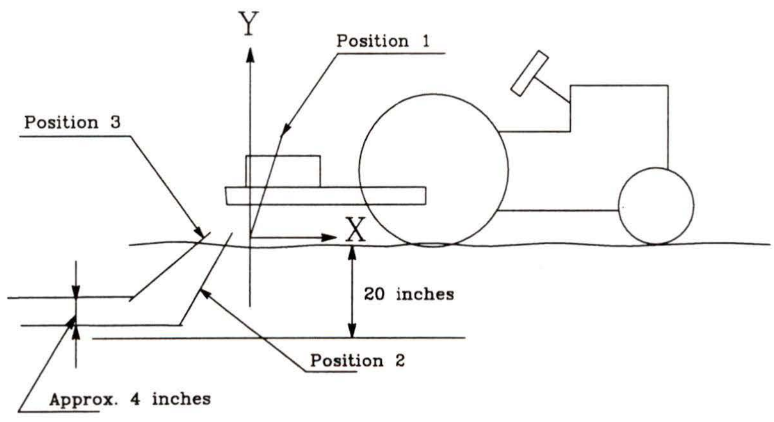

Figure 4.2: Relative Motion w.r.t. the Moving Tractor

4.2.2 Motion Constraints

From the task specification described in the previous section, the design problem can be defined as a planar motion generation problem, i.e., a rigid body is guided through a series of required positions in a plane. Specifically, the problem can be described in terms of three parameters, the coupler point location x and y, and angle θ_3 as a function of the input crank angle θ_2.

It is convenient to specify the mechanism motions relative to the moving tractor. Therefore, the fork would act as shown in Fig. 4.2. The required changes of parameters for coupler point Δx, Δy and angle $\Delta\theta_3$ can be specified as a function of the change of input crank angle $\Delta\theta_2$ as shown in Table 4.2 and corresponding motion constraints are illustrated in Figs. 4.3 to 4.5.

As specified in Table 4.2, the fork performs the deep dig (position 1 to 2) and the soil is pushed back and slightly lifted (position 2 to 3) while the crank rotates 120° and 100°, respectively. Finally, the fork is withdrawn and is moved to next starting position with respect to the moving tractor as the crank completes a full cycle.

Compensation for the tractor movement is considered when fork positions are specified and the critical fork positions correspond to the changes of angle $\Delta\theta_2 =$ 120°, 220°, and 330°, and the achieval of particular changes in Δx displacements

$\Delta\theta_2$ (deg.)	Δx (cm)	Δy (cm)	$\Delta\theta_3$ (deg.)
0	0	0	0
120	-18.8	-55	0
220	-33.4	-47.4	-15
330	-16.6	-5	-5
360	0	0	0

Table 4.2: Relative Deep Digging Task Specification

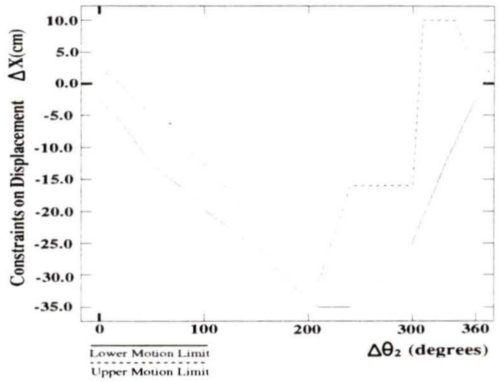

Figure 4.3: Task Specification – Displacement ΔX

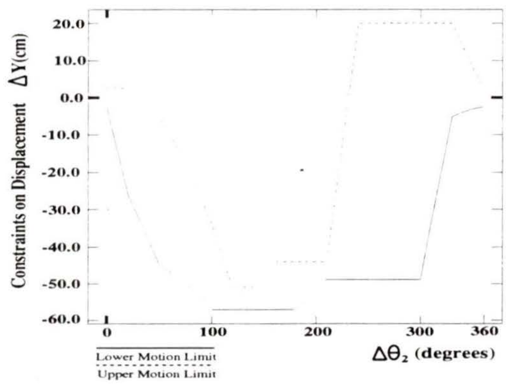

Figure 4.4: Task Specification – Displacement ΔY

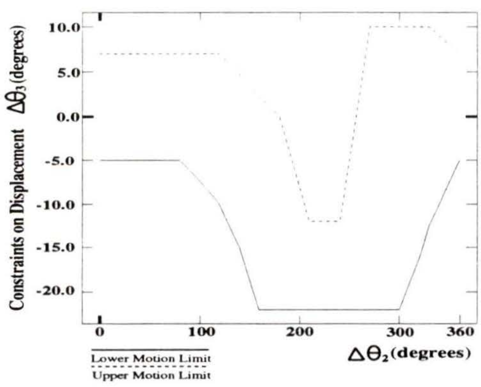

Figure 4.5: Task Specification – Displacement $\Delta\theta_3$

during the ground action (0° to 220°).

4.2.3 Other Design Constraints

1) The fork must dig up to a depth of 20 inches;

2) The fork should be lifted a distance of 3 to 4 inches when it turns and lifts the soil as shown in Figure 4.2;

3) The fork should not press the undug soil much during both the digging action and the return;

4) The mechanism is limited to be planar and to have one degree of freedom, all mechanism obeying the general degree-of-freedom equation;

5) The mechanism must be able to generate the desired output motion and satisfy the functional requirements;

6) The bit should be adjustable;

7) Overload protection is needed to accommodate rocks and small stones to prevent breaking of the forks or mechanisms.

4.3 Type Synthesis

4.3.1 Discussion of the Potential Types of Mechanisms

One of the most important stages in mechanical design is the conceptual phase, i.e., the creation of a mechanism to satisfy a desired functional requirement. There is no general approach to solve the type synthesis problem at the present time. Traditionally, it is accomplished by the designer's intuition and experience and/or with the aid of mechanism handbooks that include various kinds of mechanisms available now as well as some type synthesis approaches such as associated linkage method [10].

There are three basic mechanisms types, i.e. gear chain, cam-and-follower and linkage-based mechanisms. Some of them and various combinations of them with other mechanical and non-mechanical components, such as linkage mechanisms, geared five-bar linkages (GFBL), and cam-modulated linkages (CML), could be available for accomplishing the required task. Table 4.3 summarizes their advantages and disadvantages.

Four bar linkages are the simplest closed-loop linkages. They consist of three moving links (plus one fixed link) and four revolute joints. The link that is connected to the power source is called the input link. There are various kinds of four-bar mechanism that can be used to perform tasks of path generation, function generation or motion generation (rigid-body guidance). In motion generation, the entire motion of the coupler link is of concern.

If a four-bar linkage does not provide the performance required for a particular application, a six-bar mechanism is usually considered next. A six-bar linkage is a mechanism with two closed-loops and 7 revolute joints. Six-bar linkages can perform more complicated task requirements than four-bar mechanisms.

A GFBL provides another option when a desired motion or function is too complex to produce with a four-bar linkage. A GFBL is basically a five-bar linkage with two input angles related by a linear gearing constraint identified by the gear ratio and phase angle. With a geared ratio between its two cranks, a two-d.o.f. five-bar linkage becomes a single d.o.f. geared mechanism. It can be designed to fulfill more complicated motion requirements than a four-bar mechanism.

A CML is a mechanism consisting of one or more cam-and-follower pairs in combination with a linkage. Required input can be provided by the cam and specified

Option	Advantages	Drawbacks
Four-bar linkage	Simplest single-loop topology; Economical manufacture; Well suited to harsh vibration; Robustness of pin joints; Good maintainability	Limitation in capability for complicated function requirements.
Six-bar linkage	Same as above except for the manufacturing costs and more complex two loop topology; capable of accomplishing more complicated tasks than four-bar linkages	Increase in expense, weight, and complicity compared to four-bar linkages.
Geared five-bar linkage	Capable of accomplishing more complicated tasks than four-bar linkages; Simple single-loop topology.	Increase in expense and weight. Gears may require protection from the operating environment.
Cam-modulated linkage	Capable of providing most precise motion; easy to design.	Costly to manufacture; Poorly suited to harsh vibrational environments. Cams may require protection from the operating environment.

Table 4.3: Comparison of Type Options

output can be generated by the linkage. The primary difference between a CML and a cam-and-follower mechanism is that a CML is capable of producing function, path and coplanar motion outputs, whereas a cam-and-follower mechanism can in general produce only functional outputs. While a CML may be considered a more complicated device than a linkage mechanism because of the presence of at least one higher-pair connection, it provides for precise control over position and velocity of the output link throughout the motion cycle, including dwell during any part of the output cycle.

From Table 4.3, it is known that a linkage system has several advantages over a cam-modulated linkage for the digger application. Specifically, a linkage has better wear properties and lower manufacturing costs. Wear due to a dusty environment and higher cost are serious disadvantages of CML. Therefore, linkages are desirable choice for type synthesis. In the following examples, four-bar and six-bar mechanisms will be considered. In addition to the linkage system, a GFBL also shows some good properties, although a GFBL may have potential high cost due to the requirement of the gear system.

4.3.2 Type Synthesis

Type synthesis is the process of determining possible mechanism structures to perform a given task or combination of tasks without regard to the dimension of the components. Separating structure and function in type synthesis, as shown by Freudenstein and Maki[1979] [13], decomposes the type synthesis process into "topological synthesis" and "topological analysis".

Topological synthesis of the linkage system

Topological synthesis, the process of systematically enumerating the distinct "basic mechanisms" based strictly on the topological requirements, is divided into the enumeration of the Basic Kinematic Chains(BKCs) satisfying the topological requirements and the enumeration of distinct basic mechanisms.

Enumeration of the BKCs: Enumeration of BKCs starts with the solution of the Chebyshev-Greubler equation with full closure of the kinematic chain. The solution determines groups for four-bar linkages to have four binary links and for six-bar linkages to have four binary and two ternary links. The enumeration of the BKCs was performed (see Fig. 4.6 for results).

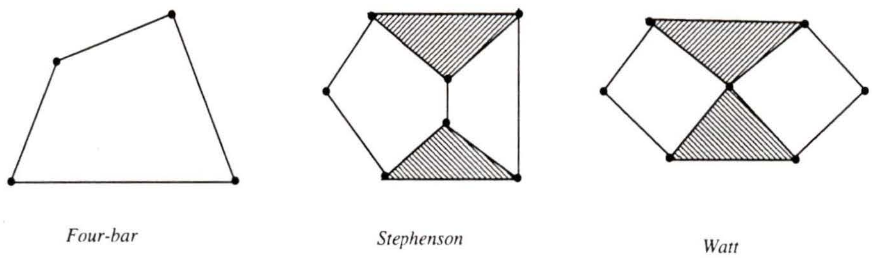

Figure 4.6: Basic Kinematic Chains (BKCs)

Enumeration of the basic mechanisms: The BKCs are transformed into basic mechanisms by assigning a ground link. Several topologically unique inversions of a given BKC can be derived as shown in Fig. 4.7 for six-bar mechanisms.

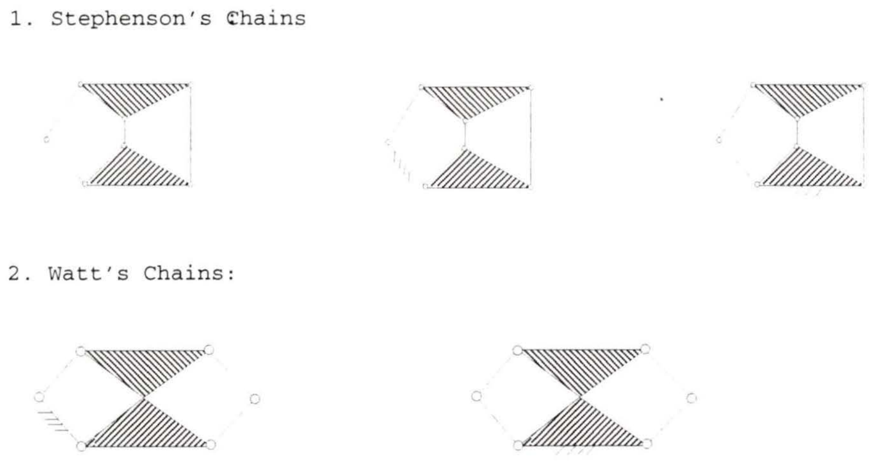

Figure 4.7: Basic Unique Mechanisms

For the six-bar mechanism, there are two BKCs satisfying the topological requirements defined before. These are referred to as a Stephenson six-bar chain if the tertiary links are separated by binary links and as a Watt six-bar chain if the tertiary links are adjacent. Five types of the six-bar linkage with a single degree-of-freedom, i.e., variations of the Stephenson and Watt chains shown in Fig. 4.7, can be derived.

For the four-bar mechanisms, there is only one Basic Kinematic Chain(BKC) satisfying the topological requirements, i.e. one degree of freedom and planar motion. Several basic mechanisms can be derived, such as crank-rocker, rocker-crank, drag-link, Grashof rocker-rocker, non-Grashof, changing point and slider-crank mechanisms. For the task specification defined before, only crank-rocker mechanisms are suitable.

Topological analysis of the linkage system

Once the unique basic mechanisms are enumerated through topological synthesis, it is necessary to analyze them with respect to their functional requirements after determining feasible combinations of input, output and joint types.

Based on the usable six-bar kinematic chains obtained and the design specification or constraints defined before, many variations of the six-bar mechanism can be generated as shown in Fig. 4.8. In Fig. 4.8 F, I and O represent the frame link, input link and output link, respectively.

Mechanism analysis: Evaluating the mechanism type based on the functional requirements is viewed as the most important step in the type synthesis. Essentially, the mechanisms are evaluated to determine if they could generate the desired output motion and if it is theoretically possible for them to satisfy the functional requirements. The evaluation results have shown that all mechanisms in Fig. 4.8 are potentially acceptable. Some desired potential six-bar basic mechanisms are shown in Fig. 4.9. The cases illustrated in Fig. 4.9 correspond to respecialization cases that could practically be utilized for the digging task.

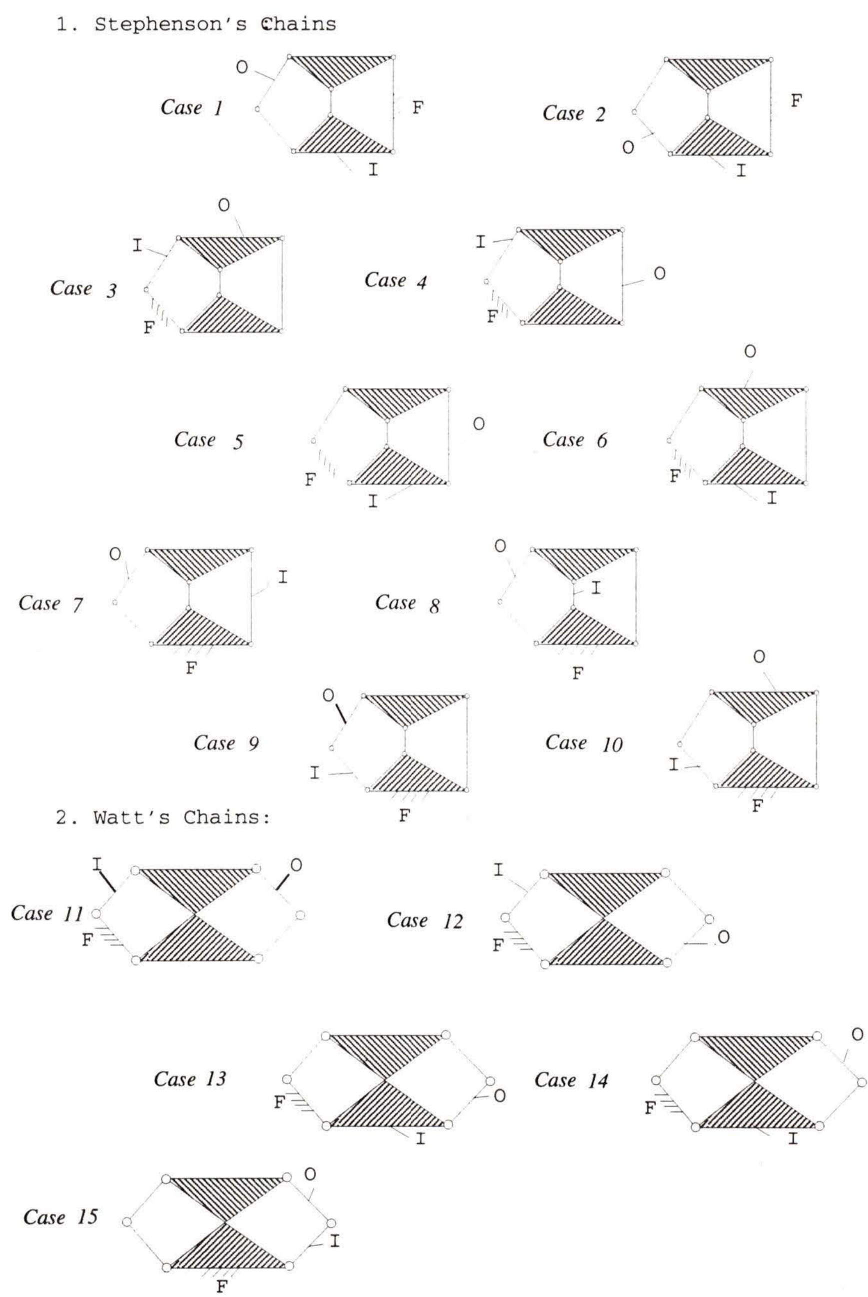

Figure 4.8: Respecialization of the Six-bar Mechanism

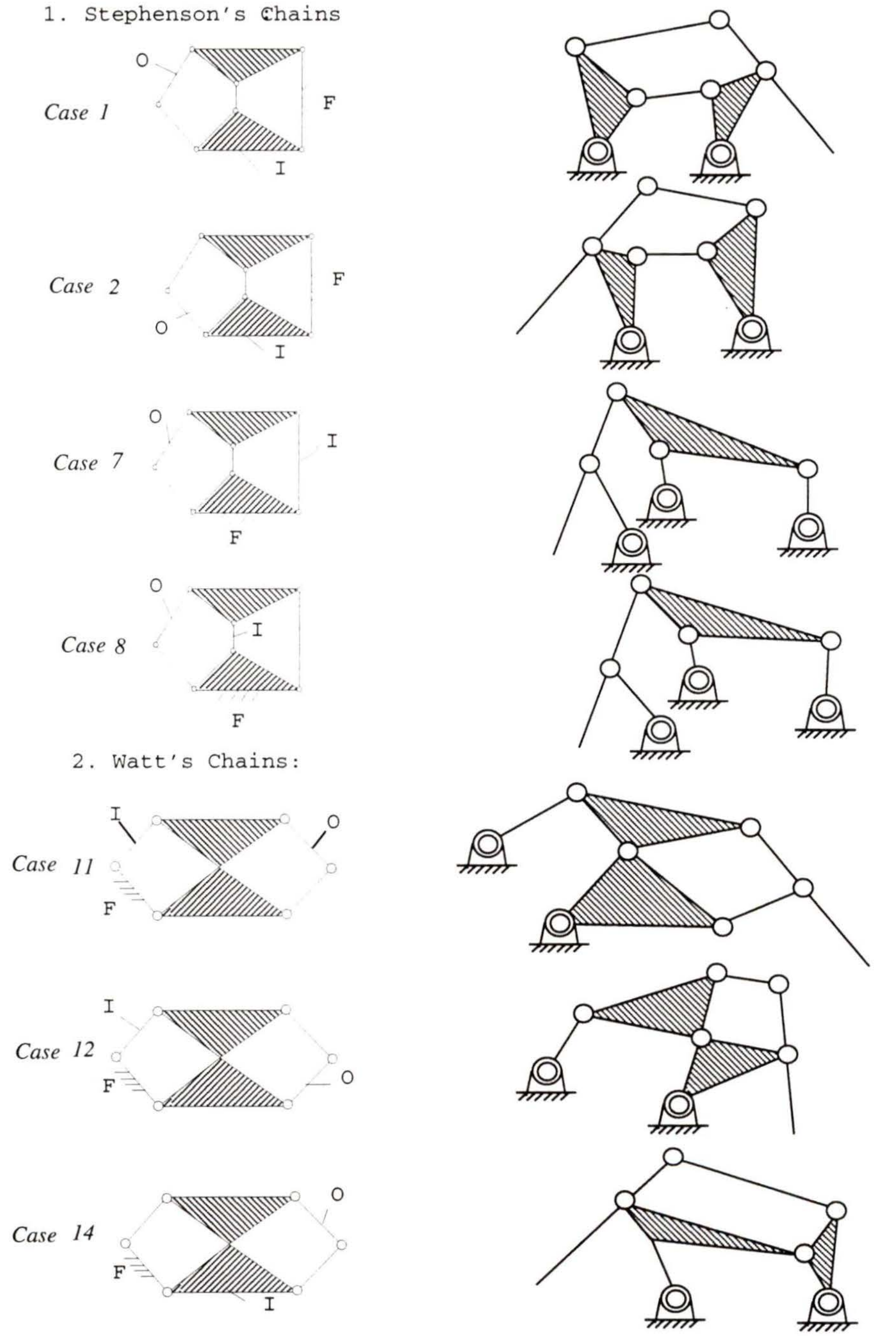

Figure 4.9: Derivation of the Six-bar Mechanisms

4.4 Dimensional Synthesis of Four-bar Mechanism

A crank-rocker four-bar mechanism having the base as the longest link was considered first to guide the digging tool through the required motion shown schematically in Fig. 4.1. This required task was specified in Section 4.1.2. Constraints on the lengths of the links of $\mathbf{r}_{upper} = \{185, 100, 165, 170, 230, 0\}^T$ cm and $\mathbf{r}_{lower} = \{25, 10, 25, 25, 5, -100\}^T$ cm exist. The initial value of the displacement of the crank (θ_{2_0}) is unconstrained (0-360°) and the values of θ_{local} are constrained to ±90°.

The task was modeled with upper and lower displacement component (task) constraints as illustrated in Figs. 4.3 to 4.5. An initial mechanism $\mathbf{r}_{initial} = \{100, 30, 90, 45, 30, -30\}^T$ cm with $\theta_{2_0} = 90°$ and $\theta_{local} = 0°$ resulted in the motion indicated with a solid line on the graphs of Figs. 4.10 to 4.12. A least-pth search (as discussed in Section 2.6.2) combined with the sub-type specific transforms (as discussed in Chapter 3) rendered a feasible crank-rocker with $\mathbf{r}_{feasible} = \{178.0, 23.8, 148.7, 55.8, 208.9, -12.8\}^T$ cm with $\theta_{2_0} = 188.7°$ and $\theta_{local} = -55.2°$ The displacement output for this feasible mechanism is indicated by the dashed lines on Figs. 4.10 to 4.12. Least-pth searches for different initial mechanisms were performed and their results are included in Table 4.4. From the results in Table 4.4 it can be observed that convergence to basically three sets of link lengths has occurred.

From the feasible mechanisms generated, it was found that lengths of linkage are very long, i.e., the four-bar is not very compact. The lengths are related to the sizes of the whole structure and to the inertia of the mechanism. Therefore, the objective of compactness was considered through the minimization of the length of the first member, i.e.,

$$f_{obj_1} = r_1. \tag{4.1}$$

The reason for minimizing the length of base link r_1 is that it is the longest link of

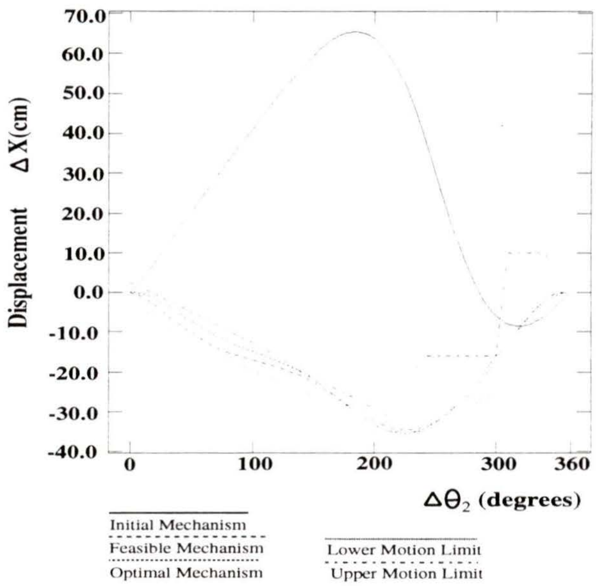

Figure 4.10: Task Specification and Resulting Mechanism Motions – ΔX

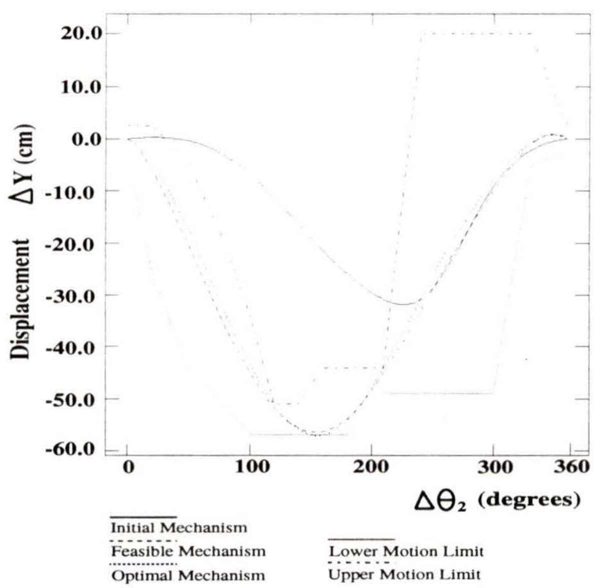

Figure 4.11: Task Specification and Resulting Mechanism Motions – ΔY

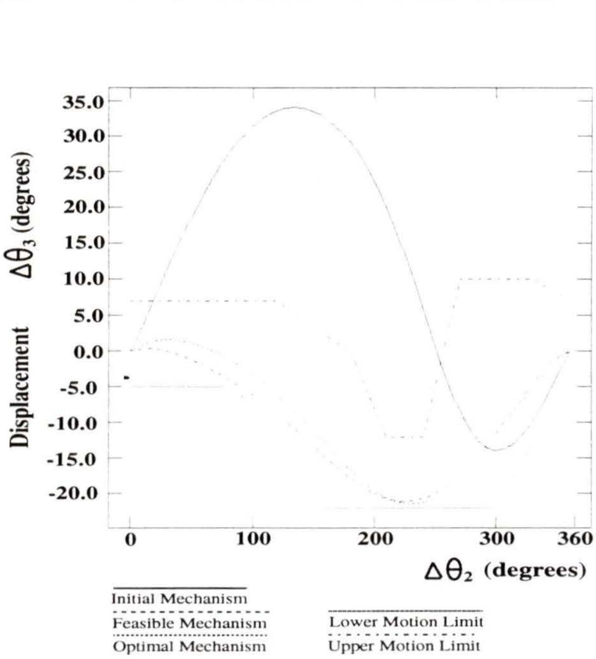

Figure 4.12: Task Specification and Resulting Mechanism Motions – $\Delta\theta_3$

this four-bar sub-type and its minimization increases the compactness of the mechanism.

The SUMT discussed in Section 2.6.3, in combination with the sub-type specific transformation method, was applied, resolving the mechanism as reported in the Obj.1 row of Table 4.5. The results from the search process are shown in Fig. 4.13. From the Table 4.5, the optimal mechanism has an r_1 length of 160.1 cm, which is approximately 90 percent of the r_1 length of 178.0 cm of the original feasible mechanism.

Simulation results of the transmission angles for the feasible and f_{obj1} optimal mechanism are shown in Figure 4.14. Since the designed mechanism will be used to perform deep-digging action, the transmission angle is of concern. Simulation of the f_{obj1} optimal mechanism found it to have a minimum transmission angle $\mu_{min} = 21.1°$ at $\Delta\theta_2 = 180°$ as shown by solid line on Fig. 4.14. This transmission angle is unacceptable. Therefore, the objective of maximizing the minimum of the transmission

ini. mechanism $\mathbf{r}_{initial}$(cm)	f_{1st} value	feasible mechanism $\mathbf{r}_{feasible}$(cm)	No. of iterations	f_{final} value
100,30,90,45, 30,-30,90,0	138.36	178.0,23.8,148.7,55.8, 208.9,-12.8,188.7,-55.2	2121	-0.206
100,30,90,45, 60,-60,60,0	149.989	181.8,23.4,149.4,58.7, 211.9,-15.2,188.2, -53.1	1532	-0.23
80,25,65,75, 30,-30,90,0	90.98	185.0,23.2,149.5,61.8, 212.1,-18.2,187.2,-51.2	1534	-0.24
80,25,65,75, 120,-20,100,0	112.404	168.8,21.4,135.5,57.5, 209.6,-7.7,189.5,-48.9	1635	-0.143
80,15,70,60, 120,-20,100,0	116.752	182.4,23.8,151.6,57.6, 212.5,-14.9,188.1,-54.7	2140	-0.222
80,35,60,60, 120,-20,100,0	135.525	166.3,23.3,138.0,55.0, 203.4,0.0,189.1,-56.6	1578	-0.105
100,30,80,90, 120,-20,100,0	124.457	185.0,23.1,149.5,61.9, 212.2,-18.2,187.2,-51.1	2033	-0.241
100,30,90,45, 120,-20,100,0	160.508	185.0,23.2,149.5,61.9, 212.0,-17.1,187.6,-52.3	2268	-0.241
100,30,45,90, 120,-20,100,0	174.168	185.0,23.2,149.5,61.8, 212.1,-18.2,187.2,-51.2	2225	-0.241
80,25,65,75, 60,-60,60,0	141.805	185.0,23.2,149.5,61.9, 212.0,-18.3,187.1,-51.2	1483	-0.241

Table 4.4: Least pth Search Results for the Four-bar Deep Digging Implement

Objective	f_{obj} value before opt.	f_{obj} value after opt.	No. of iterations	Optimized mechanism (cm)
Obj. 1	178.0 cm	160.1 cm	1237	160.1,20.4,128.6,54.6, 208.7,-4.3,190.5,-47.5
Obj. 2	21.1 degrees	36.8 degrees	1320	178.0,25.8,136.1,77.9, 199.1,-8.0,179.6,-53.9
Comb. Obj.	644	571.4	1746	168.5,25.3,135.0,67.0, 201.6,-2.3,184.5,-56.7
Obj. 1	178.0(534.0) cm	168.5(505.4) cm		
Obj. 2	21.1(110.0) degrees	34.7(65.9) degrees		

Table 4.5: Optimization Results for the Four-bar Deep Digging Mechanism

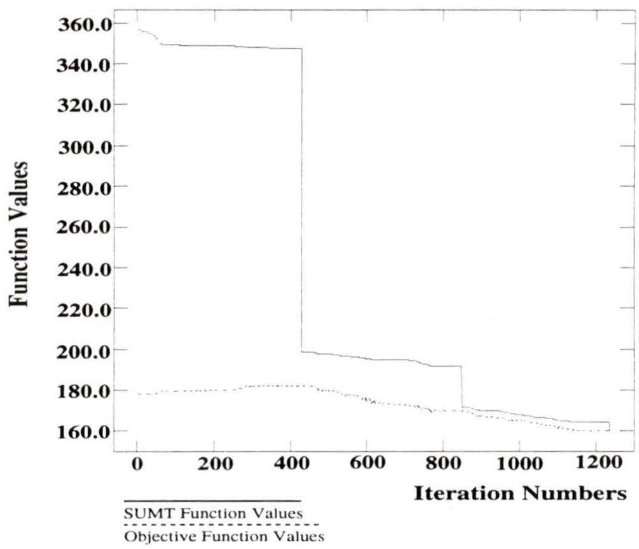

Figure 4.13: The Optimization Results ($f_{obj_1} = r_1$)

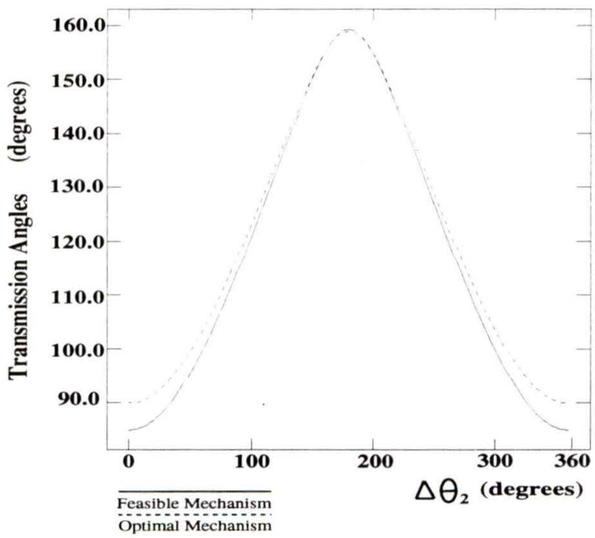

Figure 4.14: Transmission Angle - Feasible and f_{obj_1} Optimal Mechanisms

angle during the digging and withdrawing portion of the task was considered. This objective is equivalent to minimizing the inverse of the minimum of the transmission angle, i.e.,

$$f_{obj_2} = 1/\min\{\mu_{min1}, \mu_{min2}\} \tag{4.2}$$

where $\mu_{min1} = \mu_0$ and $\mu_{min2} = \mu_{180}$ are calculated from equations (2.14).

The SUMT was applied again, resolving the mechanism reported in the Obj. 2 row of Table 4.5. The results of the searching process are shown in Figure 4.15. Simulation results of transmission angles for the f_{obj_1} and f_{obj_2} optimal mechanism are depicted in Figure 4.16. The f_{obj_2} optimal mechanism has a minimum transmission angle of 36.8°, a 74 percent improvement of the minimum transmission angle of the f_{obj_1} optimal mechanism. However, the length of link 1 for the f_{obj_2} mechanism is 178.0 cm, i.e., an increase of approximately 11 percent of the length of the f_{obj_1} optimal mechanism. This result implies the sacrifice of the mechanism compactness for transmission angle improvement and is not satisfactory. Therefore, a combined weighted objective was considered with the goal of having a relatively compact mechanism with acceptable transmission angles. This weighted objective is

$$f_{obj_{combined}} = w_1 f_{obj_1} + w_2 f_{obj_2} \tag{4.3}$$

The optimization results for the combined weighted objective of equation (4.3) are shown in Table 4.5 and have about a 11 percent of improvement in terms of the weighted function. Specifically, the $f_{obj_{combined}}$ optimal mechanism's length of $r_1 =$ 168.5cm has been reduced to 94.7 percent of the original length. Moreover, the minimum transmission angle of the optimal mechanism has increased about 64 percent to 34.7°. The transmission angle values for the $f_{obj_{combined}}$ optimized mechanism are indicated by the solid line on Fig. 4.17. This results in a relatively compact mechanism and increased minimum transmission angle.

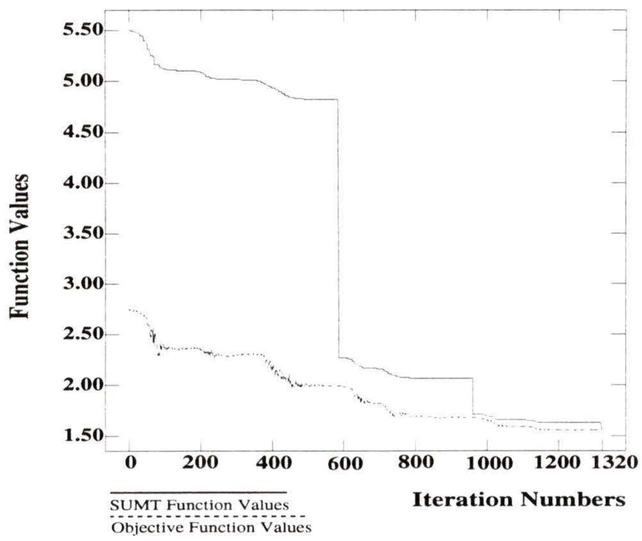

Figure 4.15: Optimization Results ($f_{obj_2} = 1/\min\{\mu_{min1}, \mu_{min2}\}$)

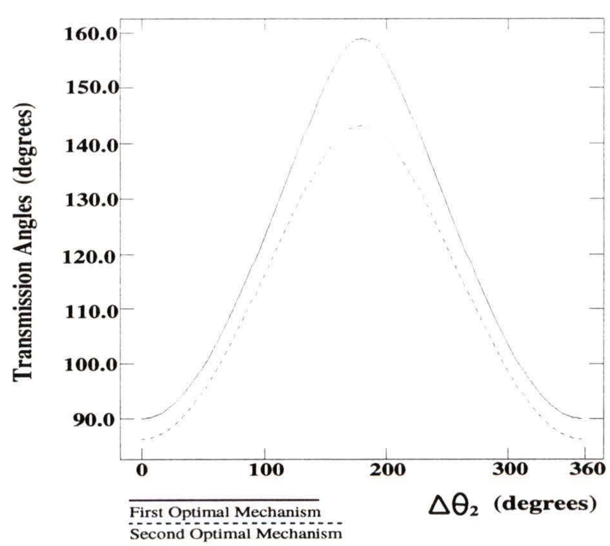

Figure 4.16: Transmission Angles - f_{obj_1} & f_{obj_2} Optimal Mechanisms

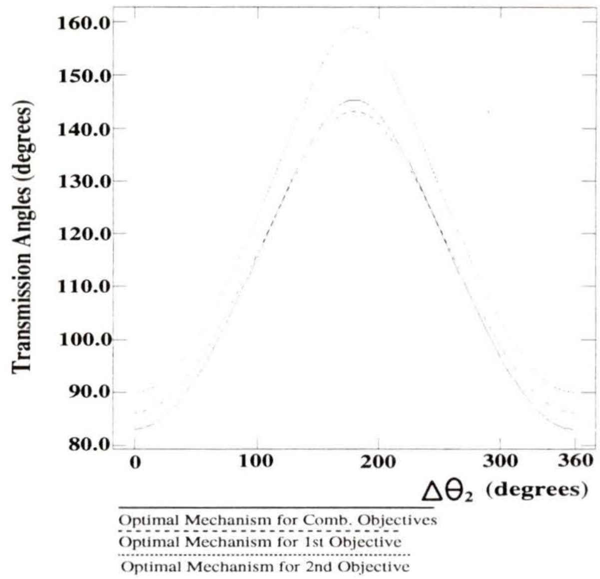

Figure 4.17: Transmission Angles - $f_{obj_{combined}}$, f_{obj_1} & f_{obj_2} Optimal Mechanisms

The displacements for the "combined objective optimum" mechanism are shown in the graphs of Figs. 4.10 to 4.12 as dotted lines and they can be seen to have moved right up to some of the constraint limits. This demonstrates that the motion constraints were the limiting factors in the searches. The search for the optimum mechanism required 1746 iterations for the combined objective, and resulted in the f_{SUMT} and f_{obj} values during searching as illustrated in Fig. 4.18. The final mechanism, coupler curve and follower arc of the deep-digging mechanism are depicted in Figure 4.19. Notice that the follower/coupler pin locations are above the ground level and the dig motion is as required.

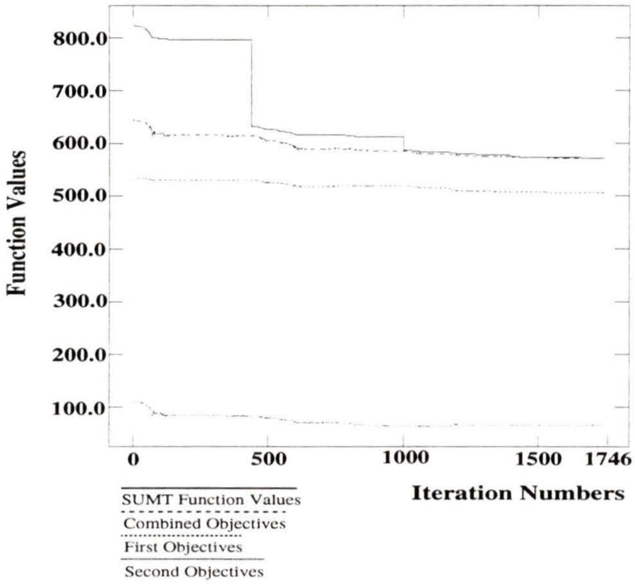

Figure 4.18: Optimization Results ($f_{obj_{combined}} = w_1 f_{obj_1} + w_2 f_{obj_2}$)

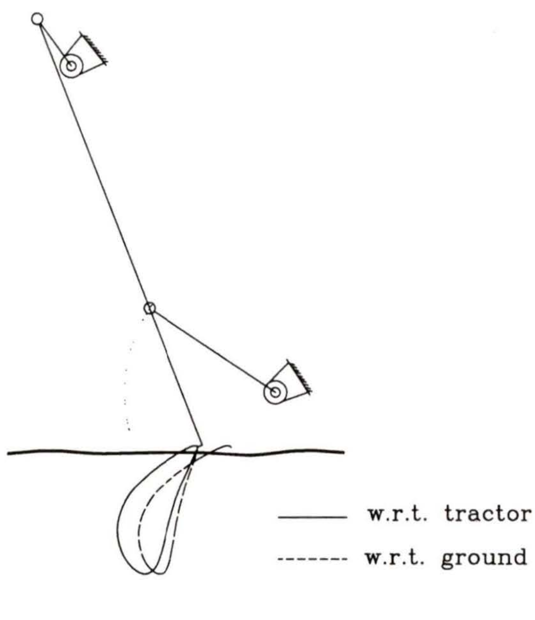

Figure 4.19: Four-bar Deep Digging Mechanism and its Coupler Curve

4.5 Dimensional Synthesis of Six-bar Mechanism

From the dimensional synthesis of the four-bar mechanism, a four-bar crank-rocker was generated to complete the required task. It has good transmission angle characteristics, but its overall size seems to be large and this will increase the cost of the product and the inertia of the mechanism. Hence, an alternative type of mechanism should be investigated. A six-bar mechanism with small individual member lengths has the potential of providing as complex or more complex motions as a four-bar.

A Watt I six-bar mechanism is considered to guide a digging tool through the positions shown schematically in Fig. 4.1. Modelling and parameters of the mechanism are discussed in Appendix A. The required task was specified in Section 4.1.2. Constraints on the lengths of the links of $\mathbf{r}_{upper} = \{170, 100, 150, 155, 150, 0, 100, 0, 120, 120, 180, 90\}^T$ cm and $\mathbf{r}_{lower} = \{25, 10, 25, 25, 5, -100, 5, -100, 5, 5, 5, -90\}^T$ cm are imposed. The initial value of the displacement of the crank (θ_{2_0}) is unconstrained (0-360°) and the values of θ_{local} are constrained to ±90°.

The task was modeled with upper and lower displacement component (task) constraints as illustrated in Figs. 4.20 to 4.22. An initial mechanism $\mathbf{r}_{initial} = \{120, 25, 65, 105, 40, -20, 30, -10, 50, 80, 150, -15\}^T$ cm with $\theta_{2_0} = 150°$ and $\theta_{local} = 0°$ resulted in the motion indicated with a solid line on the graphs of Figs. 4.20 to 4.22. A least-pth search (as discussed in Section 2.6.2) combined with the sub-type specific transforms (as discussed in Chapter 3) rendered a feasible crank-rocker with $\mathbf{r}_{feasible} = \{169.9, 19.6, 147.3, 99.6, 39.4, -91.0, 6.7, -21.4, 115.0, 114.7, 178.1, -22.6\}^T$ cm with $\theta_{2_0} = 177.9°$ and $\theta_{local} = -55.7°$. The displacement output for this feasible mechanism is indicated by the dashed lines on Figs. 4.20 to 4.22. Least-pth searches for different initial mechanisms were performed and their results are included in Table 4.6.

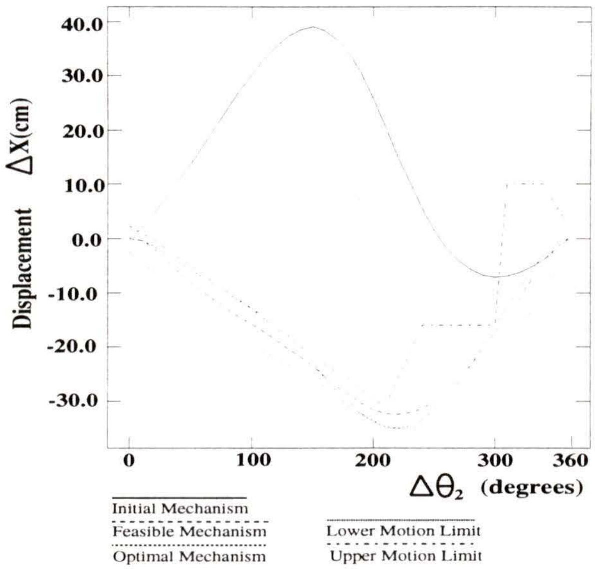

Figure 4.20: Task Specification and Resulting Mechanism Motions – ΔX

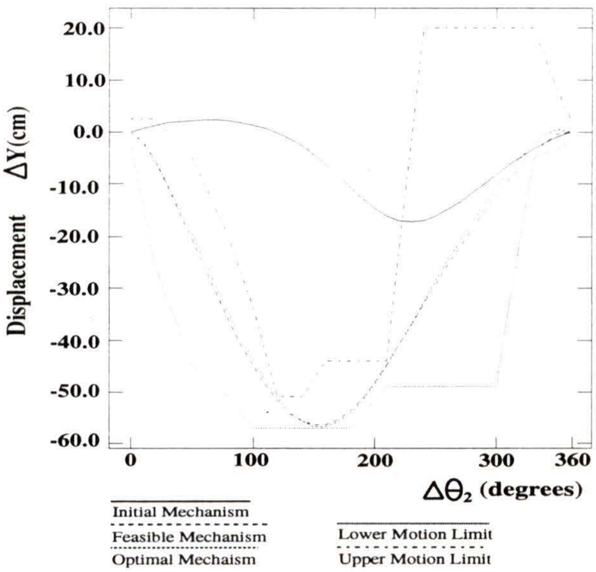

Figure 4.21: Task Specification and Resulting Mechanism Motions – ΔY

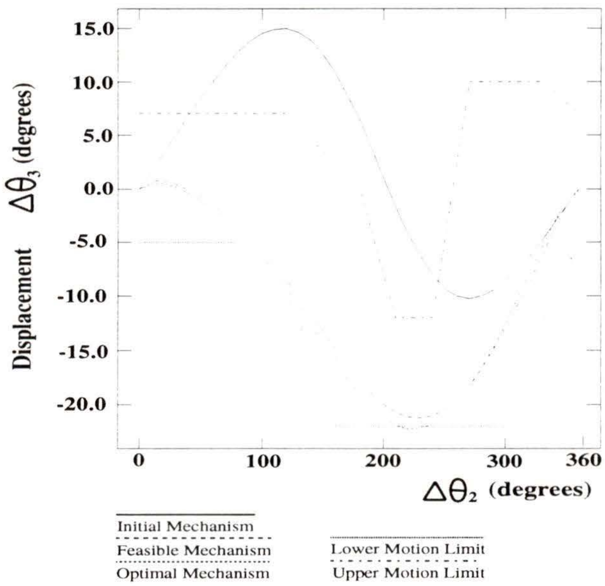

Figure 4.22: Task Specification and Resulting Mechanism Motions – $\Delta\theta_3$

ini. mechanism $\mathbf{r}_{initial}$(cm)	f_{1st} value	feasible mechanism $\mathbf{r}_{feasible}$(cm)	No. of iterations	f_{final} value
120,25,115,75,40, -20,30,10,50,80, 150,-15,150,0	89.98	170.0,19.5,150.0,155.0,42.9, -44.7,5.0,0.0,89.8,115.9, 179.4,-21.0,179.1,-49.5	3642	-0.1504
120,25,65,105,40, -20,30,-10,50,80, 150,-15,150,0	80.62	169.9,19.6,147.3,99.6,39.4, -91.0,6.7,-21.4,115.0,115.0, 178.1,-22.6,177.9,-55.7	979	-0.24
80,35,65,75,30, -30,30,-30,50,80 150,-15,90,0	249.656	170.0,20.6,47.1,143.5,24.5, 0.0,5.0,-0.3,75.2,86.4, 155.2,-29.6,185.6,-34.9	4133	-0.31
140,25,85,115,40, -20,30,-10,50,80, 150,-15,150,0	68.157	170.0,25.2,148.0,150.8,98.2, -7.2,18.0,-58.3,78.2,93.4, 150.3,35.9,187.1,-68.0	1635	-0.143

Table 4.6: Least pth Search Results for the Six-bar Deep Digging Implement

From the feasible mechanism generated, it was found that the size of the linkage is large. Thus, an objective of minimizing the maximum length of the linkage members was considered, that is,

$$f_{obj_1} = \max\{r_i\},\ i = 1, 12 \quad (4.4)$$

The SUMT, in combination with the sub-type specific transformation method, was applied, resolving an optimized mechanism of $\mathbf{r}_{optimal} = \{143.5, 18.0, 143.5, 81.2, 101.5, -80.5, 32.2, -1.5, 78.4, 57.8, 143.5, -28.9\}^T$ cm with $\theta_{2_0} = 183.2°$ and $\theta_{local} = -69.1°$. The results of the f_{SUMT} and f_{obj} values during the search process are shown in Fig. 4.23. The optimized mechanism has a maximum link length of 143.5 cm, approximately 84 percent of the maximum length of the initial feasible linkage (169.9 cm). The generated mechanism is more compact than the one generated previously for the four-bar digging mechanism.

The displacements for the "f_{obj_1} objective optimum" mechanism are shown in Figs. 4.20 to 4.22 as dotted lines and they can be seen to have moved right up to some of the constraint limits. This demonstrates that the motion constraints were limiting factors in the searches. It is also interesting to note that three links, including the base link, base loop coupler and one of the output loop couplers, had the maximum length of 143.5 cm. This demonstrates the effectiveness of the optimization at reducing the link lengths. The search for the optimum mechanism required 4962 iterations for the f_{obj_1} objective. The final mechanism and coupler curve of the six-bar deep-digging mechanism are depicted in Figure 4.24.

4.6 Discussion

In this chapter, a case study of a deep-digging implement was presented. Type synthesis and dimensional synthesis were applied. The mechanism optimization method

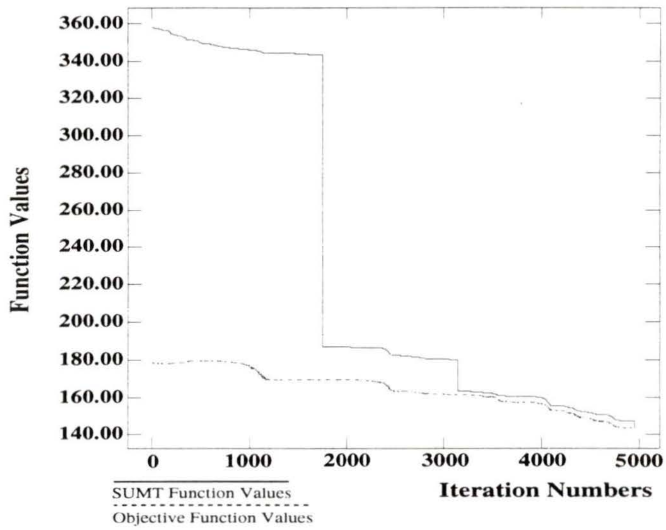

Figure 4.23: Optimization Results ($f_{obj_1} = r_1$)

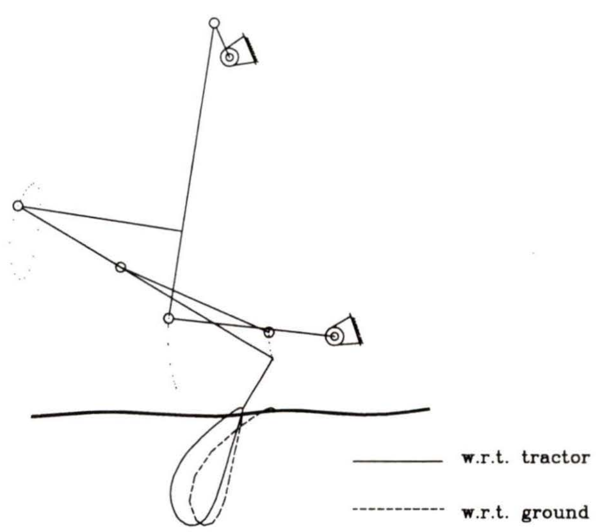

Figure 4.24: Watt I Six-bar Deep Digging Mechanism and its Coupler Curve

developed in this thesis was used to find optimal mechanism dimensions. The optimal mechanisms found for different objectives represented only local optimum for the specified motion constraints. That is, convergence to a global optimum can not be assured due of non-convex nature of the objectives in mechanism synthesis problems. Therefore, further searches from several different initial mechanisms and comparison of the found optimal values are necessary to instill confidence that a "best result" is being found.

Dimensional synthesis for the deep-digging implement was performed for both a four-bar (crank-rocker) and a six-bar (Watt I) sub-type. The required objectives were satisfied to a certain extent, e.g., motion requirements, transmission angles and input torque for the four-bar, and compactness.

Further synthesis work on the deep-digging implement would benefit from a study of the working environment. If a force model associated with digging of the earth can be set up appropriately, objectives of dynamic characteristics of the developed mechanism can be considered and hence a more appropriate mechanism could be synthesized.

The final optimized six-bar mechanism was better than the four-bar result in terms of compactness of the mechanism. However, further investigation of the six-bar mechanism in terms of associated transmission angles should be pursued in the future.

The synthesized mechanisms satisfy the motion constraints of Section 4.2.2 and the other constraints (1) through 5)) of Section 4.2.3. Adjustable bite (other constraints 6)) can be achieved by shifting the PTO ratio and adjusting the tractor speed. Overload protection for rocks (other constraints 7)) could be achieved by allowing compliance in the mechanism mounting to its frame.

Chapter 5

Conclusions and Recommendations

5.1 Conclusions of Presented Work

An optimization-based approach for the synthesis of practical mechanisms was presented in this thesis. This approach was shown to allow the consideration of practical planar mechanism types, ensures convergence to specific categories of mechanisms within specific types and allows the consideration of arbitrary performance objectives in the optimization. The presented approach was demonstrated to be effective and applicable for a variety of mechanism synthesis problems.

To facilitate the optimization of practical mechanisms, where many objectives may be considered and may be discontinuous in nature, a direct search method was used in this research. While slower to converge than gradient-based methods, direct search methods allow the consideration of arbitrary objective functions.

For direct searching, the ability to search over an unconstrained domain of search variables is an asset. In this thesis, transformations of unconstrained search variables to constrained mechanism parameters were utilized to allow unconstrained searching. A least p-th algorithm with specific tasks approximated by the specification of upper and lower limit constraints can be used to generate initial feasible mechanisms and sequentially unconstrained minimization techniques can be used to ensure motion feasibility during mechanism optimization for the specified objectives.

It is possible to transform an unconstrained n-dimensional search variable set $\mathbf{x}$ to n link parameter values $\mathbf{r}$ describing a constrained linkage within specific mechanism sub-types. The transforms were based on redefining link length limits, incorporating "Grashof-Criteria" considerations, concurrently with a transformation of the unconstrained variable set values. The technique facilitated effective optimization allowing searching and convergence within preferred mechanism sub-types.

Transforms were presented for Grashof four-bar (crank-rockers, drag-links and double-rockers) mechanisms and for six-bar (Watt I and Stephenson III) mechanisms. Similar transforms for other mechanism sub-types can be developed.

It is important to note that all of the rendered mechanisms (and all mechanisms searched over) for the examples and the case study belonged to the intended sub-type of mechanism. The optimization approach has been applied with several mechanisms types and sub-types and many tasks and has met with success in each case.

5.2 Recommendations for Future Research

Direct search techniques have been utilized to search for optimal values in this thesis. However, gradient-based methods may be considered in future research to reduce the number of evaluations. For some applications it will be either impossible or very time-consuming to obtain analytic expressions for derivatives. Consequently, difference approximations may have to be employed if gradient-based techniques are to be used.

A barrier/interior penalty method has been used in the mechanism optimization approach presented in this thesis. In the future, it may be of interest to investigate exterior/interior penalty methods. Such penalty methods would not require searching for feasible starting mechanisms.

The four-bar sub-type specific unconstrained to constrained transforms guarantee convergence within user-specified limits subject to certain conditions. However, the presented six-bar transform does not guarantee the convergence within user-specified limits for particular links. There is the possibility that the new length limits of link 9 and 10 may go beyond the original length limits for the links. Thus, further research is recommended to investigate if additional constraints (conditions) can ensure user-specified limits for six-bar link lengths.

Further research could be considered to apply the developed unconstrained to constrained transformation technique to other types of mechanisms such as geared five-bar and spatial mechanisms.

Bibliography

[1] Angeles, J., Alivizatos, A., and Akhras, R., "An Unconstrained Nonlinear Least-Square Method of Optimization of RRRR Planar Path Generators", *Mechanism and Machine Theory*, Vol. 23, No. 5, pp. 343-353, 1988.

[2] Akhras, R. and Angeles, J., "Unconstrained Nonlinear Least-Square Optimization of Planar Linkages for Ridge-Body Guidance", *Mechanism and Machine Theory*, Vol. 25, No. 1, pp. 97-118, 1990.

[3] Arora, Jasbir S., *Introduction to Optimum Design*, McGraw-Hill, New York, 1989

[4] Bagci, C. and Lee, I.J., "Optimum Synthesis of Planar Mechanisms for the Generation of Paths and Rigid-body Position via the Linear Superposition Technique", *Trans. ASME, Journal Engineering for Industry*, pp. 340-346, 1975.

[5] Beyer R., *The Kinematic Synthesis of Mechanisms*, Translated by H. Kuenzel, McGraw-Hill, New York, 1968.

[6] Burmester, L., "Lehrbuch der Kinematic", A. Felix, Leipzig, pp. 599-623, 1888, "as referenced in [7]".

[7] Chase, T.R., Erdman, A.G. and Riley, D.R., "Triad Synthesis for up to Five Design Positions with Applying to the Design of Arbitrary Planar Mechanisms",

Trans. ASME, Journal of Mechanism, Transmissions and Automation in Design, Vol. 109, pp. 426-434, 1987.

[8] Cossalter, V., Doria, A., Pasini, M., "A Simple Numerical Approach for Optimum Synthesis of a Class of Planar Mechanisms", *Mechanism and Machine Theory*, Vol. 27, pp. 357-366, 1992.

[9] Datseris, P. and Freudenstein, F., "Optimum Synthesis of Mechanisms Using Heuristics for Decomposition and Search", *Journal of Mechanical Design*, Vol. 101, pp. 381-385, July 1979.

[10] Erdman, A.G., and Sandor, G.N., *Mechanism Design: Analysis and Synthesis*, Vol.I, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[11] Fox, R.L., and Gupta, K.C., "Optimization Technology as Applied to the Design of Mechanisms", *Trans. ASME, Journal Engineering for Industry*, Vol. 95, No. 2, pp. 657-663, 1973.

[12] Fox, R.L., and Willmert, K.D., "Optimum Design of Curve-Generating Linkages with Inequality Constraints", *Trans. ASME, Journal Engineering for Industry*, Vol. 89, No. 1, pp. 144-152, 1967.

[13] Freudenstein, F., and Maki, E.R., "The Creation of Mechanism according to Kinematic Structure and Function", *General Motors Research Laboratories*, GM R-3073, Sept. 11, 1979, "as referenced in [29]".

[14] Grashof, F., "Theoretische Mashinenlehre", Leipzig, pp. 113-118, 1883, "as referenced in [28]".

[15] Hammond, C.R., and Johnson, G.E., "The Method of Alternate Formulations, an Automatic Strategy for Optimal Design", *Trans. ASME, Journal of Mechanism, Transmissions and Automation in Design*, Vol. 110, No. 4, pp. 459-464, 1988.

[16] Hanson, M.R., "A General Procedure for Dimensional Synthesis of Mechanisms", *ASME, Mechanism, Design and Synthesis*, DE-Vol. 46, pp. 67-71, 1992.

[17] Hartenberg, R.S., and Denavit, J., *Kinematic Synthesis of Linkages*, McGraw-Hill, New York, 1964.

[18] Hrones, J.A. and Nelson, L., *Analysis of Four-Bar Linkages*, John Wiley and Sons, New York, 1951.

[19] Ion, S., and Cezar, D., "On a General Method for Synthesis of the Path Approximation Mechanism", *Mechanism and Machine Theory*, Vol. 14, pp. 289-298, 1979.

[20] Jeavons, J., *How to Grow More Vegetables that You Ever Thought Possible on Less Land than You can Imagine*, Ten Speed Press. Berkeley, California, 1982.

[21] Krishnamurty, S. and Turcic, D.A., "Optimal Synthesis of Mechanisms Using Nonlinear Goal Programming Techniques", *Mechanism and Machine Theory*, Vol. 27, pp. 599-622, 1992.

[22] Kramer, S.N. and Sandor, G.N., "Selective Precision Synthesis - A General Method of Optimization for Planar Mechanisms", *Trans. ASME, Journal Engineering for Industry*, pp. 689-701, May 1975.

[23] Levitskil, N.I., and Shakvazian, K.K., *International Journal of Mechanism Science*, Vol. 2, No. 1/2, 1960, "as referenced in [2]".

[24] Levitskil, N.I., Sarkissyan, Y.L. and Gekchian, G.S., "Optimum Synthesis of 4-bar Function Generating Mechanisms", *Mechanism and Machine Theory*, Vol. 7, pp. 387-398, 1972.

[25] Lewis, D.W., and Gyory, C.K., "Kinematic Synthesis of Plane Curves", *Trans. ASME, Journal of Engineering for Industry*, Series B-89, pp. 173-176, 1967.

[26] Nolle, H., "On the Capability of the Four-bar Mechanism as a Function Generator", *Mechanical and Chemical Engineering Transactions, Institution of Engineers*, Australia, Vol. MC3, pp. 259-268, 1967.

[27] Nolle, H. and Hunt, K.H., "Optimum Synthesis of Planar Linkages to Generate Coupler Curves", *Journal of Mechanisms*, Vol. 6, pp. 267-287, 1971.

[28] Norton, T.W., Milda, A. and Howell, L.L., "Graphical Synthesis for Limit Position of a Four-Bar Mechanism Using Triangle Inequality Concept", *ASME, Mechanism, Design and Synthesis*, DE-Vol. 46, pp. 377-386, 1992.

[29] Olson, D.G., Erdman, A.G., and Riley, D.R., "A Systematic Procedure for Type Synthesis of Mechanisms with Literature Review", *Mechanism and Machine Theory*, Vol. 20, No. 4, pp. 285-295, 1985.

[30] Paradis, M.J., and Willmert, K.D., "Optimum Mechanism Design using the Gauss Constrained Method", *Trans. ASME, Journal of Mechanism, Transmissions and Automation in Design*, Vol. 105, pp. 187-196, 1983.

[31] Paul, B., "Reassessment of Grashof's Criterion" *Trans. ASME, Journal of Mechanical Design*, Vol. 101, No. 3, pp. 515-18, July, 1979.

[32] Press, W., Flannery, B. P., Teukolsky, S. A., Vetterling, W. T., *Numerical Recipes*, Cambridge University Press, Cambridge, 1986.

[33] Primrose, E.J.F., Freudenstein, F., and Sandor, G.N., "Finite Burmester Theory in Plane Kinematics", *Trans. ASME, Journal of Applied Mechanism*, pp. 683-693, 1964.

[34] Pugh, J.T., "Synthesis of Pareto Optimal Four-Bar Function Generators with Optimum Structural Error and Optimum Transmission Angles" *Trans. ASME, Journal of Mechanisms, Transmissions and Automation in Design*, Vol. 106, pp. 437-43, 1984.

[35] Rao, A.C., "Synthesis of Four-bar Function Generators Using Geometric Programming", *Mechanism and Machine Theory*, Vol. 14, pp. 141-149, 1979.

[36] Rao, S.S., Hati, S.K., "Game Theory Approach in Multicriteria Optimization of Function Generating Mechanism", *Trans. ASME, Journal of Mechanical Design*, Vol. 101, pp. 398-406, 1979.

[37] Rigelman, G.A. and Kramer, S.N., "A Computer-Aided Design Technique for the Synthesis of Planar Four-Bar Mechanisms Satisfying Specified Kinematic and Dynamic Conditions", *Trans. ASME, Journal of Mechanism, Transmissions and Automation in Design*, Vol. 110, pp. 263-268, Sept., 1988.

[38] Root, R.R., and Ragsdell, K.M., "Survey of Optimization Methods Applied to the Design of Mechanisms", *Trans. ASME, Journal of Engineering for Industry*, pp. 1036-1041, 1976.

[39] Sandor, G.N., and Erdman, A.G., *Advanced Mechanism Design: Analysis and Synthesis, Vol.II*, Prentice-Hall, Englewood Cliffs, NJ, 1984.

[40] Seireg, A., "Survey of Optimization of Mechanical Design", *Trans. ASME, Journal of Engineering for Industry*, pp. 495-499, 1972.

[41] Suh, C.H. and Ratcliffe, C.W., *Kinematics and Mechanism Design*, John Wiley, N.Y., 1978.

[42] Sutherland, G.H., and Roth, B., "Improved Least-Square Method for Designing Function-Generating Mechanisms", *Trans. ASME, Journal of Engineering for Industry*, pp. 303-307, 1975.

[43] Tandirci, M., Angeles, J. and Liu, Zheng, "Plan: A Software Package for the Optimization of Planar 4-Bar Linkages", *ASME, Computers in Engineering*, Vol. 1, 1991.

[44] Tomas, J., "The Synthesis of Mechanisms as a Non-linear Programming Problem", *Trans. ASME, Journal of Mechanisms*, pp. 119-130, 1968.

[45] Fang, X., and Podhorodeski, R.P., "Parameter Transforms allowing Unconstrained Optimization within Specific Planar Mechanism Sub-types", *Proc. of the 12th Symposium on Engineering Applications of Mechanics*, pp.163-172, Montreal, June 1994.

[46] Zhang, C., Norton R.L. and Hammond, T., "Optimization of Parameters for Specified Path Generation Using an Atlas of Coupler Curves of Geared 5-bar Linkages", *Mechanism and Machine Theory*, Vol. 19, No. 6, pp. 459-466, 1984.

Appendix A

Kinematic Modeling for Six-bar and Geared Five-bar Mechanisms

A.1 Kinematic Modeling for Six-bar Mechanisms

A.1.1 Kinematic Modeling for Watt I Six-bars

A vector representation for the Watt I six-bar is illustrated in Fig. 3.2 of Section 3.4.1. A loop-closure of vectors $\mathbf{R}_1 - \mathbf{R}_4$ representing the linkage members of the four-bar based loop yields vector equation (2.3), and θ_3, θ_4, $\dot{\theta}_3$, $\dot{\theta}_4$, $\ddot{\theta}_3$, $\ddot{\theta}_4$, ${}^lX_P$, ${}^lY_P$, ${}^l\dot{X}_P$, ${}^l\dot{Y}_P$, ${}^l\ddot{X}_P$ and ${}^l\ddot{Y}_P$ as found in Section 2.3.1. A loop-closure of vectors $\mathbf{OP}_1 - \mathbf{R}_{10} - \mathbf{R}_9 - \mathbf{OP}_2$ yields the vector equation,

$$OP_1 e^{i\theta_{OP_1}} + r_{10} e^{i\theta_{10}} - r_9 e^{i\theta_9} - OP_2 e^{i\theta_{OP_2}} = 0 \tag{A.1}$$

Breaking into X and Y components gives:

$$\begin{aligned} {}^lX_{P1} + r_{10}\cos(\theta_{10}) - r_9\cos(\theta_9) -{}^l X_{P2} &= 0 \\ {}^lY_{P1} + r_{10}\sin(\theta_{10}) - r_9\sin(\theta_9) -{}^l Y_{P2} &= 0 \end{aligned} \tag{A.2}$$

where

${}^{l}X_{P1}$ and ${}^{l}Y_{P1}$ are ${}^{l}X_{P}$ and ${}^{l}Y_{P}$ in Section 2.3.1 (equation (2.10)).

${}^{l}X_{P2} = r_1 + r_7\cos(\theta_4) + r_8\sin(\theta_4)$

${}^{l}Y_{P2} = r_7\sin(\theta_4) - r_8\cos(\theta_4)$

Solving equations (A.2) for unknowns θ_9 and θ_{10} yields,

$$\begin{aligned}\theta_9 &= 2\arctan((-B_1 \pm \sqrt{B_1^2 - 4\,A_1 C_1})/(2\,A_1))\\ \theta_{10} &= 2\arctan((-B_1 \pm \sqrt{B_1^2 - 4\,D_1 E_1})/(2\,D_1))\end{aligned}$$

where

$$\begin{aligned}A_1 &= (r_{10}^2 -{}^{l} X_{P12}^2 -{}^{l} Y_{P12}^2 - r_9^2)/r_9 - 2{}^{l}X_{P12}\\ B_1 &= 4{}^{l}Y_{P12}\\ C_1 &= (r_{10}^2 -{}^{l} X_{P12}^2 -{}^{l} Y_{P12}^2 - r_9^2)/r_9 + 2{}^{l}X_{P12}\\ D_1 &= ({}^{l}X_{P12}^2 +{}^{l} Y_{P12}^2 + r_{10}^2 - r_9^2)/r_{10} - 2{}^{l}X_{P12}\\ E_1 &= ({}^{l}X_{P12}^2 +{}^{l} Y_{P12}^2 + r_{10}^2 - r_9^2)/r_{10} + 2{}^{l}X_{P12}\end{aligned}$$

with

${}^{l}X_{P12} ={}^{l} X_{P1} -{}^{l} X_{P2}$ and ${}^{l}Y_{P12} ={}^{l} Y_{P1} -{}^{l} Y_{P2}$

For the kinematic equations describing the motion of a coupler point P3, the X and Y displacements of the coupler point P3 with respect to the local coordinate system are given by:

$$\begin{aligned}{}^{l}X_{P3} &= {}^{l}X_{P1} + r_{11}\cos(\theta_{10}) - r_{12}\sin(\theta_{10})\\ {}^{l}Y_{P3} &= {}^{l}Y_{P1} + r_{11}\sin(\theta_{10}) + r_{12}\cos(\theta_{10})\end{aligned}$$

Similarly, for the kinematic equations describing the motion of a coupler point P4, the X and Y displacements of the coupler point P4 with respect to the local coordinate system are given by:

$$\begin{aligned} {}^{l}X_{P4} &= {}^{l}X_{P2} + r_{13}\cos(\theta_9) - r_{14}\sin(\theta_9) \\ {}^{l}Y_{P4} &= {}^{l}Y_{P2} + r_{13}\sin(\theta_9) + r_{14}\cos(\theta_9) \end{aligned}$$

With respect to the global coordinates, the equations of motion for the coupler point P3 and P4 become:

$$\begin{aligned} {}^{g}X_{P3} &= {}^{l}X_{P3}\cos(\theta_{local}) -^{l} X_{P3}\sin(\theta_{local}) + X \\ {}^{g}Y_{P3} &= {}^{l}Y_{P3}\sin(\theta_{local}) +^{l} X_{P3}\cos(\theta_{local}) + Y \\ {}^{g}X_{P4} &= {}^{l}X_{P4}\cos(\theta_{local}) -^{l} X_{P4}\sin(\theta_{local}) + X \\ {}^{g}X_{P4} &= {}^{l}X_{P4}\sin(\theta_{local}) +^{l} X_{P4}\cos(\theta_{local}) + Y \end{aligned}$$

A.1.2 Search Variables for Watt I Six-bars

If the task is specified in terms of required changes in the location of the point $P3$ or $P4$, noting that the mechanism base can be relocated with respect to F_{global}, results in a reduction to twelve searching parameters, i.e., ${}^{local_g}\{x_P, y_P\} = f(\theta_2, r_i, r_j, \theta_{local})$, where $i = 1, 8$, $j = 9, 10$ (or $j = 11, 12$). In this expression, F_{local_g} refers to a F_{local} located, globally-oriented reference and allows the ability of relocating local frame F_{local} in the plane to a place that is suitable for the task.

A.1.3 Kinematic Modeling for Stephenson III Six-bars

A vector representation for the Stephenson III six-bar is illustrated in Fig. 3.3 of Section 3.4.2. A loop-closure of vectors $\mathbf{R}_1 - \mathbf{R}_4$ representing the linkage members of the four-bar based loop yields vector equation (2.3), and θ_3, θ_4, $\dot{\theta}_3$, $\dot{\theta}_4$, $\ddot{\theta}_3$, $\ddot{\theta}_4$, ${}^lX_P$, ${}^lY_P$, ${}^l\dot{X}_P$, ${}^l\dot{Y}_P$, ${}^l\ddot{X}_P$ and ${}^l\ddot{Y}_P$ as found in Section 2.3.1. A loop-closure of vectors $\mathbf{OP}_1 - \mathbf{R}_{10} - \mathbf{R}_9 - \mathbf{O_2O}$ yields the vector equation,

$$OP_1 e^{i\theta_{OP_1}} + r_{10}e^{i\theta_{10}} - r_9 e^{i\theta_9} - O_2Oe^{i\theta_{O_2O}} = 0 \tag{A.3}$$

Breaking into X and Y components gives:

$$\begin{aligned} {}^lX_{P1} + r_{10}\cos(\theta_{10}) - r_9\cos(\theta_9) - {}^lX_{O2} &= 0 \\ {}^lY_{P1} + r_{10}\sin(\theta_{10}) - r_9\sin(\theta_9) - {}^lY_{O2} &= 0 \end{aligned} \tag{A.4}$$

where

${}^lX_{P1}$ and ${}^lY_{P1}$ are ${}^lX_P$ and ${}^lY_P$ in Section 2.3.1 (equation (2.10)).
${}^lX_{O2} = r_7$ and ${}^lY_{O2} = r_8$.

Solving the above equations for the unknowns θ_9 and θ_{10} yields the same results as Section A.1.3 with ${}^lX_{P12} = {}^lX_{P1} - r_7$ and ${}^lY_{P12} = {}^lY_{P1} - r_8$.

For the kinematic equations describing the motion of a coupler point P2, the X and Y displacements of the coupler point P2 with respect to the local coordinate system are given by:

$$\begin{aligned} {}^lX_{P2} &= {}^lX_{P1} + r_{11}\cos(\theta_9) - r_{12}\sin(\theta_9) \\ {}^lY_{P2} &= {}^lY_{P1} + r_{11}\sin(\theta_9) + r_{12}\cos(\theta_9) \end{aligned}$$

With respect to the global coordinates, the displacement equations for the coupler point P2 become:

$$\begin{aligned} {}^gX_{P2} &= {}^lX_{P2}\cos(\theta_{local}) - {}^lX_{P2}\sin(\theta_{local}) + X \\ {}^gY_{P2} &= {}^lY_{P2}\sin(\theta_{local}) + {}^lX_{P2}\cos(\theta_{local}) + Y \end{aligned}$$

A.1.4 Search Variables for Stephenson III Six-bars

If the task is specified in terms of required changes in the location of the point $P2$, noting that the mechanism base can be relocated with respect to F_{global}, results in a reduction to twelve searching parameters, i.e., ${}^{local_g}\{x_P, y_P\} = f(\theta_2, r_i, \theta_{local})$, where $i = 1, 10$. In this expression, F_{local_g} refers to a F_{local} located, globally-oriented reference and allows the ability of relocating local frame F_{local} in the plane to a place that is suitable for the task.

A.2 Kinematic Modeling for Geared Five-bars

A.2.1 Displacement Analysis for Geared Five-bars

A schematic of a geared five-bar mechanism is shown in Fig. A.1. From triangle $\Delta A_0 B_0 B$, we have,

$$r_1^{*^2} = r_1^2 + r_5^2 + 2 r_1 r_5 \cos\theta_5 \tag{A.5}$$

It can be easily seen that $r^*_{1max} = (r_1 + r_5)$ when $\theta_5 = 0°$ and $r^*_{1min} = (r_1 - r_5)$ when $\theta_5 = 180°$. Note: θ_5 is a function of θ_2, i.e., $\theta_5 = f(\theta_2)$.

A loop-closure of vectors $\mathbf{r}_1^*, \mathbf{r}_2, \mathbf{r}_3$ and $\mathbf{r}_4$ representing the linkage members of the five-bar as illustrated in Fig. A.1 yields the vector equation:

$$r_2 e^{i\theta_2} + r_3 e^{i\theta_3} - r_4 e^{i\theta_4} - r_1^* e^{i\theta_1^*} = \mathbf{0} \tag{A.6}$$

Breaking into X and Y components gives:

$$\begin{aligned} r_2\cos(\theta_2) + r_3\cos(\theta_3) - r_4\cos(\theta_4) - (r_1 + r_5\cos(\theta_5)) &= 0 \\ r_2\sin(\theta_2) + r_3\sin(\theta_3) - r_4\sin(\theta_4) - r_5\sin(\theta_5) &= 0 \end{aligned} \tag{A.7}$$

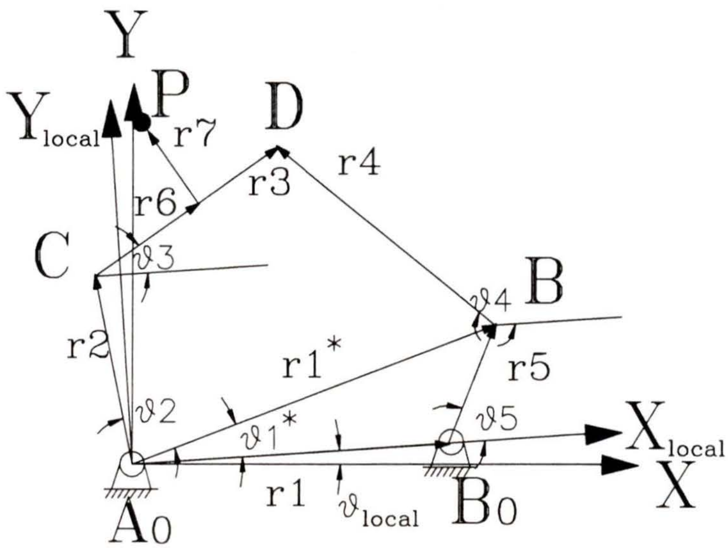

Figure A.1: Geared Five-bar Mechanism

Setting $r_2 \cos(\theta_2) - (r_1 + r_5 \cos(\theta_5)) = a$ & $r_2 \sin(\theta_2) - r_5 \sin(\theta_5) = b$ and solving the above equations for unknowns θ_3 and θ_4 yields,

$$\theta_3 = 2 \arctan((-B \pm \sqrt{B^2 - 4\,AC})/(2\,A))$$
$$\theta_4 = 2 \arctan((-B \pm \sqrt{B^2 - 4\,DE})/(2\,D)) \tag{A.8}$$

where

$$A = (r_3^2 + a^2 + b^2 - r_4^2 - 2ar_3)/(2r_3)$$
$$B = 2b$$
$$C = (r_3^2 + a^2 + b^2 - r_4^2 + 2ar_4)/(2r_3)$$
$$D = (r_4^2 + a^2 + b^2 - r_3^2 + 2ar_4)/(-2r_4)$$
$$E = (r_4^2 + a^2 + b^2 - r_3^2 - 2ar_4)/(-2r_4)$$

The X and Y displacements of the coupler point with respect to the local coordinate system are given by:

$$
\begin{aligned}
{}^{l}X_P &= r_2\cos(\theta_2) + r_6\cos(\theta_3) - r_7\sin(\theta_3) \\
{}^{l}Y_P &= r_2\sin(\theta_2) + r_6\sin(\theta_3) + r_7\cos(\theta_3)
\end{aligned}
\tag{A.9}
$$

With respect to the global coordinates, the equations of motion for the coupler point become:

$$
\begin{aligned}
{}^{g}X_P &= {}^{l}X_P C_{local} - {}^{l}Y_P S_{local} + X \\
{}^{g}Y_P &= {}^{l}X_P S_{local} + {}^{l}Y_P C_{local} + Y
\end{aligned}
$$

where θ_{local} is the angular orientation of the local coordinate system; X is X displacement of the local coordinate system; Y is Y displacement of the local coordinate system and $S_{local} = \sin(\theta_{local})$ and $C_{local} = \cos(\theta_{local})$.

A.2.2 Search Variables for Geared Five-bars

If the task is specified in terms of required changes in the location of the point P, noting that the mechanism base can be relocated with respect to F_{global}, results in a reduction to 11 searching parameters, i.e., ${}^{local_g}\{x_P, y_P\} = f(\theta_2, \theta_{20}, \theta_{50}, r_i, \theta_{local}, \mathrm{GR})$, where i=1,7. In the expression, GR is short for Geared Ratio and F_{local_g} refers to a F_{local} located, globally-oriented reference and allows the ability of relocating local frame F_{local} in the plane to a place that is suitable for the task.

Appendix B

Length Limit Redefinitions for Four-bar Mechanisms

B.1 Length Limit Redefinitions for Crank-rocker Mechanisms

The unconstrained search variable to constrained mechanism parameter transformations for crank-rockers were presented in Section 3.2. If the value returned for r_3 in steps (CRg) to (CRi) is small it is possible that the $r^*_{lower_4}$ generated by step (CRl) may be greater than $r^*_{upper_4}$. In this contradictory case the lower limits for both r_3 and r_4 must be redefined. In this appendix expressions are developed for $r^*_{lower_3}$ and $r^*_{lower_4}$ that will ensure sub-type specific mechanisms within the specified ranges. First, set a equation as follows:

$$(r^*_{upper_4} - r^*_{lower_4})/Or_4 = (r^*_{upper_3} - r^*_{lower_3})/Or_3 \tag{B.1}$$

where $Or_3 = r_{upper_3} - r_{lower_3}$ and $Or_4 = r_{upper_4} - r_{lower_4}$. This equation will ensure that the ratios of redefined range/original range will be equal for r_3 and r_4. Rearranging the above equation yields:

$$Or_4 r^*_{lower_3} - Or_3 r^*_{lower_4} = Or_4 r^*_{upper_3} - Or_3 r^*_{upper_4} \tag{B.2}$$

Then set another equation according to Grashof criteria, i.e.,

$$r^*_{lower_3} + r^*_{lower_4} \geq r_1 + r_2 \tag{B.3}$$

which ensures a Crank-rocker mechanism. Setting above equation to the limiting case yields:

$$r^*_{lower_3} + r^*_{lower_4} = r_1 + r_2 \tag{B.4}$$

Solving equations (B.2) and (B.4) for $r^*_{lower_3}$ and $r^*_{lower_4}$ yields,

$$r^*_{lower_3} = \frac{Or_3(r_1 + r_2 - r^*_{upper_4}) + Or_4 r^*_{upper_3}}{Or_3 + Or_4} \tag{B.5}$$

$$r^*_{lower_4} = \frac{Or_4(r_1 + r_2 - r^*_{upper_3}) + Or_3 r^*_{upper_4}}{Or_3 + Or_4} \tag{B.6}$$

It must be demonstrated that $r^*_{lower_i} > r_{lower_i}$, and $r^*_{upper_i} < r_{upper_i}$, i=3,4.

For $r^*_{lower_3} > r_{lower_3}$:

Setting the right hand side of equation (B.5) to r_{lower_3} and checking if $r^*_{lower_3} > r_{lower_3}$ yields the requirement that $Or_3(r_1 + r_2 - r^*_{upper_4}) + Or_4 r^*_{upper_3} - r_{lower_3}(Or_3 + Or_4) > 0$, or after rearranging it,

$$Or_3(r_1 + r_2 - r_{lower_3} - r^*_{upper_4}) + Or_4(r^*_{upper_3} - r_{lower_3}) > 0 \tag{B.7}$$

The extreme cases of the minimum possible value of $r^*_{upper_3} \leq r_{upper_3}$ and the maximum possible value of $r^*_{upper_4} \leq r_1$ must be considered. For $r^*_{upper_3} = r_{upper_3}$ and $r^*_{upper_4} = r_1$, equation (B.7) becomes

$$r_2 - r_{lower_3} + Or_4 = r_2 + r_{upper_4} - r_{lower_3} - r_{lower_4} > 0 \tag{B.8}$$

After noting that $Or_3 = r_{upper_3} - r_{lower_3}$. For the limiting case of $r_2 = r_{lower_2}$ when $x_2 \to -\infty$, we have the following condition for search limit specification:

$$r_{lower_2} + r_{upper_4} > r_{lower_3} + r_{lower_4} \tag{B.9}$$

The condition of (B.9) is easily satisfied and will guarantee $r^*_{lower_3} > r_{lower_3}$.

For $r^*_{lower_4} > r_{lower_4}$:

Setting the right hand side of equation (B.6) to r_{lower_4} and checking if $r^*_{lower_4} > r_{lower_4}$ yields the requirement that $Or_4(r_1 + r_2 - r^*_{upper_3}) + Or_3 r^*_{upper_4} - r_{lower_4}(Or_3 + Or_4) > 0$, or after rearranging it,

$$Or_4(r_1 + r_2 - r^*_{upper_3} - r_{lower_4}) + Or_3(r^*_{upper_4} - r_{lower_4}) > 0 \tag{B.10}$$

The extreme cases of the minimum possible value of $r^*_{upper_4} \leq r_{upper_4}$ and the maximum possible value of $r^*_{upper_3} \leq r_1$ must be considered. For $r^*_{upper_4} = r_{upper_4}$ and $r^*_{upper_3} = r_1$, the equation (B.10) becomes

$$r_2 - r_{lower_4} + Or_3 = r_2 + r_{upper_3} - r_{lower_3} - r_{lower_4} > 0 \tag{B.11}$$

After noting that $Or_4 = r_{upper_4} - r_{lower_4}$. For the limiting case of $r_2 = r_{lower_2}$ when $x_2 \to -\infty$, we have the following condition for search limit specification:

$$r_{lower_2} + r_{upper_3} > r_{lower_3} + r_{lower_4} \tag{B.12}$$

The condition of (B.12) is easily satisfied and will guarantee $r^*_{lower_4} > r_{lower_4}$.

For $r^*_{lower_3} < r_{upper_3}$:

Setting the right hand side of equation (B.5) to r_{upper_3} and checking if $r^*_{lower_3} < r_{lower_3}$ yields the requirement that $r_{upper_3}(Or_3 + Or_4) - Or_3(r_1 + r_2 - r^*_{upper_4}) - Or_4 r^*_{upper_3} > 0$, or after rearranging it,

$$Or_3(r_{upper_3} + r^*_{upper_4} - r_1 - r_2) + Or_4(r_{upper_3} - r^*_{upper_3}) > 0 \tag{B.13}$$

Since it is always true that $r_{upper_3} \geq r^*_{upper_3}$, we have the condition that under the assumption of the limiting case of $x_1 \rightarrow \infty$, $x_2 \rightarrow \infty$, and $r^*_{upper_4} = r_{upper_4}$,

$$r_{upper_3} + r_{upper_4} \geq r_{upper_1} + r_{upper_2} \tag{B.14}$$

which was one of the length limit compatability relations (see Section 3.2, equation (3.6)) for crank-rockers with $r_{long} = r_1$.

For $r^*_{lower_4} < r_{upper_4}$:

Setting the right hand side of eqn. (B.6) to r_{upper_4} and checking if $r^*_{lower_4} < r_{lower_4}$ yields the requirement that $r_{upper_4}(Or_3 + Or_4) - Or_4(r_1 + r_2 - r^*_{upper_3}) - Or_3 r^*_{upper_4} > 0$, or after rearranging it,

$$Or_4(r_{upper_4} + r^*_{upper_3} - r_1 - r_2) + Or_3(r_{upper_4} - r^*_{upper_4}) > 0 \tag{B.15}$$

Since it is always true that $r_{upper_4} \geq r^*_{upper_4}$, we have the condition that under the assumption of the limiting case of $x_1 \rightarrow \infty$, $x_2 \rightarrow \infty$, and $r^*_{upper_4} = r_{upper_4}$,

$$r_{upper_3} + r_{upper_4} \geq r_{upper_1} + r_{upper_2} \tag{B.16}$$

which is the identical compatability relation as equation (B.14)

Therefore, it is concluded that $r^*_{lower_3} < r_{upper_3}$ and $r^*_{lower_4} < r_{upper_4}$ without any condition.

B.2 Length Limit Redefinitions for Drag-link Mechanisms

The unconstrained search variable to constrained mechanism parameter transformations for drag-links were presented in Section 3.3. If the value returned for r_2 in steps

(DLg) to (DLi) is small it is possible that the $r^*_{lower_3}$ generated by step (DLl) may be greater than $r^*_{upper_3}$. In this contradictory case the lower limits for both r_2 and r_3 must be redefined. In this appendix expressions are developed for $r^*_{lower_2}$ and $r^*_{lower_3}$ that will ensure drag-link mechanisms within the specified ranges. Similar to the crank-rocker case of Appendix B.1, we set the equation

$$\frac{r^*_{upper_2} - r^*_{lower_2}}{Or_2} = \frac{r^*_{upper_3} - r^*_{lower_3}}{Or_3} \tag{B.17}$$

where $Or_2 = r_{upper_2} - r_{lower_2}$ and $Or_3 = r_{upper_3} - r_{lower_3}$; and observe the Grashof criteria,

$$r^*_{lower_2} + r^*_{lower_3} \geq r_1 + r_4 \tag{B.18}$$

The above equations are equivalent to

$$\begin{aligned} Or_3 r^*_{lower_2} - Or_2 r^*_{lower_3} &= Or_3 r^*_{upper_2} - \dot{O}r_2 r^*_{upper_3} \\ r^*_{lower_2} + r^*_{lower_3} &= r_1 + r_4 \end{aligned} \tag{B.19}$$

Solving equations (B.19) for $r^*_{lower_2}$ and $r^*_{lower_3}$ yields,

$$\begin{aligned} r^*_{lower_2} &= \frac{Or_2(r_1 + r_4 - r^*_{upper_3}) + Or_3 r^*_{upper_2}}{Or_2 + Or_3} \\ r^*_{lower_3} &= \frac{Or_3(r_1 + r_4 - r^*_{upper_2}) + Or_2 r^*_{upper_3}}{Or_2 + Or_3} \end{aligned} \tag{B.20}$$

It must be demonstrated that $r^*_{lower_i} > r_{lower_i}$, and $r^*_{upper_i} < r_{upper_i}$, i=2,3.

With the same methods used in Section B.1, conditions for search limit specification can be derived. In summary, these conditions are the following:

For $r^*_{lower_2} > r_{lower_2}$: $r_{lower_4} + r_{upper_3} > r_{lower_2} + r_{lower_3}$;

For $r^*_{lower_3} > r_{lower_3}$: $r_{lower_4} + r_{upper_2} > r_{lower_2} + r_{lower_3}$;

For $r^*_{lower_2} < r_{upper_2}$ and $r^*_{lower_3} < r_{upper_3}$: $r_{upper_3} + r_{upper_2} \geq r_{upper_1} + r_{upper_4}$.

The above conditions are either easily satisfied or in the case of the final two correspond to length limit compatibility for drag-links with $r_{long} = r_4$.

B.3 Length Limit Redefinitions for Double-rocker Mechanisms

The unconstrained search variable to constrained mechanism parameter transformations for double rockers were presented in Section 3.3. If the value returned for r_2 in steps (DRg) to (DRi) is small it is possible that the $r^*_{lower_4}$ generated by step (DRl) may be greater than $r^*_{upper_4}$. In this case the lower limits for both r_2 and r_4 must be redefined. In this appendix expressions are developed for $r^*_{lower_2}$ and $r^*_{lower_4}$ that will ensure double-rocker mechanisms within the specified ranges. Similar to the crank-rocker case of Appendix B.1, we set the equation

$$\frac{r^*_{upper_2} - r^*_{lower_2}}{Or_2} = \frac{r^*_{upper_4} - r^*_{lower_4}}{Or_4} \tag{B.21}$$

where $Or_2 = r_{upper_2} - r_{lower_2}$ and $Or_4 = r_{upper_4} - r_{lower_4}$; and observe the Grashof criteria,

$$r^*_{lower_2} + r^*_{lower_4} \geq r_1 + r_3 \tag{B.22}$$

The above equations are equivalent to

$$\begin{aligned} Or_4 r^*_{lower_2} - Or_2 r^*_{lower_4} &= Or_4 r^*_{upper_2} - Or_2 r^*_{upper_4} \\ r^*_{lower_2} + r^*_{lower_4} &= r_1 + r_3 \end{aligned} \tag{B.23}$$

Solving equations (B.23) for $r^*_{lower_2}$ and $r^*_{lower_4}$ yields,

$$r^*_{lower_2} = \frac{Or_2(r_1 + r_3 - r^*_{upper_4}) + Or_4 r^*_{upper_2}}{Or_2 + Or_4} \tag{B.24}$$

$$r^*_{lower_4} = \frac{Or_4(r_1 + r_3 - r^*_{upper_2}) + Or_2 r^*_{upper_4}}{Or_2 + Or_4} \tag{B.25}$$

It must be demonstrated that $r^*_{lower_i} > r_{lower_i}$, and $r^*_{upper_i} < r_{upper_i}$, i=2,4.

With the same methods used in Section B.1, conditions for search limit specification can be derived. In summary, these conditions are the following:

For $r^*_{lower_2} > r_{lower_2}$: $r_{lower_3} + r_{upper_4} > r_{lower_2} + r_{lower_4}$

For $r^*_{lower_4} > r_{lower_4}$: $r_{lower_3} + r_{upper_2} > r_{lower_2} + r_{lower_4}$

For $r^*_{lower_2} < r_{upper_2}$ and $r^*_{lower_4} < r_{upper_4}$: $r_{upper_2} + r_{upper_4} \geq r_{upper_1} + r_{upper_3}$

The above conditions are either easily satisfied or in the case of the final two correspond to length limit compatibility for double-rockers with $r_{long} = r_1$.

Appendix C

Link Length Redefinitions for Six-bar Mechanisms

C.1 Length Limit Redefinitions for Watt I Six-bar Mechanisms

The unconstrained search variable to constrained mechanism parameter transformations for Watt I six-bar mechanism were presented in Section 3.4. If the value returned for r_9 in steps (WIn) to (WIp) is small it is possible that the $r^*_{lower_{10}}$ generated by step (WIr) or (WIs) may be greater than $r^*_{upper_{10}}$. In this case the lower and upper limits for both r_9 and r_{10} must be redefined. In this appendix expressions are developed for $r^*_{lower_i}$ and $r^*_{upper_i}$, i=9,10 to ensure Watt I six-bar mechanisms with compatible $r^*_{lower_i}$ and $r^*_{upper_i}$ values. If $r^*_{upper_{10}} < r^*_{lower_{10}}$, then equations are set as follows:

$$\frac{r^*_{upper_{10}} - r^*_{lower_{10}}}{Or_{10}} = \frac{r^*_{upper_9} - r^*_{lower_9}}{Or_9}$$

$$\begin{aligned} r^*_{lower_9} + r^*_{lower_{10}} &\geq dmax \\ \mid r^*_{upper_{10}} - r^*_{lower_9} \mid &\leq dmin \\ \mid r^*_{upper_9} - r^*_{lower_{10}} \mid &\leq dmin \end{aligned} \tag{C.1}$$

where $Or_9 = r_{upper_9} - r_{lower_9}$ and $Or_{10} = r_{upper_{10}} - r_{lower_{10}}$. Assuming that $r^*_{upper_{10}} > r^*_{lower_9}$ and $r^*_{upper_9} > r^*_{lower_{10}}$ is desired allows equations (C.1) to be rewritten as

$$\begin{aligned} Or_{10}r^*_{lower_9} - Or_9r^*_{lower_{10}} - Or_{10}r^*_{upper_9} + Or_9r^*_{upper_{10}} &= 0 \\ r^*_{lower_9} + r^*_{lower_{10}} &= dmax \\ -r^*_{lower_9} + r^*_{upper_{10}} &= dmin \\ -r^*_{lower_{10}} + r^*_{upper_9} &= dmin \end{aligned} \tag{C.2}$$

Solving equations (C.2) for $r^*_{lower_9}, r^*_{upper_9}, r^*_{lower_{10}}$ and $r^*_{upper_{10}}$ yields,

$$\begin{aligned} r^*_{lower_9} &= \frac{Or_9(dmax - dmin) + Or_{10}(dmax + dmin)}{2(Or_9 + Or_{10})} \\ r^*_{upper_9} &= \frac{Or_9(dmax + 3dmin) + Or_{10}(dmax + dmin)}{2(Or_9 + Or_{10})} \\ r^*_{lower_{10}} &= \frac{Or_9(dmax + dmin) + Or_{10}(dmax - dmin)}{2(Or_9 + Or_{10})} \\ r^*_{upper_{10}} &= \frac{Or_9(dmax + dmin) + Or_{10}(dmax + 3dmin)}{2(Or_9 + Or_{10})} \end{aligned} \tag{C.3}$$

It is concluded that all solutions above are great than 0, $r^*_{upper_9} > r^*_{lower_9}$ and $r^*_{upper_{10}} > r^*_{lower_{10}}$. However, it has not been demonstrated that $r^*_{lower_i} > r_{lower_i}$ and $r^*_{upper_i} < r_{upper_i}$, i=9,10, or what further conditions are required to ensure that the redefined limits are within the original specified limits. To do so will require analytical expressions for d_{max} and d_{min} in terms of the link lengths. Derivation of any further conditions is left for further research.

Appendix D

Kinetostatic Analysis of Four-bar Mechanism

The matrix method [10] is utilized to do the force analysis of a four-bar linkage. This method considers all of the inertia forces together and its advantage is that the equations of motion are quickly derived. Referring to the three free body diagrams of Fig. D.1, three kinetostatic equilibrium equations are written for each link, i.e., $\Sigma F_{xi} = F_{gix}$, $\Sigma F_y = F_{giy}$, and $\Sigma T_g = T_{gi}$, where $\mathbf{F}_{gi} = -M_i \mathbf{A}_{gi} e^{i(\beta_i + \theta_{loc})}$ and $\beta_i = \arctan(A_{giy}/A_{gix})$, i=2, 3, 4 with $\mathbf{A}_{gi}$ denoting the center of mass acceleration of link i; $T_{gi} = -I_{Gi}\ddot{\theta}_i$, i=2, 3, 4. Let $\mathbf{F}_{ij}$ be the bearing force link i applies to link j.

Considering link 2 (Fig_A of Fig. D.1):

$$F_{12x} + F_{32x} + F_{g2x} = 0;$$
$$F_{12y} + F_{32y} + F_{g2y} = 0;$$
$$T_{g2} + T_s + F_{12x} r_{g2} S_2 - F_{12y} r_{g2} C_2 - F_{32x}(r_2 - r_{g2}) S_2 + F_{32y}(r_2 - r_{g2}) C_2 = 0;$$

where T_s is the required driving torque.

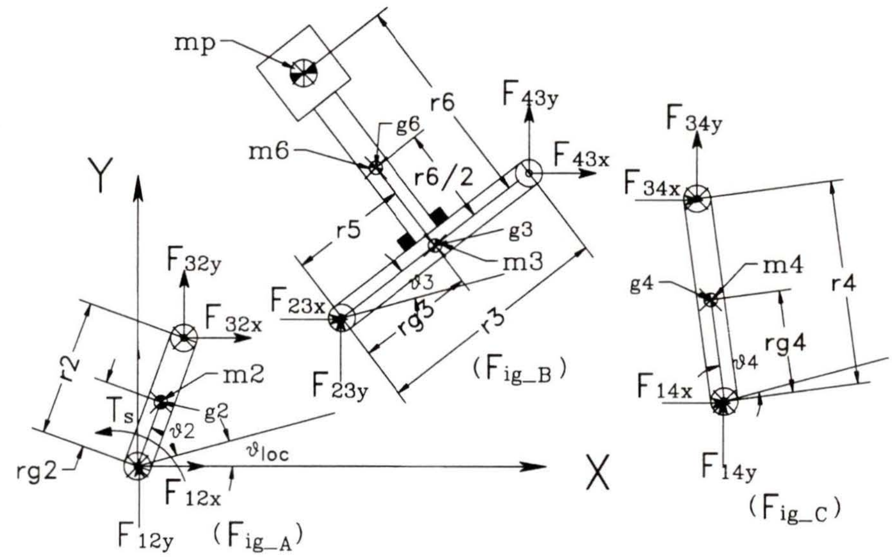

Figure D.1: Kinetostatic Free-body Diagrams

Considering link 3 (Fig_B of Fig. D.1):

$F_{23x} + F_{G3x} + F_{43x} = 0$;

$F_{23y} + F_{G3y} + F_{43y} = 0$;

$T_{G3} + F_{23x}R_{G3}SS_3 - F_{23y}R_{G3}CC_3 - F_{43x}(r_3S_3 - R_{G3}SS_3) + F_{43y}(r_3C_3 - R_{G3}CC_3) = 0;$

where $R_{G3} = \sqrt{R_{G3x}^2 + R_{G3y}^2}$, $R_{G3x} = (m_3r_{g3} + m_6r_5 + m_pr_5)/M3$, $R_{G3y} = (m_6r_6/2 + m_pr_6)/M3$, and $M3 = m_3 + m_6 + m_p$.

Considering link 4 (Fig_C of Fig. D.1):

$F_{34x} + F_{14x} + F_{g4x} = 0;$

$F_{34y} + F_{14y} + F_{g4y} = 0;$

$T_{g4} + F_{14x}r_{g4}S_4 - F_{14y}r_{g4}C_4 - F_{34x}(r_4 - r_{g4})S_4 + F_{34y}(r_4 - r_{g4})C_4 = 0;$

The equations above represent a system of nine equations linear in nine unknowns describing the instantaneous dynamic force and torque equilibrium on each moving link of the four-bar mechanism of Fig. 2.17. The nine unknowns are $F_{12x}, F_{12y}, F_{23x}$, $F_{23y}, F_{34x}, F_{34y}, F_{14x}, F_{14y}$, and T_s. Rewriting the equations so that the unknown terms are shifted to the right side yields:

$$
\begin{aligned}
F_{g2x} &= -F_{12x} + F_{23x} \\
F_{g2y} - m_2 g &= -F_{12y} + F_{23y} \\
T_{g2} &= -T_s - F_{12x} r_{g2} S_2 + F_{12y} r_{g2} C_2 - F_{23x} R_2 S_2 + F_{23y} R_2 C_2 \\
F_{G3x} &= -F_{23x} + F_{34x} \\
F_{G3y} - M_3 g &= -F_{23y} + F_{34y} \\
T_{G3} &= -F_{23x} R_{G3} SS_3 + F_{23y} R_{G3} CC_3 - F_{34x} RR31 + F_{43y} RR32 \\
F_{g4x} &= -F_{34x} - F_{14x} \\
F_{g4y} - m_4 g &= -F_{34y} - F_{14y} \\
T_{g4} &= -F_{14x} r_{g4} S_4 + F_{14y} r_{g4} C_4 + F_{34x} R4 S_4 - F_{34y} R4 C_4
\end{aligned}
\tag{D.1}
$$

where,

$R_i = r_i - r_{gi}$, $F_{ij} = -F_{ji}$, i,j = 1 to 4

$RR31 = r_3 S_3 - R_{G3} SS_3$ and $RR32 = r_3 C_3 - R_{G3} CC_3$

$S_2 = \sin(\theta_{20} + \theta_2 + \theta_{loc})$ and $C_2 = \cos(\theta_{20} + \theta_2 + \theta_{loc})$

$S_i = \sin(\theta_i + \theta_{loc})$ and $C_i = \cos(\theta_i + \theta_{loc})$, i = 3,4

$SS_3 = \sin(\theta_3 + \alpha_3 + \theta_{loc})$ and $CC_3 = \cos(\theta_3 + \alpha_3 + \theta_{loc})$

and $\alpha_3 = \arctan(R_{G3y}/R_{G3x})$

The system of equations (C.4) can be expressed in symbolic form,

$$[\mathbf{F}_I] = [\mathbf{L}]\,[\mathbf{F}_B] \tag{D.2}$$

where

$[\mathbf{F}_I] \equiv$ a column matrix of known external load plus inertia forces and torques

$$[\mathbf{F}_I] = (F_{g2x}\ F_{g2y} - m_2 g\ T_{g2}\ F_{G3x}\ F_{G3y} - M_3 g\ T_{G3}\ F_{g4x}\ F_{g4y} - m_4 g\ T_{g4})^T \quad (D.3)$$

$[\mathbf{F}_B]$: column matrix of unknown bearing forces and input torques

$$[\mathbf{F}_B] = (F_{12x}\ F_{12y}\ T_s\ F_{23x}\ F_{23y}\ F_{34x}\ F_{34y}\ F_{14x}\ F_{14y})^T \quad (D.4)$$

$[\mathbf{L}]$: square matrix of known linkage parameters and position angles

$$[\mathbf{L}] = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ r_{g2}S_2 & r_{g2}C_2 & -1 & -R_2S_2 & R_2C_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -R_{G3}SS_3 & R_{G3}CC_3 & -RR31 & RR32 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & R4S_4 & -R4C_4 & -r_{g4}S_4 & r_{g4}C_4 \end{bmatrix} \quad (D.5)$$

Assuming uniform mass/length for the links we have $M_2 = m_2$, $M_4 = m_4$,$M3 = m_3 + m_6 + m_p$, $m_i = dr_i$, where d is the mass density/unit length density and r_i is the length of the link.

The inertias are $I_{G2} = I_2$, $I_{G3} = I_3 + m_3((R_{G3x} - r_{g3})^2 + R_{G3y}^2) + m_6((R_{G3x} - r_5)^2 + (R_{G3y} - r_6/2)^2) + m_p((R_{G3x} - r_5)^2 + (R_{G3y} - r_6)^2)$, $I_{G4} = I_4$, where $I_i = m_i r_i^2/12$, i=2, 3, 4 (slender rods are assumed).

Appendix E

Optimization Program Listings

The program implemented in this work is broken into several different sub-programs. The source code for each sub-program is presented in the following sections. A brief description of what the sub-program does precedes the codes.

```
c*****************************************************************************
c       main program for four-bar linkage optimization                       *
c         - inout/output for developed optimization-based synthesis approach *
c*****************************************************************************
        parameter (novar=8,lclu=2,nocsr=9,nopt=181,nores=21,n1=9)
        common /iout/in,io,nfeval,nsi,m1,np,ncr
        common /iout1/r
        common /inilen/lc11,lc13,lc14,lc21,lc23,lc24,linkno,iparcel
        common /fct/lc,csr,w,wm,res,w2,csr0,l,fn,funk1,transang,objv,obja
        common /fct1/mc,mid,nan,ncp,ml,nk,nnp,pp,p1,lmax,iobj,mm,nn,mm1,nn1
        real l(novar),lc(lclu,novar)
        dimension csr(nocsr,nopt),w(nocsr,nopt),wm(nocsr),res(nores,nopt),x(novar),csr0(4),bb(n1,nopt)
        real lc11,lc13,lc14,lc21,lc23,lc24,lamda(novar),transang(nopt)
        dimension p(novar+1,novar),y(novar+1),t(novar),e(novar,novar),pp1(novar)
        integer mc(4),ml(novar),mid(novar),p1,lmax,pp
        character*20 fname1,fname2,fname(30)
c        *****  Open input/output data file  *****
        in=45
        io=46
        iter=0
        write(*,*) "please input the name of the data file:"
        read(*,*) fname1
        write(*,*) "please input the name of the output-data file:"
        read(*,*) fname2
        write(*,*) "fname1=",fname1,"  fname2=",fname2
        open(in,file=fname1)
        open(io,file=fname2)
        open(10,file='fname.dat')
```

```
      read(10,*) (fname(i),i=1,27)
      do 901 i=1,27
        j=i+10
        open(j,file=fname(i))
901   continue
c
c        nan: mechanism analysis flag (0=kinematic analysis;1=least pth search;
c                                      2=SUMT optimization)
c        mech: type of mechanisms (1= four-bar; 2= six-bar)
c        ncp: coupler point identifier (0=no point, 1=point)
c        incop: th2 increment option (0=specified values, 1=equally spaced generated values)
c        inc: th2 increment for incop=1, :number of specification incop=0)
c        w2: input rotational velocity
c        lmax: flag to indicate the longest link (1=base link; 2=coupler; 3=follower)
c        iobj:user specified objective function identifier
      read(in,*) nan,mech,ncp,incop,inc,w2,lmax,iobj,mm,nn,mm1,nn1
      write(io,*) "nan,mech,ncp,incop,inc,w2,lmax,iobj,mm,nn,mm1,nn1:"
     &           ,nan,mech,ncp,incop,inc,w2,lmax,iobj,mm,nn,mm1,nn1
c        ftol: fractional convergence tolerance;input
c        ptol: convergence tolerance of distance between vertices;input
c        itmax: maximum number of iteration
c        nsi: flag to print data files for simulation
c        ncr: flag to identify the cross link (0=n, 1=y)
c        p1: exponent of objective function
c        linkno: flag to identify which link to minimize
c        iparcel: flag to identify the parcel (0=n, 1=y)
      read(in,*) ftol,itmax,ptol,nsi,ncr,p1,linkno,iparcel
      write(io,*) "ftol,itmax,ptol:,etc.",ftol,itmax,ptol,nsi,ncr,p1,linkno,iparcel
c
c       mp: number of simplex's vertices; np:number of search variables;
c       ndim: number of vector dimension
      read(in,*) mp,np,ndim
      write(io,*) "simplex vertices mp,search variables np,ndim:",mp,np,ndim
      n=4
      if(ncp.eq.1) n=np
c
c        input initial values of search variables
c        l(1 to 6) : lengths of link 1 through 6, 17: th20; 18: th_local.
      read(in,*) (l(i),i=1,n)
      write(io,*) "initial variables l(*)=",(l(i),i=1,n)
c       ***** if nan=0, go kinematic analysis directly *****
      if (nan.eq.0) goto 3
c
c        input lower and upper constraints of the search variables
c        ml(*): flags to indicate various link dimensions, 1=y, 0=n
c        lc(1,*) and lc(2,*): lower and upper link dimension constraints
      read(in,*) (ml(i),i=1,n)
      read(in,*) (lc(1,i),i=1,n)
      write(io,*) "lower constraints lc(1,*)=",(lc(1,i),i=1,n)
      read(in,*) (lc(2,i),i=1,n)
      write(io,*) "upper constraints lc(2,*)=",(lc(2,i),i=1,n)
c       ***** get some of initial constraints of the search variables *****
      lc11=lc(1,1)
      lc13=lc(1,3)
      lc14=lc(1,4)
      lc21=lc(2,1)
      lc23=lc(2,3)
      lc24=lc(2,4)
c
c       modifying the searching variable constraints lc(*,*) to ensure
```

```
c        late "Grashof subroutine" compatibility
        if(lmax.eq.1) then
          if(lc11.lt.l(2)) lc(1,1)=l(2)
          if(lc13.lt.l(2)) lc(1,3)=l(2)
          if(lc23.gt.l(1)) lc(2,3)=l(1)
          lc(1,4)=l(2)+l(1)-l(3)
          if(lc24.gt.l(1)) lc(2,4)=l(1)
          elseif(lmax.eq.2) then
              if(lc13.lt.l(2)) lc(1,3)=l(2)
              if(lc11.lt.l(2)) lc(1,1)=l(2)
              if(lc21.gt.l(3)) lc(2,1)=l(3)
              lc(1,4)=l(2)+l(3)-l(1)
              if(lc24.gt.l(3)) lc(2,4)=l(3)
              else
                  if(lc14.lt.l(2)) lc(1,4)=l(2)
                  if(lc13.lt.l(2)) lc(1,3)=l(2)
                  if(lc23.gt.l(4)) lc(2,3)=l(4)
                  lc(1,1)=l(2)+l(4)-l(3)
                  if(lc21.gt.l(4)) lc(2,1)=l(4)
        endif
c
c        the constrained search variables l(*) are converted into
c        unconstrained search variables x(1 to np)
        do 7 i=1,n
          if (ml(i).eq.0) goto 7
            x(i)=alog((l(i)-lc(1,i))/(lc(2,i)-l(i)))
7       continue
        nk=n
        write(io,*) "initial unconstrained search variables=",(x(i),i=1,nk)
c
c        get specified input angles deltath2 - cst(1,i) for ith specification
c        if incop=1, go equally spaced generated values,otherwise, go specified values
3       if (incop.eq.1) goto 5
        n=inc
        read(in,*) (csr(1,i), i=1,n)
        write(io,*) "input angles deltath2=",(csr(1,i), i=1,n)
        goto 30
5       n=360/inc
        val=0.
        do 10 i=1,n+1
          csr(1,i)=val
          val=val+inc
10      continue
        n=n+1
c        *****  if nan=0, go kinematic analysis directly  *****
30      if (nan.eq.0) goto 90
c
c        *****  get specified motion constraints  *****
c          mt: optimization type, (1=least pth, 2=SUMT)
c          nw: motion constraint weighting flag (0=spec., 1=all 1)
c          mc(1 to 4): flag for disp. to be constrained (1=y, 0=n)
c          csr(i*2,j) and csr(i*2+1,j): lower and upper motion constraints
        read(in,*) mt,nw
        write(io,*) "optimization type mt,constraint weighting flag nw:",mt,nw
        read(in,*) (mc(i),i=1,4)
        write(io,*) "flag for disp. to be constrained:mc(*)=",(mc(i),i=1,4)
        do 40 i=1,4
          if (mc(i).eq.0) goto 40
            read(in,*) (csr(i*2,j),j=1,n)
            write(io,*) "lower motion constraints csr(i*2,j)=",(csr(i*2,j),j=1,n)
```

```
            read(in,*) (csr(i*2+1,j),j=1,n)
            write(io,*) "upper motion constraints csr(i*2+1,j)=",(csr(i*2+1,j),j=1,n)
            if (nw.eq.0) goto 35
                do 34 j=1,n
                  w(i*2-1,j)=1.
                  w(i*2,j)=1.
c                 w(*): weights for upper & lower constraints
34              continue
                goto 40
35          read(in,*) (w(i*2-1,j),j=1,n)
            write(io,*) "weights for lower constraints w(*):",(w(i*2-1,j),j=1,n)
            read(in,*) (w(i*2,j),j=1,n)
            write(io,*) "weights for upper constraints w(*):",(w(i*2,j),j=1,n)
40      continue
        do 51 j=1,n
            write(11,*) csr(1,j),csr(2,j)
            write(12,*) csr(1,j),csr(3,j)
            write(13,*) csr(1,j),csr(6,j)
            write(14,*) csr(1,j),csr(7,j)
            write(15,*) csr(1,j),csr(8,j)
            write(16,*) csr(1,j),csr(9,j)
51      continue
c        *****  if mt=1, go least pth only, otherwise go SUMT  *****
        if (mt.eq.1) then
c          *****  read a parameter p for least p-th method  *****
          read(in,*) pp
          write(io,*) "parameter for least pth:p=",pp
        else
c
c          input parameters for SUMT
c          mid(*): flags for max. abs. value minimization (1=y,0=n)
c          wm(*): weights for max. abs. value minimization
c          r: r value for SUMT optimization; rdiv: reduction factor for a value;
c          tol: allowable differencpe in successive convergencpe values
          read(in,*) (mid(i), i=1,np)
          write(io,*) "flags for max. abs. value minimization",(mid(i), i=1,np)
          read(in,*) (wm(i), i=1,np)
          write(io,*) "weights for max. abs. value minimization",(wm(i), i=1,np)
          read(in,*) r,rdiv,tol
          write(io,*) "r value for SUMT optimization,rdiv,tol: ",r,rdiv,tol
        endif
        nnp=n
81      nfeval=0
        ncount=0
c
c        *****     generating an initial simplex     *****
c        lamda: guess of the problem's characteristic length scale
c        p(1,*): initial starting point
        read(in,*) (lamda(i),i=1,np)
        write(io,*) "length scale:lamda(*)=",(lamda(i),i=1,np)
        do 55 i=1,np
          p(1,i)=x(i)
55      continue
c        *****  creat np-dimensional unit vector  *****
        do 4 i=1,np
          do 4 j=1,np
            if (i.eq.j) then
              e(i,j)=1
            else
              e(i,j)=0
```

```
          endif
4       continue
c       ***** computing np other points and defining an initial simplex *****
83      do 11 i=2,mp
          do 11 j=1,np
            p(i,j)=p(1,j)+lamda(j)*e(i-1,j)
11      continue
c       ***** computing function value vector y(np+1) *****
        do 31 i=1,mp
         do 41 j=1,np
           t(j)=p(i,j)
41       continue
         y(i)=funk(t)
31      continue
c       ***** call subroutine "amoeba" for flexible polyhedron optimization *****
        call amoeba(p,y,mp,np,ndim,ftol,ptol,funk,iter,itmax,funk1,pp1,objv,obja)
c       ***** if mt=1(ie. use least pth only), go to get final output, otherwise go SUMT *****
        if (mt.eq.1) goto 1001
c       ***** applying SUMT minimization method *****
        write(io,*) "absolute value of (funk-funk1)=",abs(fn-funk1)
        if (abs(fn-funk1).lt.tol) goto 1001
        r=r/rdiv
        write(io,*) "r value for SUMT optimization:rvalue=",r
        do 22 j=1,np
          p(1,j)=pp1(j)
          lamda(j)=lamda(j)/2
22      continue
        goto 83
c       ***** generating final output files *****
1001    write(io,*) "the no. of iteration:",iter
        write(io,*) "*********** final optimization results *************"
        write(io,*) "the optimum values l(*)=",(l(i),i=1,novar)
        write(io,*) "the optimum function values:",fn
        write(io,*) "  no. of function eval.=",nfeval
95      if (iobj.eq.7.or.iobj.eq.8) call kinetost(1,res,nnp,bb,iparcel)
        do 99 i=1,nnp
            write(17,*) res(1,i),res(2,i)-csr0(1)
            write(18,*) res(1,i),res(8,i)-csr0(3)
            write(19,*) res(1,i),res(11,i)-csr0(4)
            write(22,*) res(1,i),res(3,i)
            write(23,*) res(1,i),res(4,i)
            write(24,*) res(1,i),res(6,i)
            write(25,*) res(1,i),res(7,i)
            write(26,*) res(1,i),res(9,i)
            write(27,*) res(1,i),res(10,i)
            write(28,*) res(1,i),res(12,i)
            write(29,*) res(1,i),res(13,i)
            write(30,*) res(1,i),transang(i)*360/6.28
            write(31,*) res(1,i),bb(3,i)/10000
            write(37,*) res(1,i),sqrt(res(10,i)**2+res(13,i)**2)
99      continue
        if (nf1.eq.1) goto 1000
c         ***** generate a data file for four-bar simulation *****
        call fbarkine(1,ncr)
        goto 1000
c       ***** straight analysis with no synthesis (opimization) *****
90      nnp=n
        call fbarkine(1,ncr)
        if (iobj.eq.7.or.iobj.eq.8) call kinetost(1,res,nnp,bb,iparcel)
        nf1=1
```

```
      goto 95
1000  write(*,*) "Do you want to restart the search process(1=y,else=n):"
      read(*,*) nrestart
      if (nrestart.ne.1) goto 1002
      write(io,*) "function values of all the vertices", (y(j),j=1,mp)
      write(*,*) "Please input new scale length of lamda:"
      read(*,*) scale
      do 91 j=1,np
        p(1,j)=pp1(j)
        lamda(j)=lamda(j)*scale
91    continue
      goto 83
1002  stop
      end
c***************************************************************************
c      objective function for linkage mechanism optimization program      *
c        - calculation of the objective function for least pth and SUMT   *
c***************************************************************************
      function funk(x)
      parameter (novar=8,lclu=2,nocsr=9,nopt=181,nores=21,n1=9)
      common /iout/in,io,nfeval,nsi,m1,np,ncr
      common /iout1/r
      common /inilen/lc11,lc13,lc14,lc21,lc23,lc24,linkno,iparcel
      common /fct/lc,csr,w,wm,res,w2,csr0,l,fn,funk1,transang,objv,obja
      common /fct1/mc,mid,nan,ncp,ml,nk,nnp,pp,p1,lmax,iobj,mm,nn,mm1,nn1
      real l(novar),lc(lclu,novar)
      dimension csr(nocsr,nopt),w(nocsr,nopt),wm(nocsr),res(nores,nopt)
     &          ,x(novar),csr0(4),el(150),eu(150),a1(novar),bb(n1,nopt)
      real lc11,lc13,lc14,lc21,lc23,lc24,transang(nopt)
      integer mc(4),ml(novar),mid(novar),p1,lmax,pp
      pi=3.1415926d0
      dr=180./pi
c      *****  if nan=0, go kinematic analysis directly  *****
      if (nan.eq.0) goto 10
c      *****  unconstrained x values are converted to constrained mechanism parameters  *****
      do 1 i=1,nk
        a1(i)=1/(exp(-x(i))+1)
1     continue
      n=4
      if (ncp.eq.1) n=np
c
c      evaluation of grashof's criterion, mechanism must be capable of full crank rotation
      l(2)=lc(1,2)+a1(2)*(lc(2,2)-lc(1,2))
      call gtrans(a1)
      do 2 i=5,n
        l(i)=lc(1,i)+a1(i)*(lc(2,i)-lc(1,i))
2     continue
c      *****  evaluation of four-bar kinematics  *****
10    call fbarkine(0,ncr)
c      *****  if nan=0, go kinematic analysis directly  *****
      if (nan.eq.0) goto 95
c
c      *** objective function evaluation for optimization routine  ***
c
      em=-100000.
      k=0
      funk=0
c
c      motion constraint violation values, and maximum violation
c      are determined in the following block
```

```
c
        do 40 i=1,4
          if (mc(i).eq.0) goto 40
          do 35 j=1,nnp
            k=k+1
            if (nan.gt.1) goto 32
c                ***** least pth constraint evaluation *****
            el(k)=(res(3*i-1,j)-csr(i*2,j)-csr0(i))*w(i*2-1,j)
            eu(k)=(res(3*i-1,j)-csr(i*2+1,j)-csr0(i))*w(i*2,j)
            if (-el(k).gt.em) em=-el(k)
            if (eu(k).gt.em) em=eu(k)
            goto 35
c                *****  SUMT motion constraint evaluation  *****
32          el(k)=1./(res(3*i-1,j)-csr(i*2,j)-csr0(i))*w(i*2-1,j)
            eu(k)=1./(-res(3*i-1,j)+csr(i*2+1,j)+csr0(i))*w(i*2,j)
            if ((el(k).gt.0).and.(eu(k).gt.0)) goto 35
c                *****  constraint violated in SUMT optimization  *****
              funk=10000.
              write(io,*) "**** CONSTRAINT VIOLATED ** MECHANISM REJECTED *****"
              goto 100
35        continue
40      continue
        if (nan.gt.1) goto 75
c
c        *****  objective function evaluation for least pth search  *****
        q=pp*em/abs(em)
        if (em.lt.0) goto 60
c
c        following block is used if any of the motion constraints
c        were violated    (least pth analysis)
        do 45 j=1,k
          if (el(j).lt.0) funk=funk+(-el(j)/em)**q
          if (eu(j).gt.0) funk=funk+(eu(j)/em)**q
45      continue
        goto 70
c
c        the following block is utilized if all constraints are
c        satisfied  (least pth analysis)
60      do 65 j=1,k
          funk=funk+(-el(j)/em)**q+(eu(j)/em)**q
65      continue
c
70      funk=em*funk**(1./q)
        goto 95
c        *****  objective function evaluation for SUMT analysis  *****
75      do 76 i=1,k
          funk=funk+el(i)+eu(i)
76      continue
        funk=funk*r
        write(io,*) "funk after motion constraints:",funk
c
c        *****  user specified optimization functions  *****
c        if iobj=1, objective is to minimize link dimension
c
        if (iobj.eq.1) then
          val=0
          do 85 ilink=1,4
            if(linkno.eq.ilink) then
              val=l(ilink)
            endif
```

```
85        continue
          funk1=val*wm(1)
          write(io,*) "funk1 - value of link dimension:",funk1
          funk=funk+funk1
c        ***** if iobj=2, objective is to maximize transmission angle only  *****
          elseif (iobj.eq.2) then
            trmin=acos((l(3)**2+l(4)**2-(l(1)-l(2))**2)/(2*l(3)*l(4)))
            trmax=acos((l(3)**2+l(4)**2-(l(1)+l(2))**2)/(2*l(3)*l(4)))
            if(trmin.gt.pi/2) trmin=pi-trmin
            if(trmax.gt.pi/2) trmax=pi-trmax
            if(trmin.lt.trmax) then
              trans1=1/abs(trmin)
            else
              trans1=1/abs(trmax)
            endif
            funk1=trans1*wm(1)
            write(io,*) "funk1 - inverse of transmission angle:",funk1
            funk=funk+funk1
        ***** if iobj=3, objective is to maximize the potential grashof mechanism  *****
            elseif (iobj.eq.3) then
              val=1/((l(3)+l(4)-l(1)-l(2))/100)
              funk1=val*wm(1)
              write(io,*) "funk1 - inverse of grashof criteria:",funk1
              funk=funk+funk1
c
c        if iobj=4, objective is to minimize max. abs. values of vel./acc.
c        during certain periods (mm,nn)
              elseif (iobj.eq.4) then
                vmax=0.
                do 80 i=1,8
                  if (mid(i).ne.1) goto 80
                    n=2
                    if (i.gt.2) n=3
                    if (i.gt.4) n=4
                    if (i.gt.6) n=5
                    do 79 j=mm,nn
                      if (abs(res(n+i,j)).gt.vmax) vmax=abs(res(n+i,j))
79                  continue
                    funk1=vmax*wm(i)
                    funk=funk+funk1
80              continue
                write(io,*) "funk1 -- max. abs. values of vel./acc.:",funk1
c        ***** if iobj=5, objective is to minimize normal of vel. only  *****
                  elseif (iobj.eq.5) then
                  vmax=0.
                    do 77 j=mm,nn
                       v=sqrt(res(9,j)**2+res(12,j)**2)
                       if (v.gt.vmax) vmax=v
77                  continue
                    funk1=vmax*wm(1)
                    funk=funk+funk1
                    write(io,*) "funk1 -- normal of vel.:",funk1
c
c        if iobj=6, objective is to minimize max. abs. values of vel./acc.
c        during certain periods (mm,nn) and max. abs. vel. at (mm1,nn1)
              elseif (iobj.eq.6) then
                  vmax=0.
                  vmax1=0.
                    do 72 j=mm,nn
                      if (abs(res(10,j)).gt.vmax) vmax=abs(res(10,j))
```

```
                 obja=vmax*wm(1)
72             continue
               do 89 j=mm1,nn1
                 if (abs(res(9,j)).gt.vmax1) vmax1=abs(res(9,j))
                 objv=vmax1*wm(2)
89             continue
               funk1=objv+obja
               funk=funk+funk1
               write(io,*) "objv -- max. abs. values of vel..:",objv
               write(io,*) "obja -- max. abs. values of acc.:",obja
c       *****  if iobj=7, objective is to minimize input torque only  *****
             elseif (iobj.eq.7) then
             tormax=0.
             call kinetost(l,res,nnp,bb,iparcel)
             do 78 j=mm,nn
               if (bb(3,j).gt.tormax) tormax=bb(3,j)
78           continue
             funk1=tormax*wm(1)
             funk=funk+funk1
             write(io,*) "funk1 --- input torque:",funk1
c
c       if iobj=8, objective is to minimize input torque and to maximize transmission angle
c
             elseif (iobj.eq.8) then
             tormax=0.
             amax=0.
             call kinetost(l,res,nnp,bb,iparcel)
             do 73 j=mm1,nn1
               if (bb(3,j).gt.tormax) tormax=bb(3,j)
73           continue
             objv=tormax*wm(1)
             write(io,*) "objv --- input torque:",objv
             trmin=acos((l(3)**2+l(4)**2-(l(1)-l(2))**2)/(2*l(3)*l(4)))
             trmax=acos((l(3)**2+l(4)**2-(l(1)+l(2))**2)/(2*l(3)*l(4)))
             if(trmin.gt.pi/2) trmin=pi-trmin
             if(trmax.gt.pi/2) trmax=pi-trmax
             if(trmin.lt.trmax) then
               trans1=1/abs(trmin)
             else
               trans1=1/abs(trmax)
             endif
             obja=trans1*wm(2)
             write(io,*) "obja - inverse of transmission angle:",obja
             funk1=obja+objv
             funk=funk+funk1
             write(io,*) "funk1 --- combination of obja & objv:",funk1
c       *****  if iobj=9, objective is to minimize normal of acc. only  *****
             elseif (iobj.eq.9) then
             amax=0.
             do 88 j=mm,nn
               noracc=sqrt(res(10,j)**2+res(13,j)**2)
               if (noracc.gt.amax) amax=noracc
88           continue
             funk1=amax*wm(1)
             funk=funk+funk1
             write(io,*) "funk1 -- normal of acc.:",funk1
c
c       if iobj=11, objective is to minimize the maximal length of l1
c       and maximize minimal transmission angle
             elseif (iobj.eq.11) then
```

```
                  val=0
                  do 81 ilink=1,4
                    if(linkno.eq.ilink) then
                      val=l(ilink)
                    endif
81                continue
                  objv=val*wm(1)
                  write(io,*) "objv - the maximal length of l1:",objv
                  trmin=acos((l(3)**2+l(4)**2-(l(1)-l(2))**2)/(2*l(3)*l(4)))
                  trmax=acos((l(3)**2+l(4)**2-(l(1)+l(2))**2)/(2*l(3)*l(4)))
                  if(trmin.gt.pi/2) trmin=pi-trmin
                  if(trmax.gt.pi/2) trmax=pi-trmax
                  if(trmin.lt.trmax) then
                    trans1=1/abs(trmin)
                  else
                    trans1=1/abs(trmax)
                  endif
                  obja=trans1*wm(2)
                  write(io,*) "obja - inverse of transmission angle:",obja
                  funk1=obja+objv
                  funk=funk+funk1
                  write(io,*) "funk1 --- combination of obja & objv:",funk1
                else
        endif
95      fn=funk
        nfeval=nfeval+1
100     write(io,*) "l(*)=",(l(i),i=1,nk)
1000    return
        end
c*****************************************************************
c        function for evaluation of four-bar kinematics          *
c*****************************************************************
        subroutine fbarkine(nsi,ncr)
        parameter (novar=8,lclu=2,nocsr=9,nopt=181,nores=21)
        common /fct/lc,csr,w,wm,res,w2,csr0,l,fn,funk1,transang,objv,obja
        common /fct1/mc,mid,nan,ncp,ml,nk,nnp,pp,p1,lmax,iobj,mm,nn,mm1,nn1
        common /distance/rg2,rg3,rg4,R2,R3,R4,RAG3x,RAG3y,RAG3,alpha3
     &              ,RAG3xx,RAG3yy,RAG33,alph33
        common /mass/mass2,mass3,mass4,mass6,masscp
        real l(novar),lc(lclu,novar),mass2,mass3,mass4,mass6,masscp
        dimension csr(nocsr,nopt),w(nocsr,nopt),wm(nocsr),res(nores,nopt)
     &              ,csr0(4),transang(nopt)
        integer mc(4),ml(novar),mid(novar),p1,lmax,pp
        pi=3.1415926
        dr=180./pi
        c0=0.
        if (iobj.eq.7.or.iobj.eq.8) then
c        ***** mass of links (unit: kg); density (unit: kg/cm)  *****
        density=0.05
        mass2=density*l(2)
        mass3=density*l(3)
        mass4=density*l(4)
        mass6=density*l(6)
        masscp=50
c        ***** center of mass        (unit: cm)  *****
        rg2=l(2)/2
        rg3=l(3)/2
        rg4=l(4)/2
        R2=l(2)-rg2
        R3=l(3)-rg3
```

```
        R4=l(4)-rg4
        RAG3xx=(mass3*rg3+mass6*l(5))/(mass3+mass6)
        RAG3yy=(mass6*l(6)/2)/(mass3+mass6)
        RAG33=sqrt(RAG3xx**2+RAG3yy**2)
        alph33=atan2(RAG3yy,RAG3xx)
        RAG3x=(mass3*rg3+mass6*l(5)+masscp*l(5))/(mass3+mass6+masscp)
        RAG3y=(masscp*l(6)+mass6*l(6)/2)/(mass3+mass6+masscp)
        RAG3=sqrt(RAG3x**2+RAG3y**2)
        alpha3=atan2(RAG3y,RAG3x)
        endif
c        ***** solving for the four-bar displacements *****
        do 15 i=1,nnp
          th2=csr(1,i)/dr
          th20=th2+l(7)/dr
          c2=cos(th20)
          s2=sin(th20)
          term1=l(4)**2-l(1)**2-l(2)**2-l(3)**2
          term2=l(1)**2+l(2)**2-l(3)**2+l(4)**2
          a=l(1)/l(3)*c2+c2+term1/(2*l(2)*l(3))-l(1)/l(2)
          b=-2*s2
          c=l(1)/l(3)*c2-c2+term1/(2*l(2)*l(3))+l(1)/l(2)
          d=c2+term2/(2*l(2)*l(4))-l(1)/l(2)-l(1)/l(4)*c2
          e=l(1)/l(2)+term2/(2*l(2)*l(4))-(1+l(1)/l(4))*c2
          sq3=sqrt(b*b-4*a*c)
          sq4=sqrt(b*b-4*d*e)
          if (ncr.eq.1) then
            th3=2*atan((-b+sq3)/(2*a))
            th4=2*atan((-b+sq4)/(2*d))
          else
            th3=2*atan((-b-sq3)/(2*a))
            th4=2*atan((-b-sq4)/(2*d))
          endif
c        ***** if there is no coupler point, then ncp=0 *****
          if (ncp.eq.0) goto 12
          c3=cos(th3)
          s3=sin(th3)
          if (iobj.eq.7.or.iobj.eq.8) then
c        ***** alpha3 ---- angle of coupler triangle *****
            cm3=cos(th3+alpha3)
            sm3=sin(th3+alpha3)
            cm33=cos(th3+alph33)
            sm33=sin(th3+alph33)
          endif
          c4=cos(th4)
          s4=sin(th4)
          xl=l(2)*c2+l(5)*c3-l(6)*s3
          yl=l(2)*s2+l(5)*s3+l(6)*c3
c         ***** solving for the velocities and acceleration *****
12        s24=sin(th20-th4)
          s23=sin(th20-th3)
          s43=sin(th4-th3)
          s34=sin(th3-th4)
          c24=cos(th20-th4)
          c23=cos(th20-th3)
          c34=cos(th3-th4)
          c24=cos(th20-th4)
c        ***** angular vel. and acc. of link 3 and 4 *****
          th3d=l(2)*w2*s24/(l(3)*s43)
          th4d=l(2)*w2*s23/(l(4)*s43)
          th3dd=(l(4)*th4d**2-l(2)*w2**2*c24-l(3)*th3d**2*c34)/(l(3)*s34)
```

```
          th4dd=(l(3)*th3d**2+l(2)*w2**2*c23-l(4)*th4d**2*c34)/(l(4)*s43)
          if (iobj.eq.7.or.iobj.eq.8) then
c        ***** velocity and acceleration of the center of link mass *****
            xlg2d=-l(2)*w2*s2/2
            ylg2d=l(2)*w2*c2/2
            xlg2dd=-l(2)*w2**2*c2/2
            ylg2dd=-l(2)*w2**2*s2/2
            xlg4d=-l(4)*th4d*s4/2
            ylg4d=l(4)*th4d*c4/2
            xlg4dd=-l(4)*th4d**2*c4/2-l(4)*th4dd*s4/2
            ylg4dd=-l(4)*th4d**2*s4/2+l(4)*th4dd*c4/2
c                ***** for center of mass of link 3 and link 6 *****
            xlg3d=-RAG33*th3d*sm33-l(2)*w2*s2
            ylg3d=RAG33*th3d*cm33+l(2)*w2*c2
            xlg3dd=-RAG33*th3d**2*cm33-RAG33*th3dd*sm33-l(2)*w2**2*c2
            ylg3dd=-RAG33*th3d**2*sm33+RAG33*th3dd*cm33-l(2)*w2**2*s2
c              ***** for center of mass of link 3 and mass on the coupler *****
            xgl3d=-RAG3*th3d*sm3-l(2)*w2*s2
            ygl3d=RAG3*th3d*cm3+l(2)*w2*c2
            xgl3dd=-RAG3*th3d**2*cm3-RAG3*th3dd*sm3-l(2)*w2**2*c2
            ygl3dd=-RAG3*th3d**2*sm3+RAG3*th3dd*cm3-l(2)*w2**2*s2
          endif
c        ***** if there is no coupler point, then ncp=0 *****
          if (ncp.eq.0) goto 14
c        ***** velocity and acceleration of the coupler point *****
          xld=-l(2)*w2*s2-l(5)*th3d*s3-l(6)*th3d*c3
          yld=l(2)*w2*c2+l(5)*th3d*c3-l(6)*th3d*s3
          xldd=-l(2)*w2**2*c2-l(5)*(th3dd*s3+th3d**2*c3)-l(6)*(th3dd*c3-th3d**2*s3)
          yldd=-l(2)*w2**2*s2+l(5)*(th3dd*c3-th3d**2*s3)-l(6)*(th3dd*s3+th3d**2*c3)
14        res(1,i)=th2*dr
          res(2,i)=th3*dr
          res(3,i)=th3d
          res(4,i)=th3dd
          res(5,i)=th4*dr
          res(6,i)=th4d
          res(7,i)=th4dd
          if (ncp.eq.0) goto 15
          sl=sin(l(8)/dr)
          cl=cos(l(8)/dr)
          res(8,i)=xl*cl-yl*sl
          res(9,i)=xld*cl-yld*sl
          res(10,i)=xldd*cl-yldd*sl
          res(11,i)=yl*cl+xl*sl
          res(12,i)=yld*cl+xld*sl
          res(13,i)=yldd*cl+xldd*sl
          if (iobj.eq.7.or.iobj.eq.8) then
c        ***** acceleration and its direction of center of mass for link 2-4 *****
            res(14,i)=sqrt((xlg2dd*cl-ylg2dd*sl)**2+(ylg2dd*cl+xlg2dd*sl)**2)
            res(15,i)=sqrt((xlg3dd*cl-ylg3dd*sl)**2+(ylg3dd*cl+xlg3dd*sl)**2)
            res(16,i)=sqrt((xlg4dd*cl-ylg4dd*sl)**2+(ylg4dd*cl+xlg4dd*sl)**2)
            res(17,i)=sqrt((xgl3dd*cl-ygl3dd*sl)**2+(ygl3dd*cl+xgl3dd*sl)**2)
            res(18,i)=atan2(ylg2dd,xlg2dd)
            res(19,i)=atan2(ylg3dd,xlg3dd)
            res(20,i)=atan2(ylg4dd,xlg4dd)
            res(21,i)=atan2(ygl3dd,xgl3dd)
          endif
          transang(i)=acos((l(3)**2+l(4)**2-l(1)**2-l(2)**2
     &                  +2*l(1)*l(2)*cos(res(1,i)/dr))/(2*l(3)*l(4)))
c        ***** if nsi=1, get a data file for simulation *****
16        if (nsi.ne.1) goto 15
```

```
          x1=l(2)*c2
          y1=l(2)*s2
          x2=l(2)*c2+l(3)*c3
          y2=l(2)*s2+l(3)*s3
          x3=l(2)*c2+l(5)*c3
          y3=l(2)*s2+l(5)*s3
          write(20,*) c0,c0,c0
          write(20,*) x1,y1,c0
          write(20,*) x2,y2,c0
          write(20,*) l(1),c0,c0
          write(20,*) x3,y3,c0
          write(20,*) x1,y1,c0
15      continue
        csr0(1)=res(2,1)
        csr0(2)=res(5,1)
        csr0(3)=res(8,1)
        csr0(4)=res(11,1)
50      return
        end
c****************************************************************************
c       subroutine for transformations based on redifined link length limits *
c****************************************************************************
        subroutine gtrans(a1)
        parameter (novar=8,lclu=2,nocsr=9,nopt=181,nores=21)
        common /inilen/lc11,lc13,lc14,lc21,lc23,lc24,linkno,iparcel
        common /fct/lc,csr,w,wm,res,w2,csr0,l,fn,funk1,transang,objv,obja
        common /fct1/mc,mid,nan,ncp,ml,nk,nnp,pp,p1,lmax,iobj,mm,nn,mm1,nn1
        real l(novar),lc(lclu,novar)
        dimension csr(nocsr,nopt),w(nocsr,nopt),wm(nocsr),res(nores,nopt)
     &,csr0(4),a1(novar)
        real lc11,lc13,lc14,lc21,lc23,lc24,transang(nopt)
        integer mc(4),ml(novar),mid(8),p1,lmax,pp
c        *****  when the longest link is l1  *****
        if(lmax.eq.1) then
c            *****  checking lower limit of l(1)  *****
          if(lc11.lt.l(2)) then
            lc(1,1)=l(2)
          else
            lc(1,1)=lc11
          endif
          l(1)=lc(1,1)+a1(1)*(lc(2,1)-lc(1,1))
c        *****  checking lower and upper limit of l(3)  *****
          if(lc13.lt.l(2)) then
            lc(1,3)=l(2)
          else
            lc(1,3)=lc13
          endif
          if(lc23.gt.l(1)) then
            lc(2,3)=l(1)
          else
            lc(2,3)=lc23
          endif
          l(3)=lc(1,3)+a1(3)*(lc(2,3)-lc(1,3))
c        *****  checking lower and upper limit of l(4)  *****
          if(lc14.lt.l(2)) then
            lc(1,4)=l(2)
          else
            lc(1,4)=lc14
          endif
          if(lc24.gt.l(1)) then
```

```
        lc(2,4)=l(1)
      else
        lc(2,4)=lc24
      endif
      if(lc(1,4).lt.(l(2)+l(1)-l(3))) lc(1,4)=l(2)+l(1)-l(3)
      if(lc(2,4).lt.lc(1,4)) then
        0ra3=lc23-lc13
        0ra4=lc24-lc14
        lc(1,3)=(0ra3*(l(2)+l(1)-lc(2,4))+0ra4*lc(2,3))/(0ra3+0ra4)
        lc(1,4)=(0ra3*lc(2,4)+0ra4*(l(2)+l(1)-lc(2,3)))/(0ra3+0ra4)
      endif
      l(3)=lc(1,3)+al(3)*(lc(2,3)-lc(1,3))
      l(4)=lc(1,4)+al(4)*(lc(2,4)-lc(1,4))
c     *****  when the longest link is 13  *****
      elseif(lmax.eq.2) then
c          ***** checking lower limit of l(3)  *****
            if(lc13.lt.l(2)) then
              lc(1,3)=l(2)
            else
              lc(1,3)=lc13
            endif
            l(3)=lc(1,3)+al(3)*(lc(2,3)-lc(1,3))
c     *****  checking lower and upper limit of l(1)  *****
            if(lc11.lt.l(2)) then
              lc(1,1)=l(2)
            else
              lc(1,1)=lc11
            endif
            if(lc21.gt.l(3)) then
              lc(2,1)=l(3)
            else
              lc(2,1)=lc21
            endif
            l(1)=lc(1,1)+al(1)*(lc(2,1)-lc(1,1))
c     *****  checking lower and upper limit of l(4)  *****
            if(lc14.lt.l(2)) then
              lc(1,4)=l(2)
            else
              lc(1,4)=lc14
            endif
            if(lc24.gt.l(3)) then
              lc(2,4)=l(3)
            else
              lc(2,4)=lc21
            endif
            if(lc(1,4).lt.(l(2)+l(3)-l(1))) lc(1,4)=l(2)+l(3)-l(1)
            if(lc(2,4).lt.lc(1,4)) then
               0ra1=lc21-lc11
              0ra4=lc24-lc14
              lc(1,1)=(0ra1*(l(2)+l(3)-lc(2,4))+0ra4*lc(2,1))/(0ra1+0ra4)
              lc(1,4)=(0ra1*lc(2,4)+0ra4*(l(2)+l(3)-lc(2,1)))/(0ra1+0ra4)
            endif
            l(1)=lc(1,1)+al(1)*(lc(2,1)-lc(1,1))
            l(4)=lc(1,4)+al(4)*(lc(2,4)-lc(1,4))
c     *****  when the longest link is 14  *****
      else
c         *****  checking lower and limit of l(4)  *****
              if(lc14.lt.l(2)) then
                lc(1,4)=l(2)
              else
```

```
              lc(1,4)=lc14
            endif
            l(4)=lc(1,4)+al(4)*(lc(2,4)-lc(1,4))
c        *****  checking lower and upper limit of l(3)  *****
            if(lc13.lt.l(2)) then
              lc(1,3)=l(2)
            else
              lc(1,3)=lc13
            endif
            if(lc23.gt.l(4)) then
              lc(2,3)=l(4)
            else
              lc(2,3)=lc23
            endif
            l(3)=lc(1,3)+al(3)*(lc(2,3)-lc(1,3))
c        *****  checking lower and upper limit of l(1)  *****
            if(lc11.lt.l(2)) then
              lc(1,1)=l(2)
            else
              lc(1,1)=lc11
            endif
            if(lc21.gt.l(4)) then
              lc(2,1)=l(4)
            else
              lc(2,1)=lc21
            endif
            if(lc(1,1).lt.(l(2)+l(4)-l(3))) lc(1,1)=l(2)+l(4)-l(3)
            if(lc(2,4).lt.lc(1,4)) then
              Ora1=lc21-lc11
              Ora3=lc23-lc13
              lc(1,3)=(Ora3*(l(2)+l(4)-lc(2,1))+Ora1*lc(2,3))/(Ora3+Ora1)
              lc(1,1)=(Ora3*lc(2,1)+Ora1*(l(2)+l(4)-lc(2,3)))/(Ora3+Ora1)
            endif
            l(1)=lc(1,1)+al(1)*(lc(2,1)-lc(1,1))
            l(3)=lc(1,3)+al(3)*(lc(2,3)-lc(1,3))
      endif
      return
      end
c**********************************************************************
c     subroutine for kinetostatic analysis of four-bar mechanism      *
c**********************************************************************
      subroutine kinetost(l,res,nnp,bb,iparcel)
      parameter (n1=9,np1=9,m1=1,mp1=1,novar=8,nopt=181,nores=21)
      common /distance/rg2,rg3,rg4,R2,R3,R4,RAG3x,RAG3y,RAG3,alpha3
     &              ,RAG3xx,RAG3yy,RAG33,alph33
      common /mass/mass2,mass3,mass4,mass6,masscp
      dimension a(np1,np1),b(n1,m1),res(nores,nopt),bb(n1,nopt),xy(n1,n1)
      real l(novar),I3,I4,I2,mass2,mass3,mass4,mass6,masscp,II3
      pi=3.1415926
c      *****  inertia acc.        (unit: cm/s^2)  *****
      g=981
      dr=180./pi
c      *****  inertia of link        (unit: kg cm^2)  *****
      I2=mass2*l(2)**2/12
      II3=mass3*l(3)**2/12+mass3*((RAG3xx-rg3)**2+RAG3yy**2)
     &              +mass6*((RAG3xx-l(5))**2+(RAG3yy-l(6)/2)**2)
      I3=mass3*l(3)**2/12+mass3*((RAG3x-rg3)**2+RAG3y**2)
     &        +mass6*((RAG3x-l(5))**2+(RAG3y-l(6)/2)**2)
     &        +masscp*((RAG3x-l(5))**2+(RAG3y-l(6))**2)
      I4=mass4*l(4)**2/12
```

```
c
c        calculating inertia force component and inertia moment of each link
c        Fo2x,Fo2y,To2,Fo3x,Fo3y,To3,Fo4x,Fo4y,To4 -- b(*,i)
c
      mass31=mass3+mass6+masscp
      mass32=mass3+mass6
      do 15 i=1,nnp
        bb(1,i)=-mass2*(res(14,i)*cos(res(18,i)+l(8)/dr))
        bb(2,i)=-mass2*(res(14,i)*sin(res(18,i)+l(8)/dr)+g)
        bb(3,i)=-I2*0
c        ***** when iparcel=1, run with parcel on the coupler; otherwise without parcel *****
      if((iparcel.eq.1).and.(i.le.16)) then
        bb(4,i)=-mass31*(res(17,i)*cos(res(21,i)+l(8)/dr))
        bb(5,i)=-mass31*(res(17,i)*sin(res(21,i)+l(8)/dr)+g)
        bb(6,i)=-I3*res(4,i)
      else
        bb(4,i)=-mass32*(res(15,i)*cos(res(19,i)+l(8)/dr))
        bb(5,i)=-mass32*(res(15,i)*sin(res(19,i)+l(8)/dr)+g)
        bb(6,i)=-II3*res(4,i)
      endif
        bb(7,i)=-mass4*(res(16,i)*cos(res(20,i)+l(8)/dr))
        bb(8,i)=-mass4*(res(16,i)*sin(res(20,i)+l(8)/dr)+g)
        bb(9,i)=-I4*res(7,i)
c        ***** Formation of Matrix a(np1,np1) *****
        c2=cos(res(1,i)/dr+l(7)/dr+l(8)/dr)
        s2=sin(res(1,i)/dr+l(7)/dr+l(8)/dr)
        c4=cos(res(5,i)/dr+l(8)/dr)
        s4=sin(res(5,i)/dr+l(8)/dr)
        data (a(1,j),j=1,n1)/-1.0000,2*0.0,1.0000,5*0.0/
        data (a(2,j),j=1,n1)/0.0,-1.0000,2*0.0,1.0000,4*0.0/
        data a(3,3),(a(3,j),j=6,n1)/-1.0000,4*0.0/
        data (a(4,j),j=1,n1)/3*0.0,-1.0000,0,1.0000,3*0.0/
        data (a(5,j),j=1,n1)/4*0.0,-1.000,0.0,1.0000,2*0.0/
        data (a(6,j),j=1,3),a(6,8),a(6,9)/5*0.0/
        data (a(7,j),j=1,n1)/5*0.0,-1.0000,0.0,-1.0000,0.0/
        data (a(8,j),j=1,n1)/6*0.0,-1.0000,0.0,-1.0000/
        data (a(9,j),j=1,5)/5*0.0/
        a(3,1)=-rg2*s2
        a(3,2)=rg2*c2
        a(3,4)=-R2*s2
        a(3,5)=R2*c2
c        ***** when iparcel=1, run with parcel on the coupler; otherwise without parcel *****
      if((iparcel.eq.1).and.(i.le.16)) then
        c3=cos(res(2,i)/dr+l(8)/dr)
        s3=sin(res(2,i)/dr+l(8)/dr)
        cc3=cos(res(2,i)/dr+l(8)/dr+alpha3)
        ss3=sin(res(2,i)/dr+l(8)/dr+alpha3)
        a(6,4)=-RAG3*ss3
        a(6,5)=RAG3*cc3
        a(6,6)=-(l(3)*s3-RAG3*ss3)
        a(6,7)=(l(3)*c3-RAG3*cc3)
      else
        c3=cos(res(2,i)/dr+l(8)/dr)
        s3=sin(res(2,i)/dr+l(8)/dr)
        cc3=cos(res(2,i)/dr+l(8)/dr+alph33)
        ss3=sin(res(2,i)/dr+l(8)/dr+alph33)
        a(6,4)=-RAG33*ss3
        a(6,5)=RAG33*cc3
        a(6,6)=-(l(3)*s3-RAG33*ss3)
        a(6,7)=(l(3)*c3-RAG33*cc3)
```

```
        endif
          a(9,6)=R4*s4
          a(9,7)=-R4*c4
          a(9,8)=-rg4*s4
          a(9,9)=rg4*c4
          do 25 j=1,n1
            b(j,m1)=bb(j,i)
25        continue
        do 20 k=1,n1
          do 20 j=1,n1
            xy(k,j)=a(k,j)
20      continue
        call gaussj(a,n1,np1,b,m1,mp1)
        do 21 k=1,n1
          do 21 j=1,n1
            a(k,j)=xy(k,j)
21      continue
          do 35 j=1,n1
            bb(j,i)=b(j,m1)
35        continue
15      continue
        return
        end
c***************************************************************************
c     subroutine of matrix manipulation with Gauss-Jordan method          *
c***************************************************************************
      SUBROUTINE GAUSSJ(A,N,NP,B,M,MP)
      PARAMETER (NMAX=50)
      DIMENSION A(NP,NP),B(NP,MP),IPIV(NMAX),INDXR(NMAX),INDXC(NMAX)
      DO 11 J=1,N
        IPIV(J)=0
11    CONTINUE
      DO 22 I=1,N
        BIG=0.
        DO 13 J=1,N
          IF(IPIV(J).NE.1)THEN
            DO 12 K=1,N
              IF (IPIV(K).EQ.0) THEN
                IF (ABS(A(J,K)).GE.BIG)THEN
                  BIG=ABS(A(J,K))
                  IROW=J
                  ICOL=K
                ENDIF
              ELSE IF (IPIV(K).GT.1) THEN
                PAUSE 'Singular matrix'
              ENDIF
12          CONTINUE
          ENDIF
13      CONTINUE
        IPIV(ICOL)=IPIV(ICOL)+1
        IF (IROW.NE.ICOL) THEN
          DO 14 L=1,N
            DUM=A(IROW,L)
            A(IROW,L)=A(ICOL,L)
            A(ICOL,L)=DUM
14        CONTINUE
          DO 15 L=1,M
            DUM=B(IROW,L)
            B(IROW,L)=B(ICOL,L)
            B(ICOL,L)=DUM
```

```
15          CONTINUE
          ENDIF
          INDXR(I)=IROW
          INDXC(I)=ICOL
          IF (A(ICOL,ICOL).EQ.0.) PAUSE 'Singular matrix.'
          PIVINV=1./A(ICOL,ICOL)
          A(ICOL,ICOL)=1.
          DO 16 L=1,N
            A(ICOL,L)=A(ICOL,L)*PIVINV
16        CONTINUE
          DO 17 L=1,M
            B(ICOL,L)=B(ICOL,L)*PIVINV
17        CONTINUE
          DO 21 LL=1,N
            IF(LL.NE.ICOL)THEN
              DUM=A(LL,ICOL)
              A(LL,ICOL)=0.
              DO 18 L=1,N
                A(LL,L)=A(LL,L)-A(ICOL,L)*DUM
18            CONTINUE
              DO 19 L=1,M
                B(LL,L)=B(LL,L)-B(ICOL,L)*DUM
19            CONTINUE
            ENDIF
21        CONTINUE
22      CONTINUE
        DO 24 L=N,1,-1
          IF(INDXR(L).NE.INDXC(L))THEN
            DO 23 K=1,N
              DUM=A(K,INDXR(L))
              A(K,INDXR(L))=A(K,INDXC(L))
              A(K,INDXC(L))=DUM
23          CONTINUE
          ENDIF
24      CONTINUE
        RETURN
        END

c***************************************************************************
c         subroutine for flexible polyhedron search technique             *
c***************************************************************************
          subroutine amoeba(p,y,mp,np,ndim,ftol,ptol,funk,iter,itmax,funk1,pp1,objv,obja)
c
c          this is flexible polyhedron method
c          funk(x): function of m-d minimization where x is ndim vector
c          p: logical dimensions p(ndim+1,ndim);input/output
c          p: phisical dimensions p(mp,np); input/output
c          y: input/output, function value vector y(np+1)
c          ftol: fractional convergence tolerance;input
c          iter: number of iteration;output
c
          parameter (nmax=20,alpha=1.0,beta=.5,gamma=2.0)
          dimension p(mp,np),y(mp),pr(nmax),prr(nmax),pbar(nmax),pp1(np)
          in=45
          io=46
          mpts=ndim+1
1         ilo=1
c         *** determine which point in the highest(worst),next highest and lowerest ***
          if(y(1).gt.y(2)) then
            ihi=1
```

```
          inhi=2
        else
          ihi=2
          inhi=1
        endif
c
c        by looping over the points in the simplex
c
        do 11 i=1,mpts
          if(y(i).lt.y(ilo)) ilo=i
          if(y(i).gt.y(ihi)) then
            inhi=ihi
            ihi=i
          else if(y(i).gt.y(inhi)) then
            if(i.ne.ihi) inhi=i
          endif
11      continue
c        compute the fractional range from the highest to lowest and return if satisfactory
        rtol=abs(y(ihi)-y(ilo))
        fptol=0
        do 31 i=1,mpts
          if(i.ne.ilo) then
            do 32 j=1,ndim
                fptol=fptol+(p(ilo,j)-p(i,j))**2
32          continue
          endif
31        continue
        do 33 j=1,ndim
           pp1(j)=p(ilo,j)
33      continue
        iter=iter+1
        write(io,*) "iteration no.",iter,";  rtol=",rtol,";  fptol=",fptol
        write(io,*) "current function value y(ilo)=",y(ilo)
        write(33,*) iter,y(ilo)
        write(32,*) iter,funk1
        write(34,*) iter,objv
        write(35,*) iter,obja
        if(fptol.lt.ptol) return
        if(iter.eq.itmax) pause 'amoeba exceeding maximum iterations.'
        if(y(ilo).lt.0) write(io,*) "all of the displacement constraints are satisfied"
        write(io,*) "function values of all the vertices", (y(j),j=1,mp)
        write(io,*) "-------------------------------------------------"
        do 12 j=1,ndim
          pbar(j)=0.
12      continue
c
c        begin a new iteration. compute the vector average of all points
c        except the highest, i.e. the center of the "face"of the simplex
c        across from the high point. we will subsequently explore along
c        the ray from the hight point through the center.
c
        do 14 i=1,mpts
          if(i.ne.ihi) then
            do 13 j=1,ndim
              pbar(j)=pbar(j)+p(i,j)
13          continue
          endif
14      continue
c        extrapolate by alpha through the face, i.e. reflect the simplex from the high point
        do 15 j=1,ndim
```

```
          pbar(j)=pbar(j)/ndim
          pr(j)=(1.+alpha)*pbar(j)-alpha*p(ihi,j)
15      continue
c        evaluate the function at the reflected point. give a result better
c        than the best point, so try an additional extrapolation by GAMMA
        ypr=funk(pr)
        if(ypr.lt.y(ilo)) then
          do 16 j=1,ndim
            prr(j)=gamma*pr(j)+(1.-gamma)*pbar(j)
16        continue
c        check out function. the additional extrapolation succeeded, and replaces the high point
          yprr=funk(prr)
          if(yprr.lt.y(ilo)) then
            do 17 j=1,ndim
              p(ihi,j)=prr(j)
17          continue
            y(ihi)=yprr
          else
c        the additional extrapolation failed, but we can still use the reflected point
            do 18 j=1,ndim
                p(ihi,j)=pr(j)
18          continue
            y(ihi)=ypr
          endif
c        the reflected point is worse than the second-highest.
c        if it's better than the highest, then replace the highest,
        else if(ypr.gt.y(inhi)) then
          if(ypr.lt.y(ihi)) then
            do 19 j=1,ndim
                p(ihi,j)=pr(j)
19          continue
            y(ihi)=ypr
          endif
c        but look for an intermediate lower point, in other words, perform
c        a contraction of the simplex along one dimension. then evaluate the function.
          do 21 j=1,ndim
            prr(j)=beta*p(ihi,j)+(1.-beta)*pbar(j)
21        continue
c        *****  contraction gives an improvement, so accept it.  *****
          yprr=funk(prr)
          if(yprr.lt.y(ihi)) then
            do 22 j=1,ndim
                p(ihi,j)=prr(j)
22          continue
            y(ihi)=yprr
          else
c        can't seem to get rid of that hight point. better contract around the lowest (best) point.
            do 24 i=1,mpts
                    if(i.ne.ilo) then
                  do 23 j=1,ndim
                    pr(j)=0.5*(p(i,j)+p(ilo,j))
                    p(i,j)=pr(j)
23                continue
                y(i)=funk(pr)
                    endif
24         continue
           endif
        else
c        we arrive here if the original reflection gives a middling point.
c        replace the old high point and continue
```

```
          do 25 j=1,ndim
            p(ihi,j)=pr(j)
25        continue
            y(ihi)=ypr
        endif
c       ***** for the test of doneness and the next iteration.  *****
        go to 1
        end
c**************************************************************************
c       main program for six-bar linkage optimization                    *
c         - inout/output for developed optimization-based synthesis approach *
c**************************************************************************
        parameter (novar=14,lclu=2,nocsr=17,nopt=37,nores=25)
        common /iout/in,io,nfeval,nsi,m1,np,ncr,ncr1
        common /iout1/r
        common /inilen/lc11,lc13,lc14,lc21,lc23,lc24,lc19,lc29,lc110,lc210,linkno
        common /fct/lc,csr,w,wm,res,w2,csr0,l,fn,funk1,transang,objv,obja
        common /fct1/mc,mid,nan,ncp,ml,nk,nnp,pp,p1,lmax,iobj,mm,nn,mm1,nn1
        real l(novar),lc(lclu,novar),transang(nopt)
        dimension csr(nocsr,nopt),w(nocsr,nopt),wm(nocsr),res(nores,nopt),x(novar),csr0(8)
        real lc11,lc13,lc14,lc21,lc23,lc24,lamda(novar),lc110,lc210,lc19,lc29
        dimension p(novar+1,novar),y(novar+1),t(novar),e(novar,novar),pp1(novar)
        integer mc(8),ml(novar),mid(novar),p1,lmax,pp
        character*20 fname1,fname2,fname(30)
c        *****  Open input/output data file  *****
        in=45
        io=46
        iter=0
        write(*,*) "please input the name of the data file:"
        read(*,*) fname1
        write(*,*) "please input the name of the output-data file:"
        read(*,*) fname2
        write(*,*) "fname1=",fname1,"  fname2=",fname2
        open(in,file=fname1)
        open(io,file=fname2)
        open(10,file='fname.dat')
        read(10,*) (fname(i),i=1,27)
        do 901 i=1,26
          j=i+10
          open(j,file=fname(i))
901     continue
        read(in,*) nan,mech,ncp,incop,inc,w2,lmax,iobj,mm,nn,mm1,nn1
        write(io,*) "nan,mech,ncp,incop,inc,w2,lmax,iobj,mm,nn,mm1,nn1:"
     &             ,nan,mech,ncp,incop,inc,w2,lmax,iobj,mm,nn,mm1,nn1
c       ***  ncr,ncr1: flag to identify the cross link (0=n, 1=y)  ***
        read(in,*) ftol,itmax,ptol,nsi,ncr,ncr1,p1,linkno
        write(io,*) "ftol,itmax,ptol:,etc.",ftol,itmax,ptol,nsi,ncr,ncr1,p1,linkno
        read(in,*) mp,np,ndim
        write(io,*) "simplex vertices mp,search variables np,ndim:",mp,np,ndim
        nk=np
        read(in,*) (l(i),i=1,np)
        write(io,*) "initial variables l(*)=",(l(i),i=1,np)
        if (nan.eq.0) goto 3
        read(in,*) (ml(i),i=1,np)
        read(in,*) (lc(1,i),i=1,np)
        write(io,*) "lower constraints lc(1,*)=",(lc(1,i),i=1,np)
        read(in,*) (lc(2,i),i=1,np)
        write(io,*) "upper constraints lc(2,*)=",(lc(2,i),i=1,np)
c        *****  get some of initial constraints of the search variables  *****
        lc11=lc(1,1)
```

```
      lc13=lc(1,3)
      lc14=lc(1,4)
      lc21=lc(2,1)
      lc23=lc(2,3)
      lc24=lc(2,4)
      lc19=lc(1,9)
      lc29=lc(2,9)
      lc110=lc(1,10)
      lc210=lc(2,10)
      if(lmax.eq.1) then
        if(lc11.lt.l(2)) lc(1,1)=l(2)
        if(lc13.lt.l(2)) lc(1,3)=l(2)
        if(lc23.gt.l(1)) lc(2,3)=l(1)
        lc(1,4)=l(2)+l(1)-l(3)
        if(lc24.gt.l(1)) lc(2,4)=l(1)
      endif
3     if (incop.eq.1) goto 5
      n=inc
      read(in,*) (csr(1,i), i=1,n)
      write(io,*) "input angles deltath2=",(csr(1,i), i=1,n)
      goto 30
5     n=360/inc
      val=0.
      do 10 i=1,n+1
        csr(1,i)=val
        val=val+inc
10    continue
      n=n+1
30    nnp=n
c     ***** make sure that six-bar could be assembled together *****
      if (n.gt.9) then
        call fbarkine1(0,ncr,ncr1,nas,dmax,dmin)
        if(lc110.lt.abs(dmax-l(9))) lc(1,10)=abs(dmax-l(9))
        if(lc210.gt.(dmin+l(9))) lc(2,10)=dmin+l(9)
      endif
      do 7 i=1,np
        if (ml(i).eq.0) goto 7
        x(i)=alog((l(i)-lc(1,i))/(lc(2,i)-l(i)))
7     continue
      if (nan.eq.0) goto 90
      read(in,*) mt,nw
      write(io,*) "optimization type mt,constraint weighting flag nw:",mt,nw
      read(in,*) (mc(i),i=1,8)
      write(io,*) "flag for disp. to be constrained:mc(*)=",(mc(i),i=1,8)
      do 40 i=1,8
        if (mc(i).eq.0) goto 40
          read(in,*) (csr(i*2,j),j=1,n)
          write(io,*) "lower motion constraints csr(i*2,j)=",(csr(i*2,j),j=1,n)
          read(in,*) (csr(i*2+1,j),j=1,n)
          write(io,*) "upper motion constraints csr(i*2+1,j)=",(csr(i*2+1,j),j=1,n)
          if (nw.eq.0) goto 35
              do 34 j=1,n
                w(i*2-1,j)=1.
                w(i*2,j)=1.
c               w(*): weights for upper & lower constraints
34            continue
              goto 40
35        read(in,*) (w(i*2-1,j),j=1,n)
          write(io,*) "weights for lower constraints w(*):",(w(i*2-1,j),j=1,n)
          read(in,*) (w(i*2,j),j=1,n)
```

```
            write(io,*) "weights for upper constraints w(*):",(w(i*2,j),j=1,n)
40        continue
          do 51 j=1,n
            write(11,*) csr(1,j),csr(10,j)
            write(12,*) csr(1,j),csr(11,j)
            write(13,*) csr(1,j),csr(14,j)
            write(14,*) csr(1,j),csr(15,j)
            write(15,*) csr(1,j),csr(16,j)
            write(16,*) csr(1,j),csr(17,j)
51        continue
          j=14
          if (mt.eq.1) then
            read(in,*) pp
            write(io,*) "parameter for least pth:p=",pp
          else
            read(in,*) (mid(i), i=1,j)
            write(io,*) "flags for max. abs. value minimization",(mid(i), i=1,j)
            read(in,*) (wm(i), i=1,j)
            write(io,*) "weights for max. abs. value minimization",(wm(i), i=1,j)
            read(in,*) r,rdiv,tol
            write(io,*) "r value for SUMT optimization,rdiv,tol: ",r,rdiv,tol
          endif
8         nfeval=0
          ncount=0
c          *****    generating an initial simplex     *****
          read(in,*) (lamda(i),i=1,np)
          write(io,*) "length scale:lamda(*)=",(lamda(i),i=1,np)
          do 55 i=1,np
            p(1,i)=x(i)
55          continue
          do 4 i=1,np
            do 4 j=1,np
              if (i.eq.j) then
                e(i,j)=1
              else
                e(i,j)=0
              endif
4          continue
83         do 11 i=2,mp
              do 11 j=1,np
                p(i,j)=p(1,j)+lamda(j)*e(i-1,j)
11         continue
          do 31 i=1,mp
            do 41 j=1,np
              t(j)=p(i,j)
41          continue
            y(i)=funk(t)
31        continue
c          ***** call subroutine "amoeba" for flexible polyhedron optimization *****
          call amoeba(p,y,mp,np,ndim,ftol,ptol,funk,iter,itmax,funk1,pp1,objv,obja)
c          ***** if mt=1(ie. use least pth only), go to get final output, otherwise go SUMT *****
          if (mt.eq.1) goto 1001
c          ***** applying SUMT minimization method *****
          write(io,*) "absolute value of (funk-funk1)=",abs(fn-funk1)
          if (abs(fn-funk1).lt.tol) goto 1001
          r=r/rdiv
          write(io,*) "r value for SUMT optimization:rvalue=",r
          do 22 j=1,np
            p(1,j)=pp1(j)
            lamda(j)=lamda(j)/2
```

```
22      continue
        goto 83
c        ***** generating final output files *****
1001    write(io,*) "the no. of iteration:",iter
        write(io,*) "*********** final optimization results *************"
        write(io,*) "the optimum values l(*)=",(l(i),i=1,novar)
        write(io,*) "the optimum function values:",fn
        write(io,*) "  no. of function eval.=",nfeval
        do 99 i=1,nnp
            write(17,*) res(1,i),res(14,i)-csr0(5)
            write(18,*) res(1,i),res(20,i)-csr0(7)
            write(19,*) res(1,i),res(23,i)-csr0(8)
            write(22,*) res(1,i),res(2,i)
            write(23,*) res(1,i),res(5,i)
            write(24,*) res(1,i),res(15,i)
            write(25,*) res(1,i),res(7,i)
            write(26,*) res(1,i),res(9,i)
            write(27,*) res(1,i),res(10,i)
            write(28,*) res(1,i),res(12,i)
            write(29,*) res(1,i),res(13,i)
            write(30,*) res(1,i),transang(i)*360/6.28
99      continue
1000    write(*,*) "Do you want to restart the search process(1=y,else=n):"
        read(*,*) nrestart
        if (nrestart.ne.1) goto 1002
        write(io,*) "function values of all the vertices", (y(j),j=1,mp)
        write(*,*) "Please input new scale length of lamda:"
        read(*,*) scale
        do 91 j=1,np
          p(1,j)=pp1(j)
          lamda(j)=lamda(j)*scale
91      continue
        goto 83
1002    stop
        end
c************************************************************************
c        objective function for linkage mechanism optimization program      *
c          - calculation of the objective function for least pth and SUMT    *
c************************************************************************
        function funk(x)
        parameter (novar=14,lclu=2,nocsr=17,nopt=37,nores=25)
        common /iout/in,io,nfeval,nsi,m1,np,ncr,ncr1
        common /iout1/r
        common /inilen/lc11,lc13,lc14,lc21,lc23,lc24,lc19,lc29,lc110,lc210,linkno
        common /fct/lc,csr,w,wm,res,w2,csr0,l,fn,funk1,transang,objv,obja
        common /fct1/mc,mid,nan,ncp,ml,nk,nnp,pp,p1,lmax,iobj,mm,nn,mm1,nn1
        real l(novar),lc(lclu,novar),lmax1
        dimension csr(nocsr,nopt),w(nocsr,nopt),wm(nocsr),res(nores,nopt)
     &,x(novar),csr0(8),el(150),eu(150),a1(novar)
        real lc11,lc13,lc14,lc21,lc23,lc24,lc19,lc29,lc110,lc210,transang(nopt)
        integer mc(8),ml(novar),mid(novar),p1,lmax,pp
        pi=3.1415926d0
        dr=180./pi
        if (nan.eq.0) goto 10
        do 1 i=1,nk
          a1(i)=1/(exp(-x(i))+1)
1       continue
        n=4
        if (ncp.eq.1) n=np
c        evaluation of grashof's criterion, mechanism must be capable of full crank rotation
```

```
        l(2)=lc(1,2)+a1(2)*(lc(2,2)-lc(1,2))
        call gtrans(a1)
        do 2 i=5,n
          l(i)=lc(1,i)+a1(i)*(lc(2,i)-lc(1,i))
2       continue
        call fbarkine1(nsi,ncr,ncr1,nas,dmax,dmin)
        if(lc110.lt.abs(dmax-l(9))) lc(1,10)=abs(dmax-l(9))
        if(lc(1,10).lt.abs(l(9)-dmin)) lc(1,10)=abs(l(9)-dmin)
        if(lc210.gt.(dmin+l(9))) lc(2,10)=dmin+l(9)
c       ***** checking lower and upper limit of l(10) *****
        if(lc(1,10).gt.lc(2,10)) then
          Ora9=lc29-lc19
          Ora10=lc210-lc110
          denorm=2*(Ora10+Ora9)
          lc(1,9)=(Ora9*(dmax-dmin)+Ora10*(dmax+dmin))/denorm
          lc(1,10)=(Ora9*(dmax+dmin)+Ora10*(dmax-dmin))/denorm
          lc(2,9)=(Ora9*(dmax+3*dmin)+Ora10*(dmax+dmin))/denorm
          lc(2,10)=(Ora9*(dmax+dmin)+Ora10*(dmax+3*dmin))/denorm
        endif
        l(9)=lc(1,9)+a1(9)*(lc(2,9)-lc(1,9))
        l(10)=lc(1,10)+a1(10)*(lc(2,10)-lc(1,10))
c        *****  evaluation of four-bar kinematics  *****
        call fbarkine(0,ncr,ncr1,nas)
        if (nan.eq.0) goto 95
c        *** objective function evaluation for optimization routine  ***
        em=-100000.
        k=0
        funk=0
c
c        motion constraint violation values, and maximum violation
c        are determined in the following block
c
        do 40 i=1,8
          if (mc(i).eq.0) goto 40
          do 35 j=1,nnp
            k=k+1
            if (nan.gt.1) goto 32
c                ***** least pth constraint evaluation  *****
            el(k)=(res(3*i-1,j)-csr(i*2,j)-csr0(i))*w(i*2-1,j)
            eu(k)=(res(3*i-1,j)-csr(i*2+1,j)-csr0(i))*w(i*2,j)
            if (-el(k).gt.em) em=-el(k)
            if (eu(k).gt.em) em=eu(k)
            goto 35
c                *****  SUMT motion constraint evaluation  *****
32          el(k)=1./(res(3*i-1,j)-csr(i*2,j)-csr0(i))*w(i*2-1,j)
            eu(k)=1./(-res(3*i-1,j)+csr(i*2+1,j)+csr0(i))*w(i*2,j)
            if ((el(k).gt.0).and.(eu(k).gt.0)) goto 35
c                *****  constraint violated in SUMT optimization  *****
                funk=10000.
                write(io,*) "**** CONSTRAINT VIOLATED ** MECHANISM REJECTED *****"
                goto 100
35        continue
40      continue
c        *****  objective function evaluation for least pth search  *****
        q=pp*em/abs(em)
        if (em.lt.0) goto 60
c        following block is used if any of the motion constraints were violated
        do 45 j=1,k
          if (el(j).lt.0) funk=funk+(-el(j)/em)**q
          if (eu(j).gt.0) funk=funk+(eu(j)/em)**q
```

```
45      continue
        goto 70
c        the following block is utilized if all constraints are satisfied
60      do 65 j=1,k
          funk=funk+(-el(j)/em)**q+(eu(j)/em)**q
65      continue
c
70      funk=em*funk**(1./q)
        goto 95
c        *****  objective function evaluation for SUMT analysis  *****
75      do 76 i=1,k
          funk=funk+el(i)+eu(i)
76      continue
        funk=funk*r
        write(io,*) "funk after motion constraints:",funk
c
c        *****  user specified optimization functions  *****
c        if iobj=1, objective is to minimize link dimension
c
        if (iobj.eq.1) then
          val=0
          do 85 ilink=1,4
            if(linkno.eq.ilink) then
              val=l(ilink)
            endif
85        continue
          funk1=val*wm(1)
          write(io,*) "funk1 - value of link dimension:",funk1
          funk=funk+funk1
c  ***** if iobj=3, objective is to minimize the maximal length of links *****
          elseif (iobj.eq.3) then
              lmax1=0
              do 82 i=1,12
                if(l(i).gt.lmax1) lmax1=l(i)
82            continue
              funk1=lmax1*wm(1)
              write(io,*) "funk1 - the maximal length of links",funk1
              funk=funk+funk1
c  ***** if iobj=9, objective is to minimize the maximal length from r2-12 *****
          elseif (iobj.eq.9) then
              lmax1=0
              do 88 i=3,12
                if(l(i).gt.lmax1) lmax1=l(i)
88            continue
              funk1=lmax1*wm(1)
              write(io,*) "funk1 - the maximal length from r2-r12",funk1
              funk=funk+funk1
        else
        endif
95      fn=funk
        nfeval=nfeval+1
100     write(io,*) "l(*)=",(l(i),i=1,nk)
1000    return
        end
c*****************************************************************
c        function for evaluation of six-bar kinematics          *
c*****************************************************************
        subroutine fbarkine(nsi,ncr,ncr1,nas)
        parameter (novar=14,lclu=2,nocsr=17,nopt=37,nores=25)
        common /fct/lc,csr,w,wm,res,w2,csr0,l,fn,funk1,transang,objv,obja
```

```
      common /fct1/mc,mid,nan,ncp,ml,nk,nnp,pp,p1,lmax,iobj,mm,nn,mm1,nn1
      real l(novar),lc(lclu,novar),transang(nopt)
      dimension csr(nocsr,nopt),w(nocsr,nopt),wm(nocsr),res(nores,nopt),csr0(8)
      integer mc(8),ml(novar),mid(novar),p1,lmax,pp
      pi=3.1415926
      dr=180./pi
      c0=0.
      nas=0
c      ***** solving for the four-bar displacements *****
      do 15 i=1,nnp
        th2=csr(1,i)/dr
        th20=th2+l(13)/dr
        c2=cos(th20)
        s2=sin(th20)
        term1=l(4)**2-l(1)**2-l(2)**2-l(3)**2
        term2=l(1)**2+l(2)**2-l(3)**2+l(4)**2
        a=l(1)/l(3)*c2+c2+term1/(2*l(2)*l(3))-l(1)/l(2)
        b=-2*s2
        c=l(1)/l(3)*c2-c2+term1/(2*l(2)*l(3))+l(1)/l(2)
        d=c2+term2/(2*l(2)*l(4))-l(1)/l(2)-l(1)/l(4)*c2
        e=l(1)/l(2)+term2/(2*l(2)*l(4))-(1+l(1)/l(4))*c2
        sq3=sqrt(b*b-4*a*c)
        sq4=sqrt(b*b-4*d*e)
        if (ncr.eq.1) then
          th3=2*atan((-b+sq3)/(2*a))
          th4=2*atan((-b+sq4)/(2*d))
        else
          th3=2*atan((-b-sq3)/(2*a))
          th4=2*atan((-b-sq4)/(2*d))
        endif
        if (ncp.eq.0) goto 12
        c3=cos(th3)
        s3=sin(th3)
        xl=l(2)*c2+l(5)*c3-l(6)*s3
        yl=l(2)*s2+l(5)*s3+l(6)*c3
c        ***** solving for the velocities and acceleration *****
12      s24=sin(th20-th4)
        s23=sin(th20-th3)
        s43=sin(th4-th3)
        s34=sin(th3-th4)
        c24=cos(th20-th4)
        c23=cos(th20-th3)
        c34=cos(th3-th4)
        c24=cos(th20-th4)
        th3d=l(2)*w2*s24/(l(3)*s43)
        th4d=l(2)*w2*s23/(l(4)*s43)
        th3dd=(l(4)*th4d**2-l(2)*w2**2*c24-l(3)*th3d**2*c34)/(l(3)*s34)
        th4dd=(l(3)*th3d**2+l(2)*w2**2*c23-l(4)*th4d**2*c34)/(l(4)*s43)
        if (ncp.eq.0) goto 14
c      ***** vel. and acc. of the coupler point *****
        xld=-l(2)*w2*s2-l(5)*th3d*s3-l(6)*th3d*c3
        yld=l(2)*w2*c2+l(5)*th3d*c3-l(6)*th3d*s3
        xldd=-l(2)*w2**2*c2-l(5)*(th3dd*s3+th3d**2*c3)-l(6)*(th3dd*c3-th3d**2*s3)
        yldd=-l(2)*w2**2*s2+l(5)*(th3dd*c3-th3d**2*s3)-l(6)*(th3dd*s3+th3d**2*c3)
c      ***** six-bar kinematic analysis ******
        s4=sin(th4)
        c4=cos(th4)
        xlp2=l(1)+l(7)*c4+l(8)*s4
        ylp2=l(7)*s4-l(8)*c4
        xlp12=xl-xlp2
```

```
          ylp12=yl-ylp2
18        term3=l(10)**2-xlp12**2-ylp12**2-l(9)**2
          term4=xlp12**2+ylp12**2+l(10)**2-l(9)**2
          a1=term3/l(9)-2*xlp12
          b1=4*ylp12
          c1=term3/l(9)+2*xlp12
          d1=term4/l(10)-2*xlp12
          e1=term4/l(10)+2*xlp12
          if (((b1*b1-4*a1*c1).ge.0).and.((b1*b1-4*d1*e1).ge.0)) goto 19
          nas=1
          goto 50
19        sq5=sqrt(b1*b1-4*a1*c1)
          sq6=sqrt(b1*b1-4*d1*e1)
c
          if (ncr1.eq.1) then
            th5=2*atan((-b1+sq5)/(2*a1))
            th6=2*atan((-b1+sq6)/(2*d1))
          else
            th5=2*atan((-b1-sq5)/(2*a1))
            th6=2*atan((-b1-sq6)/(2*d1))
          endif
          c5=cos(th5)
          s5=sin(th5)
          c6=cos(th6)
          s6=sin(th6)
c
c         xlp3=xlp2+l(11)*c5+l(12)*s5
c         ylp3=ylp2+l(11)*s5-l(12)*c5
          xlp3=xl+l(11)*c6-l(12)*s6
          ylp3=yl+l(11)*s6+l(12)*c6
c
14        res(1,i)=th2*dr
          res(2,i)=th3*dr
          res(3,i)=th3d
          res(4,i)=th3dd
          res(5,i)=th4*dr
          res(6,i)=th4d
          res(7,i)=th4dd
          if (ncp.eq.0) goto 15
c         l(14) is orientation of the local CS
          sl=sin(l(14)/dr)
          cl=cos(l(14)/dr)
          res(8,i)=xl*cl-yl*sl
          res(9,i)=xld*cl-yld*sl
          res(10,i)=xldd*cl-yldd*sl
          res(11,i)=yl*cl+xl*sl
          res(12,i)=yld*cl+xld*sl
          res(13,i)=yldd*cl+xldd*sl
          res(14,i)=th6*dr
          res(17,i)=th5*dr
          res(20,i)=xlp3*cl-ylp3*sl
          res(23,i)=xlp3*sl+ylp3*cl
          res(25,i)=sqch
15      continue
        csr0(5)=res(14,1)
        csr0(6)=res(17,1)
        csr0(7)=res(20,1)
        csr0(8)=res(23,1)
50      return
        end
```

```
c****************************************************************
c        function for evaluation of dmax and dmin               *
c****************************************************************
        subroutine fbarkine(nsi,ncr,ncr1,nas,dmax,dmin)
        parameter (novar=14,lclu=2,nocsr=17,nopt=37,nores=25)
        common /fct/lc,csr,w,wm,res,w2,csr0,l,fn,funk1,transang,objv,obja
        common /fct1/mc,mid,nan,ncp,ml,nk,nnp,pp,p1,lmax,iobj,mm,nn,mm1,nn1
        real l(novar),lc(lclu,novar),transang(nopt)
        dimension csr(nocsr,nopt),w(nocsr,nopt),wm(nocsr),res(nores,nopt),csr0(8)
        integer mc(8),ml(novar),mid(novar),p1,lmax,pp
        pi=3.1415926
        dr=180./pi
        c0=0.
        nas=0
        dmax=0
        dmin=1000000
        do 15 i=1,73
          th2=(i-1)*pi/36
          th20=th2+l(13)/dr
          c2=cos(th20)
          s2=sin(th20)
          term1=l(4)**2-l(1)**2-l(2)**2-l(3)**2
          term2=l(1)**2+l(2)**2-l(3)**2+l(4)**2
          a=l(1)/l(3)*c2+c2+term1/(2*l(2)*l(3))-l(1)/l(2)
          b=-2*s2
          c=l(1)/l(3)*c2-c2+term1/(2*l(2)*l(3))+l(1)/l(2)
          d=c2+term2/(2*l(2)*l(4))-l(1)/l(2)-l(1)/l(4)*c2
          e=l(1)/l(2)+term2/(2*l(2)*l(4))-(1+l(1)/l(4))*c2
          sq3=sqrt(b*b-4*a*c)
          sq4=sqrt(b*b-4*d*e)
          if (ncr.eq.1) then
            th3=2*atan((-b+sq3)/(2*a))
            th4=2*atan((-b+sq4)/(2*d))
          else
            th3=2*atan((-b-sq3)/(2*a))
            th4=2*atan((-b-sq4)/(2*d))
          endif
          if (ncp.eq.0) goto 12
          c3=cos(th3)
          s3=sin(th3)
          xl=l(2)*c2+l(5)*c3-l(6)*s3
          yl=l(2)*s2+l(5)*s3+l(6)*c3
c         ******  six-bar kinematic analysis  ********
          s4=sin(th4)
          c4=cos(th4)
          xlp2=l(1)+l(7)*c4+l(8)*s4
          ylp2=l(7)*s4-l(8)*c4
          xlp12=xl-xlp2
          ylp12=yl-ylp2
          sqch=sqrt(xlp12**2+ylp12**2)
          if (sqch.gt.dmax) dmax=sqch
          if (sqch.lt.dmin) dmin=sqch
15      continue
50      return
        end
```

VITA

Surname: Fang | Given Names: Xiaoqi

Place of Birth: Chang-chun city, P.R.China | Date of Birth: April, 27, 1956

Educational Institutions Attended:

University of Victoria	1992 to 1994
Dalian Institute of Technology	1985 to 1986
Jilin University of Technology	1978 to 1982

Degrees Awarded:

B.Sc. in Engineering	Jilin University of Technology	1982

Honours and Awards:

Univ. of Victoria Teaching Fellowship	1992 to 1994

Publications:

Fang, X., and Podhorodeski, R.P., "Parameter Transforms allowing Unconstrained Optimization within Specific Planar Mechanism Sub-types", *Proc. of the 12th Symposium on Engineering Applications of Mechanics*, pp.163-172, Montreal, June 1994.

PARTIAL COPYRIGHT LICENSE

I hereby grant the right to lend my thesis to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Title of Thesis:

An Optimization-Based Approach for the Synthesis of Practical Mechanisms

Author: [redacted]

XIAOQI FANG

Nov. 3rd, 1994

Date