NUMERICAL CONTINUATION AND BIFURCATION METHODS FOR MECHANISM WORKSPACE AND CONTROLLABILITY ANALYSIS

by

Frederick Addison Adkins IV

An Abstract

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Applied Mathematical and
Computational Sciences in the
Graduate College of The
University of Iowa

July 1996

Thesis supervisor: Professor Edward J. Haug

ABSTRACT

Many mechanical system performance questions can be formulated as determining boundary surfaces of implicitly defined volumes in which a mechanism or manipulator can carry out useful work. Mathematical modelling and related formulations that characterize these questions are developed. A systematic method using numerical continuation and computational geometric techniques to calculate the associated boundary surfaces are presented. Automatic differentiation techniques are employed to calculate higher order derivatives needed for computing tangents to continuation curves. New design formulations for operational envelopes and domains of interference involve determining where parametrically defined volumes can move or interfere in space. Two C–language automatic differentiation codes are compared in computing the operational envelope for a spatial Stewart Platform with dome.

New results, showing equivalence of extracting boundary surfaces of implicitly defined volumes before or after reduction to one-dimensional curves in two dimensional output planes, motivate faster solution methods for tangents at bifurcation points and validate modelling methods previously applied. Several previously unresolved issues in computing branches at multiple bifurcation points are resolved, increasing the scope of solvable problems. Dense output using Hermite interpolation is added to PITCON for the boundary calculation, to improve graphical output clarity. An efficient numerical continuation algorithm, using optimized step size control tailored to boundary surface calculation, is developed.

Interior singular curves that satisfy conditions similar to exterior boundary surfaces are often associated with restrictions on mechanism control. A new method, interpreting these interior singular surfaces, uses the Taylor expansion of an output functional, taking advantage of the ability to describe the normal to curves found by numerical methods while determining such singular surfaces. Properties of output and velocity restrictions that exist across such interior singular curves are analyzed. Analytical criteria that characterize these properties are presented.

Design issues associated with workspace analysis; e.g., range of motion, accessible output set, domains of operation, domains of interference, and control of robots and mechanisms, are used to illustrate bifurcation branch computations, numerical algorithms, and associated boundary surface calculation. These design issues are applied to models of the six degree of freedom Iowa Driving Simulator motion platform, a serial robotic arm, and other manipulators.

Abstract approved: ________________________________
Thesis supervisor

____
Title and department

____
Date

NUMERICAL CONTINUATION AND BIFURCATION METHODS FOR MECHANISM WORKSPACE AND CONTROLLABILITY ANALYSIS

by

Frederick Addison Adkins IV

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Applied Mathematical and
Computational Sciences in the
Graduate College of The
University of Iowa

July 1996

Thesis supervisor: Professor Edward J. Haug

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Frederick Addison Adkins IV

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Applied Mathematical and Computational Sciences at the July 1996 graduation.

Thesis committee: ______________________
Thesis supervisor

____
Member

____
Member

____
Member

____
Member

To GOD, my parents, my wife and my daughter who
continually provided encouragement, motivation, and love

ACKNOWLEDGEMENTS

My deepest respect, gratitude, and appreciation to my advisor, mentor, and Professor Edward J. Haug, for his many years of patience, encouragement, guidance, thoughtful suggestions, and critical analysis which helped me shape this dissertation and refine my goals in life. Additional appreciation for the discussions and guidance from Professor Herbert Hethcote, Applied Mathematical and Computational Sciences program chair.

I am grateful to my family, especially my mother and father, JoAnne and Frederick, for their love, support, encouragement, and confidence. Special thanks must go to my wife, Stacie Sawallish-Adkins, for her persistence in motivating my completion of this thesis, and my daughter, Amanda, for providing fun and joy throughout the last two years.

ABSTRACT

Many mechanical system performance questions can be formulated as determining boundary surfaces of implicitly defined volumes in which a mechanism or manipulator can carry out useful work. Mathematical modelling and related formulations that characterize these questions are developed. A systematic method using numerical continuation and computational geometric techniques to calculate the associated boundary surfaces are presented. Automatic differentiation techniques are employed to calculate higher order derivatives needed for computing tangents to continuation curves. New design formulations for operational envelopes and domains of interference involve determining where parametrically defined volumes can move or interfere in space. Two C–language automatic differentiation codes are compared in computing the operational envelope for a spatial Stewart Platform with dome.

New results, showing equivalence of extracting boundary surfaces of implicitly defined volumes before or after reduction to one-dimensional curves in two dimensional output planes, motivate faster solution methods for tangents at bifurcation points and validate modelling methods previously applied. Several previously unresolved issues in computing branches at multiple bifurcation points are resolved, increasing the scope of solvable problems. Dense output using Hermite interpolation is added to PITCON for the boundary calculation, to improve graphical output clarity. An efficient numerical continuation algorithm, using optimized step size control tailored to boundary surface calculation, is developed.

Interior singular curves that satisfy conditions similar to exterior boundary surfaces are often associated with restrictions on mechanism control. A new method, interpreting these interior singular surfaces, uses the Taylor expansion of an output functional, taking advantage of the ability to describe the normal to curves found by numerical methods while determining such singular surfaces. Properties of output and velocity restrictions that exist across such interior singular curves are analyzed. Analytical criteria that characterize these properties are presented.

Design issues associated with workspace analysis; e.g., range of motion, accessible output set, domains of operation, domains of interference, and control of robots and mechanisms, are used to illustrate bifurcation branch computations, numerical algorithms, and associated boundary surface calculation. These design issues are applied to models of the six degree of freedom Iowa Driving Simulator motion platform, a serial robotic arm, and other manipulators.

TABLE OF CONTENTS

List of Tables

List of Figures

Figure Page

CHAPTER 1
INTRODUCTION AND LITERATURE SURVEY

Mechanisms and manipulators are being designed more robustly in less time and at lower cost using modern computer aided design and engineering techniques. As analysis software available to industry becomes better, less expensive, and more widely used, engineers are freed to ask detailed design questions and are empowered and obligated to rapidly design better mechanisms and manipulators. Questions such as whether mechanisms can achieve a desired range of motion, have singular configurations, or experience component interference within such ranges are important design considerations. Comprehensive comprehension of the complete kinematic behavior of mechanisms before they are manufactured can save both time and money, as well as significantly improve the quality of the product.

Most current analysis of mechanism workspaces is specific to particular problems, or is limited to certain classes of mechanisms. This research is to further the development of a general methodology for workspace analysis and related design issues. These issues are posed as the determination of boundary surfaces of working domains that are implicitly defined by a set of non-linear algebraic equations. The methods contained herein are broadly applicable for solution of such surfaces, regardless of the application and methodology from which the nonlinear constraint equations were modelled, and form a general numerical scheme for determining implicitly defined boundary surfaces. Examples presented herein are chosen from the area of mechanism and manipulator design to characterize barriers to output control, ranges of operation, and domains of interference that may occur between bodies. The equations defining domains are derived in terms of equations defining the kinematics of the mechanism and the geometry of the working bodies. Analytical criteria defining the boundaries of such domains are derived and numerical methods for computing families of generators on their boundaries are outlined. The criteria and numerical methods presented are applied to both planar and spatial examples.

1.1 Introduction and History

Criteria for boundaries of workspaces of mechanisms and manipulators, using a variety of conditions, formulations, and examples, have received considerable attention. This section presents a survey of this work, followed by an a overview of current research directions to develop a unified numerical framework for workspace analysis and related mathematical works.

A number of authors have developed terminology and formulations for analysis of workspace characteristics that are dependent on specific mechanisms. For serial manipulators, Piper and Roth [75] suggested a partitioning of links into *regional and orientation structures*, separating gross motion of the manipulator from end effector orientation. This work was used when Roth [83] and Shimano and Roth [86] presented the first works related to manipulator workspace. They explained relationships of the dimensions of the workspace to manipulator parameters, showing that the location of the end effector is controlled by certain links, while its orientation is controlled by others.

Litvin [58] used the implicit function theorem to define singular configurations of mechanisms as criteria for boundaries of workspaces. The singularity of constraint Jacobian or velocity transformation matrices has been used by a number of authors to develop criteria for boundaries of workspaces of mechanisms and manipulators in the recent past. Kumar and Waldron [51] described a method for locating the *reachable workspace* by examining the stability of the manipulator via singularity of velocity transformation matrices between input and output coordinates. Waldron, Wang, and Bolin [96, 98]; Litvin, Fanghella, Tan, Zhang [59]; Litvin Castelli and Innocenti [60] developed methods based on Jacobian singularity which were applied to serial robots. Lipkin and Pohl [57] determined singular configurations due to zero determinant of Jacobian minors.

Analytical conditions associated with special features of the geometry of specific manipulators have been used by a number of authors to obtain explicit criteria for boundaries of workspaces. Tsai and Soni [94, 95] examined two and three joint planar manipulators and developed an algorithm for the general *n-revolute* jointed robot. Hsu and Kohli [46, 47] developed algorithms based on Jacobian matrices that were applied to the workspace boundary of regional structures and showed that there were boundaries separating output regions where multiple input configurations are possible. Spanos and Kohli [88] applied a method of polynomial discriminants to workspace boundary of regional structures. Lia and Yang [54, 55] presented a singularity analysis of simple six-link manipulators. Other work on workspace analysis of specific mechanisms includes Refs. 1, 2, 3, 19, 23, 64, 67, 79, 87, 92, and 104.

Some work on determining the ability to reach locations, with some specified orientability range of the end effector, the so–called *dexterous workspace*, was done by Gupta and Roth [29]. Analytical and numerical methods were developed for to dexterous workspace analysis by Haug, Wang and Wu [41, 97]. Numerical methods were also applied to *working capability analysis* by Yang and Haug [102].

Barriers to output control of mechanisms and manipulators is a more recently developed area. Yang and Haug [102] comment that "the use of the term *workspace* to represent the feasible position space may be misleading, since a mechanism may not be capable of working freely within its feasible position space." Hence, determination of such barriers is an important part of workspace analysis. Crossable and noncrossable boundary surfaces were first presented in Oblak and Kohli [74]. This paper addressed Jacobian singularities for regional structures and considered the effects of limited joint displacement, but did not present a unified method for identifying a crossable surface or directions of admissible movement. Recent work on crossable surfaces for serial manipulators was done by Abdel-Malek and Yeh [1, 105, 106]. Young and Wu [107] presented work on path feasibility and modification. Haug, Adkins, Qiu and Yen [36] presented a numerical method using Taylor expansions of an output functional, which takes advantage of numerical methods for determining singular curves in accessible output sets for general mechanisms. This work is detailed in Chapter 5.

In the early 1980's some numerical methods were used to analyze workspace boundaries [56, 101]. More recently, several numerical methods for mapping boundaries of workspaces of mechanisms and manipulators have been presented [40, 41, 49, 103]. These methods are summarized in Section 1.3, with accompanying numerical methods. This thesis furthers the development of a general purpose numerical formulation and solution framework begun with these works by increasing applicability, level of analysis, and efficiency of solution techniques.

Development of numerical methods for workspace analysis has been fuelled by advances in numerical analysis of systems of nonlinear equations and related bifurcation issues. Numerical continuation is the basis for many of the solution methods. The work by Rheinboldt [80, 81, 82] and the associated computer code PITCON [80] are important solution techniques. The book, "Numerical Continuation," by Allgower and Georg [6] in an invaluable reference on this topic, and Refs. 5, 7, 8, 11, 16, 22, 45 and 100 present important advances in the area of numerical continuation. The solution sets to systems of nonlinear equations often have bifurcation points, where the solution splits into two or more components. Several recent volumes in the *International Series of Numerical Mathematics* [4, 53, 66, 85] contain articles on the issue of such bifurcation points, especially Refs. 52 and 76. Other sources of information on bifurcation points include Refs. 13, 15, 17, 24, 28, 44, 42, 63, 68, and 77. Methods for solving systems of polynomial equations are needed for determining branches at multiple bifurcation points. HOMPACK [69, 71, 99] solves such systems using polynomial continuation [21]. Other work on the solution of polynomial systems can be found in Refs. 20 and 70. Derivative calculation is necessary for computation of bifurcation branches, numerical continuation, and points on boundary surfaces. The development of computer codes for it automatic differentiation enables easy computation of such derivatives. A-DOLC [27] and ADIC [10] are such codes, whose related theories are presented in Refs. 25, 26, and 48.

1.2 Analytical Conditions For Working Domains And Boundaries

A variety of workspace formulations are possible to determine physical properties of mechanisms and manipulators. These include accessible output sets [40, 49], dexterous accessible output sets [41], operational envelopes [34, 35], domains of interference [33, 34], and domains of mobility [32]. These formulations all have in common a collection of equations that govern the kinematics of the mechanism and additional equations that determine either performance requirements or the shape of a working body.

In order to define working domains for bodies that move with an underlying mechanism, the mechanism is modeled with a generalized coordinate vector $\mathbf{c}$ that controls its configuration, and the shapes of domains of its working bodies are parameterized by a vector $\boldsymbol{\mu}$. Defining the n_z-vector $\mathbf{z} = \left[\mathbf{c}^T, \boldsymbol{\mu}^T\right]^T$, the n_u-vector $\mathbf{u}$ to a working point on a working body is given analytically in the form $\mathbf{u} = \mathbf{g}\left(\mathbf{z}\right)$, where the vector function $\mathbf{g}(\mathbf{z})$ is presumed to be twice continuously differentiable. The kinematic constraint equations for the mechanism, equations involving parameterization of the shapes of working bodies which may account for inequalities with slack variables [49] to define domains, and the relation $\mathbf{u} - \mathbf{g}\left(\mathbf{z}\right) = \mathbf{0}$ are collected are written in the form

$$\mathbf{\Phi}\left(\mathbf{u}, \mathbf{z}\right) = \mathbf{0} \tag{1.1}$$

where $\mathbf{\Phi}\left(\mathbf{u}, \mathbf{z}\right) : \Re^{n_u} \times \Re^{n_z} \mapsto \Re^{m}$ is such that $\mathbf{\Phi} \in C^2(\Re^{n_u} \times \Re^{n_z}, \Re^{m})$. If the desired output coordinates $\mathbf{u}$ are naturally used in modelling the mechanism then there is no need to introduce $\mathbf{u} = \mathbf{g}\left(\mathbf{z}\right)$ and the collection of variables can still be partitioned into $\mathbf{u}$ and $\mathbf{z}$. To unify the following presentation, assume that all variables are incorporated into a generalized coordinate vector $\mathbf{q} = \left[\mathbf{u}^T, \mathbf{z}^T\right]^T$. The typical workspace problem is to determine the boundary of the solution set of Eq. 1.1 in the $\mathbf{u}$–space; i.e., the domain of interest is the set

$$A = \left\{\mathbf{u} \in \Re^{n_u} : \mathbf{\Phi}\left(\mathbf{u}, \mathbf{z}\right) = \mathbf{0},\ \mathit{for\ some}\ \mathbf{z}\right\} \tag{1.2}$$

To characterize this domain, criteria for points on its boundary are needed. Let $\mathbf{u}^* \in A$. Then, there is a $\mathbf{z}^*$ such that $\mathbf{\Phi}\left(\mathbf{u}^*, \mathbf{z}^*\right) = \mathbf{0}$. If $\mathbf{\Phi}_{\mathbf{z}}\left(\mathbf{u}^*, \mathbf{z}^*\right)$ has full row rank then, by the implicit function theorem [50], there exists a twice-continuously differentiable function $\mathbf{h}\left(\mathbf{u}\right)$ and neighborhoods $\mathbf{U}$ and $\mathbf{Z}$ of $\mathbf{u}^*$ and $\mathbf{z}^*$, respectively, and $\mathbf{z} = \mathbf{h}\left(\mathbf{u}\right)$ such that $\mathbf{\Phi}\left(\mathbf{u}, \mathbf{h}\left(\mathbf{u}\right)\right) = \mathbf{0}$, for all $\mathbf{u} \in \mathbf{U}$ and $\mathbf{z} \in \mathbf{Z}$. The existence of the function $\mathbf{z} = \mathbf{h}(\mathbf{u})$ implies that for any curve in the output space passing through $\mathbf{u}^*$ that, at least locally, there is some value of the generalized coordinate for each point on any such curve that $\boldsymbol{\Phi}(\mathbf{u}, \mathbf{h}(\mathbf{u})) = \mathbf{0}$. Hence all points in neighborhood $\mathbf{U}$ of $\mathbf{u}^*$ can be achieved, implying that $\mathbf{u}^*$ is in the interior of A. Thus, a necessary condition for $\mathbf{u}^*$ to be on the boundary of A, denoted ∂A, is thus that $\mathbf{u}^* \in A$ and $\boldsymbol{\Phi}_{\mathbf{z}}(\mathbf{u}^*, \mathbf{z}^*)$ is row-rank deficient. A necessary and sufficient condition for $\boldsymbol{\Phi}_{\mathbf{z}}(\mathbf{u}^*, \mathbf{z}^*)$ to be row rank deficient is that there is some $\boldsymbol{\tau} \in \Re^m$, $\boldsymbol{\tau}^T\boldsymbol{\tau} = 1$, such that

$$\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{u}^*, \mathbf{z}^*)\,\boldsymbol{\tau} = \mathbf{0} \tag{1.3}$$

Thus a necessary condition for describing the boundary of the solution set of Eq. 1.1 in the $\mathbf{u}$–space is

$$\partial A \subset S \equiv \{\mathbf{u} \in \Re^{n_u} : \boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\tau} = \mathbf{0},\ \boldsymbol{\tau}^T\boldsymbol{\tau} = 1,\ \boldsymbol{\Phi}(\mathbf{u}, \mathbf{z}) = \mathbf{0},\ \text{for some } \mathbf{z} \text{ and } \boldsymbol{\tau}\} \tag{1.4}$$

Due to the necessary, but not sufficient conditions used to determine ∂A, S has additional curves or surfaces which are not in ∂A. These curves or surfaces can be interpreted as possible barriers to local control of the mechanism or manipulator, and details of this analysis are presented in Chapter 5. The boundary ∂A can be found using numerical continuation techniques which are presented in the next section.

1.3 Numerical Methods For Mapping Boundaries

A brief summary of calculations involved in mapping one-dimensional generators for ∂A [40, 61] is given here.

1.3.1 Finding an Interior Point of A

Simulations can often be carried out to find a configuration of the mechanism and points on working bodies satisfying criteria of Eq. 1.1, yielding a point $\mathbf{u}^0$ interior to A. Alternatively, an iterative method can be used to find such a point, as follows [61]. Recall $\mathbf{q} = \left[\mathbf{u}^T,\ \mathbf{z}^T\right]^T$ and $\mathbf{q}^{(0)}$ is an estimate of a point in A. The generalized Newton method [40, 61], defined by the recursive relation

$$\mathbf{\Phi_q}\left(\mathbf{q}^{(j)}\right)\Delta\mathbf{q}^{(j)} = -\mathbf{\Phi}\left(\mathbf{q}^{(j)}\right),\ \mathbf{q}^{(j+1)} = \mathbf{q}^{(j)} + \Delta\mathbf{q}^{(j)} \tag{1.5}$$

where $j = 1,\ 2,\ \ldots$ is the iteration counter and the first of Eqs. 1.5 is solved using the Moore-Penrose generalized inverse [6]. If the method converges, the resulting $\mathbf{u}^{(j+1)}$ is in A and is denoted $\mathbf{u}^0$.

1.3.2 Finding an Initial Point on ∂A

Finding an initial point on ∂A, as the basis for initiating continuation computation to locate all candidate points on ∂A is a non-trivial task. As in Luh [61], a unit vector $\mathbf{d}$ is selected to define a ray emanating from the point $\mathbf{u}^0$ interior to A that is to be traversed using a small step-size h, until a point outside A is encountered. From a candidate point $\mathbf{u}^{i-1}$, a new point is defined by $\mathbf{u}^i = \mathbf{u}^{i-1} + h\mathbf{d},\ \ i = 1,\ 2, \ldots$.

At a predicted point $\mathbf{u}^i$, the recursive relation of Eq. 1.5, with $\mathbf{q}$ replaced by $\mathbf{z}$, is solved using the Moore-Penrose generalized inverse. If the iterative process converges to a solution of Eq. 1.1, then a point interior to A has been determined and another step is taken along ray $\mathbf{d}$. This process is continued until a point is reached at which the iterative process fails to converge, signaling that the point is outside A. An interval halving technique is employed to subdivide the interval containing the boundary point, until a boundary point $\mathbf{q}^*$ is found.

Once a point on the boundary is found, the vector $\boldsymbol{\tau}$ from Eq. 1.3 must be determined. Depending on the solution tolerances used to stop the interval halving technique, the matrix $\mathbf{\Phi_z}(\mathbf{q})$ may have a small singular value but it may not be row rank deficient. Therefore, the system

$$\begin{bmatrix} \mathbf{\Phi}_{\mathbf{z}}^{T}\left(\mathbf{u}^{*}, \mathbf{z}^{*}\right) \boldsymbol{\tau} \\ \boldsymbol{\tau}^{T}\boldsymbol{\tau} - 1 \end{bmatrix} = \mathbf{0} \tag{1.6}$$

may have no solution for $\boldsymbol{\tau}$.

A least squares solution [9] for $\boldsymbol{\tau}$ from the following system of linear equations is used to approximate $\boldsymbol{\tau}$

$$\begin{bmatrix} \mathbf{\Phi}_{\mathbf{z}}^{T}(\mathbf{q}^{*}) \\ \mathbf{e}_{i}^{T} \end{bmatrix} \boldsymbol{\tau} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \tag{1.7}$$

where $\mathbf{e}_i$ is the unit vector with i-th component equal to unity, $i = 1, 2, \ldots, m$. The least squares solution to Eq. 1.7 is the solution $\boldsymbol{\tau}^i$ of [40]

$$(\mathbf{\Phi}_{\mathbf{z}}\mathbf{\Phi}_{\mathbf{z}}^{T} + \mathbf{e}_i\mathbf{e}_i^{T})\boldsymbol{\tau}^i = \mathbf{e}_i \tag{1.8}$$

The vector $\boldsymbol{\tau}^i$ is obtained from Eq. 1.8, using any linear solver, $i = 1, 2, \ldots, m$. The condition number of the matrix $(\mathbf{\Phi}_{\mathbf{z}}\mathbf{\Phi}_{\mathbf{z}}^{T} + \mathbf{e}_i\mathbf{e}_i^{T})$ is obtained as it is decomposed. For the smallest condition number corresponding to the matrix $(\mathbf{\Phi}_{\mathbf{z}}\mathbf{\Phi}_{\mathbf{z}}^{T} + \mathbf{e}_i\mathbf{e}_i^{T})$, the associated $\boldsymbol{\tau}^i$ is most reliable. Thus, $\boldsymbol{\tau}^*$ is chosen to be the $\boldsymbol{\tau}^i$ such that the condition number of $(\mathbf{\Phi}_{\mathbf{z}}\mathbf{\Phi}_{\mathbf{z}}^{T} + \mathbf{e}_i\mathbf{e}_i^{T})$ is the smallest, among $i = 1, 2, \ldots, m$ [40].

The $\boldsymbol{\tau}^*$ obtained from the above algorithm, together with the approximate singular configuration $\mathbf{q}^*$ obtained using the generalized inverse method presented earlier in this section, are used as an initial estimate to begin the continuation process for mapping boundary curves.

1.3.3 Mapping One-Dimensional Solution Curves on ∂A

Defining the complete set of variables involved in the formulation as $\mathbf{x} = \left[\mathbf{u}^T,\ \mathbf{z}^T,\ \boldsymbol{\tau}^T\right]^T$, the equations defining the sets from Eq. 1.4 are

$$\mathbf{G}^*(\mathbf{x}) = \begin{bmatrix} \boldsymbol{\Phi}(\mathbf{u},\ \mathbf{z}) \\ \boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{u},\ \mathbf{z})\,\boldsymbol{\tau} \\ \boldsymbol{\tau}^T\boldsymbol{\tau} - 1 \end{bmatrix} = \mathbf{0} \tag{1.9}$$

If $\mathbf{u}$ is in the plane, then Eq. 1.9 defines a one-dimensional curve that is a candidate for a segment of ∂A. Note that if $n_u = 3$, then ∂A is a surface in $\Re^3$. A practical approach used to map this boundary [40] is to intersect it with a family of planes and then find the corresponding one dimensional curves in each plane. Define the family of cutting planes in the output set as $\mathbf{L}^T\mathbf{u} = b$, where the vector $\mathbf{L}$ is the normal to the plane intersecting the solution set of Eq. 1.1 and the parameter b defines the location of the plane along the vector $\mathbf{L}$. Selecting a grid of values of b yields a family of one-dimensional curves on ∂A that are generators of the surface. If $n_u > 3$, then $n_u - 2$ hyperplanes of $\mathbf{L}^T\mathbf{u} = \mathbf{b}$ can similarly be introduced.

Mapping one-dimensional boundary segments can be done through the use of numerical continuation. The code PITCON [80] implements such a numerical continuation method. Numerical continuation is based on a tangent to the one-dimensional curve, from an initial point. The coordinates of the initial point are perturbed by a small step in the tangent direction and a correction scheme is used to locate another point such that $\mathbf{G}(\mathbf{x}) = \mathbf{0}$, as shown in Fig. 1.1. This is detailed in the following section.

1.3.4 Determining Tangents to One-Dimensional Curves

When the Jacobian $\mathbf{G}_{\mathbf{x}}(\mathbf{x})$ has full row rank, tangent vectors to continuation curves are uniquely characterized by the following condition [6, 80]:

$$\mathbf{G}_{\mathbf{x}}(\mathbf{x})\mathbf{h}(\mathbf{x}) = \mathbf{0} \tag{1.10a}$$

$$\mathbf{h}(\mathbf{x})^T\mathbf{h}(\mathbf{x}) = 1 \tag{1.10b}$$

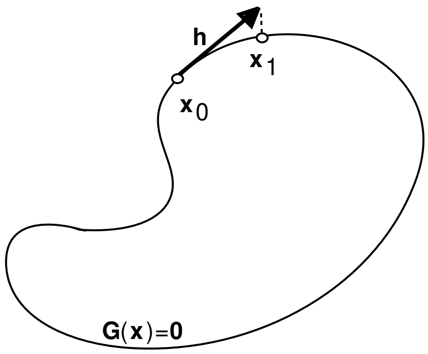

Figure 1.1: Numerical Continuation Process

$$\det \begin{bmatrix} \mathbf{G_x}(\mathbf{x}) \\ \mathbf{h}(\mathbf{x})^T \end{bmatrix} > 0 \tag{1.10c}$$

where $\mathbf{h}(\mathbf{x})$ is a map of class $C^{r-1}, r \geq 2$. Equation 1.10a requires that the tangent be in the null space of $\mathbf{G_x}(\mathbf{x})$. This is the first order necessary condition for a tangent. Equation 1.10b determines the magnitude of the tangent, and Eq. 1.10c specifies consistent tangent orientation. Difficulties arise when the Jacobian matrix $\mathbf{G_x}(\mathbf{x})$ is row rank deficient at some continuation point. Points where $\mathbf{G_x}(\mathbf{x})$ is row rank deficient are candidate bifurcation points; i.e., points at which the continuation curve branches into several curves. A simple bifurcation point $\mathbf{x}_0$ of $\mathbf{G} : \Re^{n_x} \rightarrow \Re^{n_x - 1}$ is a point such that the Jacobian matrix $\mathbf{G_x}(\mathbf{x}_0)$ is row rank deficient of order one; i.e., $\text{rank}\mathbf{G_x}(\mathbf{x}_0) = n_x - 2$ [6, 80]. Multiple bifurcation points occur when $\text{rank}\mathbf{G_x}(\mathbf{x}_0) < n_x - 2$ [6, 80].

For the case of simple bifurcation points, two solution curves that intersect at the bifurcation point are characterized by tangent vectors satisfying the first order necessary condition of Eq. 1.10. The source of singular behavior at $\mathbf{x}_0$ is identified by a series of elementary row operations applied to $\mathbf{G_x}(\mathbf{x}_0)$. The elementary row operations are collected in the matrix $\mathbf{Q}$ [90], and the Jacobian matrix is reduced to the form

$$\mathbf{Q}^T\mathbf{G}_{\mathbf{x}}(\mathbf{x}_0) = \begin{bmatrix} \mathbf{F}_{\mathbf{x}}(\mathbf{x}_0) \\ \mathbf{0} \end{bmatrix} \tag{1.11}$$

where $\mathbf{F}_{\mathbf{x}}(\mathbf{x}_0)$ is an $(n_x-2)\times n_x$ matrix with full row rank. The transformation matrix $\mathbf{Q}$ does not change the solution set of Eq. 1.10, since it involves only interchanging rows, multiplying one row by a non-zero constant, and adding one row to another. A practical numerical approach to obtain the source of singular behavior is by applying QR decomposition to the Jacobian matrix; i.e.,

$$\mathbf{G}_{\mathbf{x}}(\mathbf{x}_0) = \mathbf{Q}\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{Q}$ is an orthogonal matrix; i.e., $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ and $\mathbf{R}$ is an $n_x \times n_x$ upper triangular matrix. Note that

$$\mathbf{Q}^T\mathbf{G}_{\mathbf{x}}(\mathbf{x}_0) = \mathbf{Q}^T\mathbf{Q}\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{\mathbf{x}}(\mathbf{x}_0) \\ \mathbf{0} \end{bmatrix}$$

Let $\mathbf{q}_{n_x-1}$ denote the last column of $\mathbf{Q}$. Then, the source of singular behavior is collected in the *bifurcation equation* [6, 52]

$$f(\mathbf{x}) \equiv \mathbf{q}_{n_x-1}^T\mathbf{G} = 0 \tag{1.12}$$

The Jacobian of the bifurcation function $f(\mathbf{x})$ at $\mathbf{x}_0$ is the last row of $\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$; i.e.,

$$f_{\mathbf{x}}(\mathbf{x}_0) = 0 \tag{1.13}$$

Note that $\mathbf{F}_{\mathbf{x}}(\mathbf{x}_0)$ is a $(n_x-2)\times n_x$ matrix with full row rank; i.e., $\text{rank}\mathbf{F}_{\mathbf{x}}(\mathbf{x}_0) = n_x - 2$, the null space of $\mathbf{F}_{\mathbf{x}}(\mathbf{x}_0)$ is of dimension 2, so there exist two distinct nonzero vectors, $\boldsymbol{\xi}^1$ and $\boldsymbol{\xi}^2$, such that

$$\mathbf{F_x}(\mathbf{x}_0)\boldsymbol{\xi}^i = \mathbf{0}, \qquad i = 1, 2 \tag{1.14}$$

where ${\boldsymbol{\xi}^i}^T \boldsymbol{\xi}^j = \delta_{ij}, i = 1, 2$. The objective is to find tangent vectors $\mathbf{h}$ of branches that intersect at the bifurcation point $\mathbf{x}_0$ of the form

$$\mathbf{h} = \alpha_j \boldsymbol{\xi}^j \tag{1.15}$$

where summation over repeated Latin indices is implied.

To be tangent to the solution curve, the equations governing the solution curve must be satisfied to lowest, non-trivial order Taylor approximation [40, 52]

$$\mathbf{G}(\mathbf{x}_0 + \mathbf{h}) \approx \mathbf{G}(\mathbf{x}_0) + \begin{bmatrix} \mathbf{F_x}(\mathbf{x}_0)\mathbf{h} \\ \frac{1}{2}\mathbf{h}^T f_{\mathbf{xx}}(\mathbf{x}_0)\mathbf{h} \end{bmatrix} = \begin{bmatrix} \mathbf{F_x}(\mathbf{x}_0)\alpha_j \boldsymbol{\xi}^j \\ \frac{1}{2}\alpha_i {\boldsymbol{\xi}^i}^T f_{\mathbf{xx}}(\mathbf{x}_0)\alpha_k \boldsymbol{\xi}^k \end{bmatrix} = \mathbf{0} \tag{1.16}$$

where

$$f_{\mathbf{xx}}(\mathbf{x}_0) = \left[\frac{\partial^2 f(\mathbf{x}_0)}{\partial x_i \partial x_j} \right] \tag{1.17}$$

is the Hessian of $f(\mathbf{x})$ evaluated at $\mathbf{x}_0$. Since the vectors $\boldsymbol{\xi}^i$ are in the null space of $\mathbf{F_x}(\mathbf{x}_0)$, the upper subvector on the right side of Eq. 1.16 is zero, for all α_j. Thus, Eq. 1.16 implies that

$$\boldsymbol{\alpha}^T \mathbf{H} \boldsymbol{\alpha} = 0 \tag{1.18}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2]^T$ and

$$\mathbf{H} = [{\boldsymbol{\xi}^i}^T f_{\mathbf{xx}}(\mathbf{x}_0)\boldsymbol{\xi}^j] \equiv [H_{ij}] \tag{1.19}$$

is a 2×2 matrix. Equation 1.18 is the second order necessary condition for tangents.

Equation 1.18 can be transformed to a quadratic equation in the ratio of the two variables; i.e.,

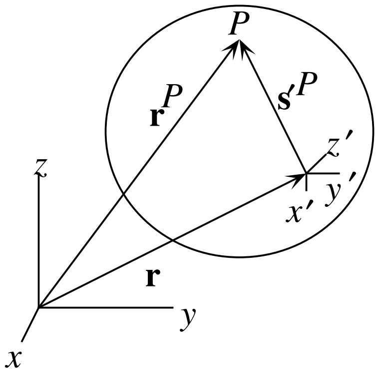

Figure 1.2: Locating Point P on Body

$$H_{11}\left(\frac{\alpha_1}{\alpha_2}\right)^2 + 2H_{12}\left(\frac{\alpha_1}{\alpha_2}\right) + H_{22} = 0 \tag{1.20}$$

and easily solved. With two values of $\boldsymbol{\alpha}$, tangents $\mathbf{h}$ are determined by Eq. 1.15.

The above presentation for simple bifurcation points can be extended for multiple bifurcation points, where $\operatorname{rank}\mathbf{G}_{\mathbf{x}}(\mathbf{x}_0) = k < n_x - 2$. This is done by collecting the last $n_x - k - 1$ columns of $\mathbf{Q}$, from $\mathbf{G}_{\mathbf{x}}(\mathbf{x}_0) = \mathbf{QR}$, which determine the system of *bifurcation equations*, similar to Eq. 1.12. This is addressed in Section 3.2.5.

1.4 Accessible Output Sets

The simplest domain of operation associated with a working body is the accessible output set. It consists of the set of all points in space that can be reached by a working point P that is fixed in the working body, for some value of the mechanism general coordinate vector $\mathbf{c}$ [2, 19, 29, 51, 40, 34, 94, 101]. Working point P is fixed in a working body with an x'-y'-z' body-fixed reference frame, located by the body-fixed vector $\mathbf{s}'^P$, as shown in Fig. 1.2. It is transformed to the global x-y-z frame by the orthogonal orientation transformation matrix $\mathbf{A}(\mathbf{c})$ that is a function of the generalized coordinates; i.e., $\mathbf{s}^P = \mathbf{A}(\mathbf{c})\,\mathbf{s}'^P$. The vector $\mathbf{r}$ that locates the

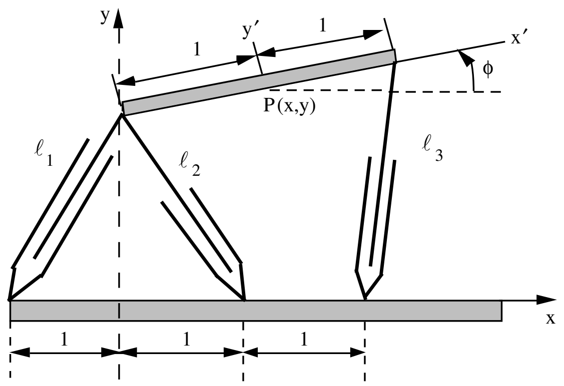

Figure 1.3: Planar Stewart Platform

origin of the body-fixed reference frame is a function of the vector $\mathbf{c}$ of generalized coordinates of the mechanism that controls motion of the working body. The output vector $\mathbf{u}$ in this case is $\mathbf{u} = \mathbf{r}^P = \mathbf{r}(\mathbf{c}) + \mathbf{A}(\mathbf{c})\,\mathbf{s}'^P \equiv \mathbf{g}(\mathbf{c})$. The system of equations of Eq. 1.1 for the accessible output set includes the kinematic constraint equations and $\mathbf{u} - \mathbf{r}(\mathbf{c}) - \mathbf{A}(\mathbf{c})\,\mathbf{s}'^P = \mathbf{0}$. No shape parameters nor shape equations are needed, and thus $\mathbf{z} = \mathbf{c}$. The accessible output set is $A = \{\mathbf{u} : \ \mathbf{\Phi}(\mathbf{u},\ \mathbf{z}) = \mathbf{0},\ for\ some\ \mathbf{z}\}$ as given in Eq. 1.4.

Sample formulations of the accessible output set for planar and spatial Stewart platforms are presented below. The Stewart Platform [89] has been the focus of much research attention [18, 62, 108] and is a realistic real world example for demonstrating mechanism and manipulator workspace analysis. While the information in this section and the following comprise introductory material, the author played a significant role in its development.

1.4.1 Planar Stewart Platform

The planar Stewart platform shown in Fig. 1.3 consists of a moving upper platform and three actuators that control the three degrees of freedom of the platform

[36, 40, 61, 97]. The working point is chosen at the center of the top platform, so $\mathbf{s}'^P = \mathbf{0}$. The actuator lengths ℓ_1, ℓ_2, and ℓ_3 are input coordinates that control the position $\mathbf{u} = \mathbf{r} = [x,\ y]^T$ of point P and the orientation ϕ of the top platform. Actuator length unilateral constraints are $0 < \ell_i^{\min} \leq \ell_i \leq \ell_i^{\max}$, $i = 1,\ 2,\ 3$, where $\ell_1^{\min} = \ell_2^{\min} = \sqrt{2}$, $\ell_3^{\min} = 1$, $\ell_1^{\max} = \ell_2^{\max} = 2$, and $\ell_3^{\max} = \sqrt{3}$. Introducing new input coordinates $v_1,\ v_2$, and v_3 so that

$$\ell_i = a_i + b_i \sin v_i, i = 1,\ 2,\ 3 \tag{1.21}$$

where $a_1 = a_2 = \left(\sqrt{2}+2\right)/2$, $a_3 = \left(1+\sqrt{3}\right)/2$, $b_1 = b_2 = \left(2-\sqrt{2}\right)/2$, and $b_3 = \left(\sqrt{3}-1\right)/2$, the actuator unilateral constraints are automatically satisfied. The generalized coordinate vector is thus $\mathbf{q} = [x,\ y,\ \phi,\ v_1,\ v_2,\ v_3]^T$. Since $\mathbf{s}'^P = \mathbf{0}$, $\mathbf{u} = [x,\ y]^T \equiv \mathbf{g}(\mathbf{z})$ does not need to be added to determine $\mathbf{u}$ from the generalized coordinates. The constraint equations for this manipulator are

$$\boldsymbol{\Phi}(\mathbf{q}) \equiv \begin{bmatrix} \ell_1^2 - (x - \cos\phi + 1)^2 - (y - \sin\phi)^2 \\ \ell_2^2 - (1 - x + \cos\phi)^2 - (y - \sin\phi)^2 \\ \ell_3^2 - (x + \cos\phi - 2)^2 - (y + \sin\phi)^2 \end{bmatrix} = \mathbf{0} \tag{1.22}$$

where ℓ_i is a function of v_i.

To obtain an initial point to begin numerical calculations, the platform is raised to a mid-point in which the upper platform is horizontal; i.e., $\phi = 0$. The initial values for the generalized coordinates for this point interior to A are thus [40]

$$\mathbf{q}_0 = [1,\ 1.5607,\ 0.0,\ 0.5236,\ 0.5236,\ 0.5236]^T$$

Selecting a search direction parallel to the x-axis; i.e., $\mathbf{d} = [-1,\ 0]^T$, the search process of Section 1.3.2 for an initial point on ∂A is carried out, yielding the point $\mathbf{u}^*$ on ∂A shown in Fig. 1.4. The final configuration on ∂A obtained in this process is $\mathbf{q}^* = [0.6018,\ 1.5607,\ -0.5209,\ 1.5694,\ 1.5658,\ 0.1235]^T$. With this approximate point on ∂A, a least squares approximate solution of Eq. 1.3, with $\mathbf{e}_j{}^T\boldsymbol{\tau}^j = 1$, $\mathbf{e}_j = [\delta_{ij}]$,

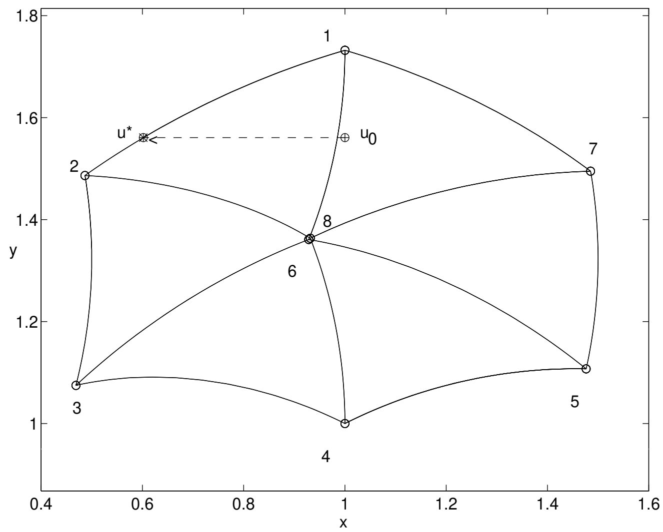

Figure 1.4: Planar Stewart Platform Accessible Output Set

and j set such that $\mathbf{\Phi_z \Phi_z}^T + \mathbf{e}_j \mathbf{e}_j{}^T$ has smallest condition number [40], is obtained as $\boldsymbol{\tau}^2 = [0.0,\ 1.0,\ 0.9284]^T$. With this starting point, candidate curves on ∂A are mapped using the continuation method outlined in Section 1.3.3. The curves obtained are shown in Fig. 1.4, where numbered points are bifurcation points. Points 6 and 8, while appearing to be close together, are actually the projections onto the **u**-space of two distinctly different bifurcation points. The generalized coordinate vector for point 6 is $[x,\ y,\ \phi]^T = [0.928571,\ 1.36090,\ -0.38075]^T$ and that for point 8 is $[0.931663,\ 1.366332,\ 0.37183]^T$.

1.4.2 Spatial Stewart Platform

The Stewart platform [89] shown in Fig. 1.5 consists of a moving upper platform attached to ground by six extendable actuators. The lengths of the six actuators serve as inputs, with upper and lower bounds imposed on the lengths of each, to control the motion of the six-degree-of-freedom platform. The Iowa Driving Simulator (IDS) shown in Fig. 1.6 is a high fidelity ground vehicle simulator that incorporates such a

Figure 1.5: A Spatial Stewart Platform

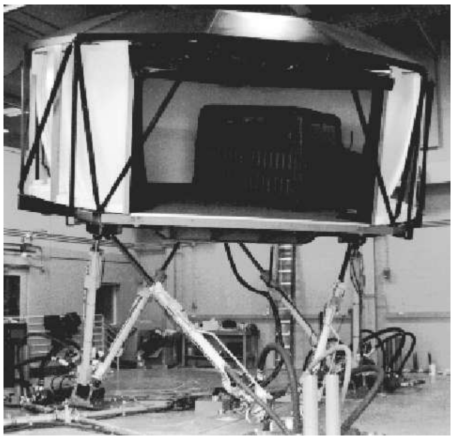

Figure 1.6: Iowa Driving Simulator

Stewart platform as the motion base for six-degree-of-freedom motion control. Details of the kinematic formulation are given in Ref. 41 and an analysis of the accessible output set calculation is given in Ref. 62. This example provides a realistic application for an actual mechanism and will be used to illustrate and compare a variety of techniques throughout this thesis.

A brief discussion of the kinematics formulation and the associated accessible output set is given here. Limitations on the lengths of the six actuators are specified as unilateral constraints of the form

$$0 < \ell_i^{min} \leq \ell_i \leq \ell_i^{max}, i = 1, \ldots, 6 \tag{1.23}$$

They can be accounted for by introducing auxiliary input variables v_i through the relations

$$\ell_i = \ell_i^0 + \ell_i^r \sin v_i \tag{1.24}$$

where $\ell_i^0 = \frac{\ell_i^{max} + \ell_i^{min}}{2}$, $\ell_i^r = \frac{\ell_i^{max} - \ell_i^{min}}{2}$, $i = 1, \ldots, 6$. As v_i varies from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$, the length ℓ_i varies from ℓ_i^{min} to ℓ_i^{max}, ensuring satisfaction of the actuator length unilateral constraints. Kinematic constraint equations for the Stewart platform manipulator are composed of six distance constraints associated with the six actuators and one normalization constraint on the Euler rotation vector $\boldsymbol{\beta}$ [41]. The distance constraint equation for the i-th actuator is

$$\Phi_i = (\mathbf{r} + \mathbf{A}(\alpha, \boldsymbol{\beta})\mathbf{a}_i' - \mathbf{t}_i)^T(\mathbf{r} + \mathbf{A}(\alpha, \boldsymbol{\beta})\mathbf{a}_i' - \mathbf{t}_i) - (\ell_i^0 + \ell_i^r \sin v_i)^2 = 0 \tag{1.25}$$

where $\mathbf{r}$ is the vector from the origin of the $x - y - z$ frame to the origin of the $x' - y' - z'$ frame in Fig. 1.5 and $\mathbf{a}_i'$ and $\mathbf{t}_i$ are the constant local coordinate vectors to the upper and lower actuator attachment locations. Actual data from the Iowa Driving Simulator's Stewart Platform motion base [62] is used in the computations described below. The matrix [41]

$$\mathbf{A}(\alpha, \boldsymbol{\beta}) = \cos\alpha \mathbf{I} + (1 - \cos\alpha)\boldsymbol{\beta}\boldsymbol{\beta}^T + \sin\alpha\tilde{\boldsymbol{\beta}} \tag{1.26}$$

is the orientation transformation matrix for the upper platform, where α is the angle of rotation of the upper platform about a rotation axis that is specified by the unit vector $\boldsymbol{\beta} = [\beta_x, \beta_y, \beta_z]^T$. The skew symmetric matrix $\tilde{\boldsymbol{\beta}}$ in Eq. 1.26 is

$$\tilde{\boldsymbol{\beta}} = \begin{bmatrix} 0 & -\beta_z & \beta_y \\ \beta_z & 0 & -\beta_x \\ -\beta_y & \beta_x & 0 \end{bmatrix} \tag{1.27}$$

The normalization constraint on $\boldsymbol{\beta}$ is

$$\Phi_7 = \boldsymbol{\beta}^T\boldsymbol{\beta} - 1 = 0 \tag{1.28}$$

The vector of generalized coordinates is $\mathbf{q} = \left[\mathbf{r}^T, \alpha, \boldsymbol{\beta}^T, \mathbf{v}^T\right]^T \in \Re^{13}$. Since $\mathbf{s}'^P = \mathbf{0}$ for the center of the upper platform, output coordinates are the coordinates of the position vector from the origin of the global x-y-z frame to the center of the upper platform; i.e., $\mathbf{u} = \mathbf{r} = [x, y, z]^T \in \Re^3$. Thus no additional equations of the form $\mathbf{u} = \mathbf{g}(\mathbf{q})$ are needed. Input coordinates that prescribe the lengths of the six actuators via Eq. 1.24 are $\mathbf{v} = [v_1, v_2, v_3, v_4, v_5, v_6]^T$ and intermediate coordinates are $\mathbf{w} = \left[\alpha, \boldsymbol{\beta}^T\right]^T \in \Re^4$, which specify the orientation of the upper platform. Thus, the kinematic constraint equations are

$$\boldsymbol{\Phi}(\mathbf{q}) = [\Phi_1, \Phi_2, \ldots, \Phi_7]^T = \mathbf{0} \tag{1.29}$$

Since the kinematic constraint equations consist of seven independent equations in thirteen variables, the manipulator has six-degrees-of-freedom.

The accessible output set is a domain in $\Re^3$ and to determine its boundary a family of cutting planes of the form $z - b = 0$ are added to the continuation equation, Eq. 1.9. Resulting singular curves for several cutting planes are presented in Figs. 1.7, 1.8, and 1.9. Points labeled with circles are bifurcation points. Note the increasing level of complexity as the value of z is decreased. Figure 1.10 shows the complete boundary surface, ∂A, for the Stewart Platform.

This manipulator is used for performance comparisons and implementation feasibility for the methods presented in Chapters 2, 3, and 4. Combined with a projection

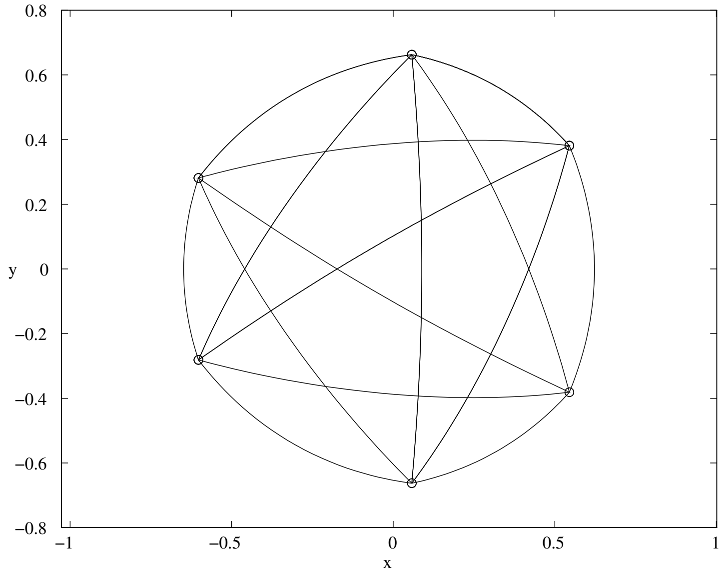

Figure 1.7: All Singular Curves at z = 3.2 Meters

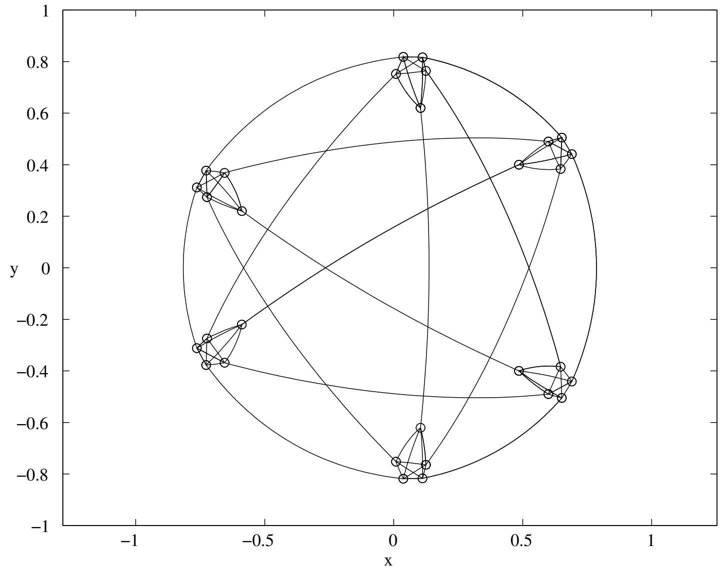

Figure 1.8: All Singular Curves at z = 3.1 Meters

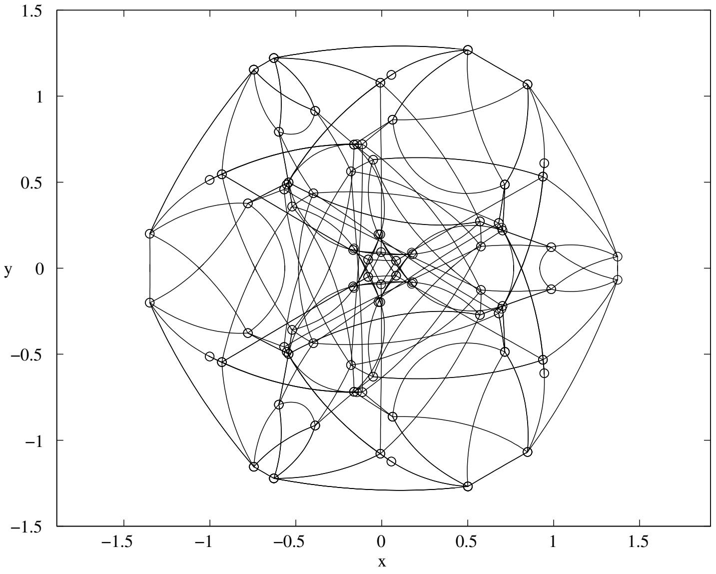

Figure 1.9: All Singular Curves at z = 2.6 Meters

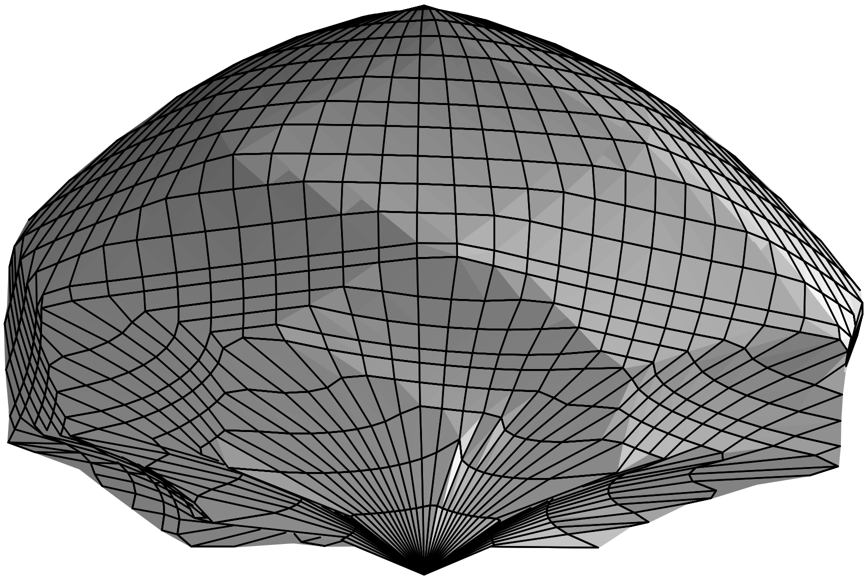

Figure 1.10: ∂A for Stewart Platform of Iowa Driving Simulator

dome shape, it is the basis for new operational envelope results presented in Chapter 6.

1.5 Scope of Research

Note that if $n_u = 3$, then ∂A is a surface in $\Re^3$. A practical approach to map this boundary [40] is to intersect it with a family of planes and then find the corresponding one dimensional curves in each plane.

Chapter 2 shows that intersecting the set defined by Eq. 1.4 with a family of planes is equivalent to first intersecting the solution set of Eq. 1.2 with a family of planes and then applying analytical necessary condition to determine the boundary of the sets in each plane. The method of first intersecting the solution set of Eq. 1.2 with a family of planes is the one to which the method of Ref. 78 for determination of tangents at bifurcation points applies. However, this method also results in an additional equation and variable in the continuation calculation. If a method were available to calculate branches at bifurcation points for the set of Eq. 1.4, intersected by a family of planes, then the continuation calculation can be done without first adding the intersection equation to the set of equations $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$. This would avoid unnecessarily inflating both the dimension of $\boldsymbol{\tau}$ and the number of continuation equations. Chapter 3 develops such a method.

Chapter 3 also presents a new formulation for finding tangents that is more efficient than previous methods and applies in cases of output space of dimension two or greater. This new formulation extends previous work to include cases in which the row rank deficiency of the Jacobian $\mathbf{G}_{\mathbf{x}}(\mathbf{x})$ is greater than one. Since computational efficiency is of concern, performance tests are run on numerical examples to verify the methods presented in Chapters 2 and 3. These tests show significant improvement in computation times.

Chapter 4 presents an alternative numerical continuation scheme to the general purpose PITCON [80] code. This scheme takes advantage of structure of the equations for boundary surfaces. This chapter also presents an extension to the PITCON code, to do dense output for fixed pseudo-arclength step size in either output or full coordinate space.

As noted in Section 1.2, additional singular curves and surfaces may be present in solution set S of the equations $\mathbf{G}(\mathbf{x}) = \mathbf{0}$. Chapter 5 presents a method for interpretation of these interior singular surfaces. This method uses Taylor expansions of an output functional, which takes advantage of the ability to describe normals to curves found by numerical methods while determining singular surfaces in accessible output sets for general mechanisms. This is a new and important result for workspace analysis and manipulator control.

Chapter 6 gives new results for domains of operation and domains of interference. The general formulation and numerical methods for implicit boundary surface calculation applies directly to these models. These areas of workspace analysis are new and, while expanding the scope of design questions that can be answered, provide additional motivation for refinement of a robust computational environment for workspace analysis that can be made available for the working engineer. Results presented in this chapter contain a special purpose implementation and performance comparison for the new automatic differentiation code ADIC [10] relative to the code ADOL-C [27].

Finally, Chapter 7 contains some concluding remarks and discussion of future directions of research.

CHAPTER 2
MAPPING OUTPUT SETS OF DIMENSION > 2

In Section 1.2, an analytic necessary condition for being on the $\mathbf{u}$–space boundary of the solution set of $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$ is formulated. When $\dim \mathbf{u} \geq 3$, a practical approach to mapping the boundary is to define a family of planes (or hyperplanes if $\dim \mathbf{u} > 3$) that intersect the boundary in one dimensional sets that can be numerically mapped. It is shown in this chapter that families of boundary curves obtained in this fashion are identical to curves that result from first intersecting the solution set of $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$ with planes (or hyperplanes) and then determining the boundary of the resulting sets, using the analytic necessary condition.

This is a valuable result because it shows that the bifurcation branch computation method of Qiu, Luh, and Haug [78] can be applied to the formulation, which first applies the intersection and will get the same results as the formulation that applies the intersection after the necessary condition to determine the boundary of the surface has been used. The method in Ref. 78 only applies to the formulation that first applies the intersection to $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$. A new method for bifurcation branch computation, which applies to both formulations, is presented in Chapter 3 and is motivated by the results presented here.

2.1 Preliminaries

Assume that constraint equations representing the set to be mapped are given by Eq. 1.1; i.e., $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$. The generalized coordinates are partitioned as $\mathbf{q} = \left[\mathbf{u}^T, \mathbf{w}^T, \mathbf{v}^T\right]^T$, with the understanding that $\mathbf{u}$ is comprised of output coordinates, $\mathbf{v}$

is comprised of input coordinates and $\mathbf{w}$ is comprised of all the remaining coordinates necessary to describe the mechanism, called the intermediate coordinates. With this notation $\mathbf{z} = \left[\mathbf{w}^T, \mathbf{v}^T\right]^T$. If no natural partitioning is present, a numerical one, such as detailed in Section 3.2.1, can be used. The solution set of Eq. 1.1 in the $\mathbf{u}$–space is Eq. 1.2,

$$A = \{\mathbf{u} \in \Re^{n_u} : \mathbf{\Phi}(\mathbf{u}, \mathbf{w}, \mathbf{v}) = \mathbf{0} \text{ for some } \mathbf{w} \text{ and } \mathbf{v}\} \tag{2.1}$$

A necessary condition for $\mathbf{u} \in A$ to be on the boundary ∂A is that $\mathbf{\Phi}_{\mathbf{w},\mathbf{v}}(\mathbf{q})$ is row rank deficient [40]. A necessary and sufficient condition for $\mathbf{\Phi}_{\mathbf{w},\mathbf{v}}(\mathbf{q})$ to be row rank deficient is that there is some $\boldsymbol{\tau} \in \Re^m$, $\boldsymbol{\tau} \neq \mathbf{0}$, such that,

$$\mathbf{\Phi}^T_{\mathbf{w},\mathbf{v}}(\mathbf{q})\boldsymbol{\tau} = \mathbf{0} \tag{2.2}$$

Thus, a necessary condition for points on the boundary of A in the $\mathbf{u}$–space is

$$\partial A \subset S \equiv \{\mathbf{u} \in A : \mathbf{\Phi}^T_{\mathbf{w},\mathbf{v}}\boldsymbol{\tau} = \mathbf{0},\ \boldsymbol{\tau}^T\boldsymbol{\tau} = 1,\ \mathbf{\Phi}(\mathbf{u}, \mathbf{w}, \mathbf{v}) = \mathbf{0},\ \text{ for some } \mathbf{w}, \mathbf{v}, \text{ and } \boldsymbol{\tau}\} \tag{2.3}$$

Note that if $n_u \equiv \dim \mathbf{u} = 3$, then ∂A is a surface in $\Re^3$. A practical approach used to map this boundary in Ref. 40 is to intersect it with a family of planes and then find the corresponding boundary curves in each plane.

2.2 Analytical Example– Serial Manipulator

The example presented here is that of determining the three dimensional accessible output set for a serial manipulator. A manipulator is serial if component bodies are each only attached to two other bodies and no closed loops of bodies are formed. The mechanism of Fig. 3.2 is an example of a serial manipulator. The constraint equations are given as algebraic relations between inputs $\mathbf{v}$, which determine the location of the output working point $\mathbf{u} = [u_1, u_2, u_3]^T$. There are no intermediate coordinates in this example. The constraint equations can be written as

$$\mathbf{\Phi} = \begin{bmatrix} u_1 + f_1(\mathbf{v}) \\ u_2 + f_2(\mathbf{v}) \\ u_3 + f_3(\mathbf{v}) \end{bmatrix} = \mathbf{0}$$

When adding a cutting plane constraint of the form $l_1 u_1 + l_2 u_2 + l_3 u_3 + b = 0$,

$$\mathbf{\Psi} = \begin{bmatrix} \mathbf{\Phi} \\ l_1 u_1 + l_2 u_2 + l_3 u_3 + b \end{bmatrix} = \mathbf{0} \tag{2.4}$$

Adding an equation to $\mathbf{\Phi}$ reduces the number of independent variables and one of the inputs coordinates is shifted to the intermediates coordinates. Assuming $l_3 \neq 0$ and taking the variable u_3 as the new intermediate coordinate,

$$\mathbf{z} = \left[\mathbf{v}^T\right]^T$$
$$\mathbf{z}^* = \left[u_3, \mathbf{v}^T\right]^T$$
$$\boldsymbol{\tau}^* = \left[\boldsymbol{\tau}^T, \xi\right]^T = [\tau_1, \tau_2, \tau_3, \xi]^T$$

and

$$\mathbf{\Psi}_{\mathbf{z}^*} = \begin{bmatrix} 0 & f_{1\mathbf{v}} \\ 0 & f_{2\mathbf{v}} \\ 1 & f_{3\mathbf{v}} \\ l_3 & \mathbf{0}^T \end{bmatrix}$$

so

$$\mathbf{\Psi}_{\mathbf{z}^*}^T \boldsymbol{\tau}^* = \begin{bmatrix} \tau_3 + l_3 \xi \\ f_{1\mathbf{v}}^T \tau_1 + f_{2\mathbf{v}}^T \tau_2 + f_{3\mathbf{v}}^T \tau_3 \end{bmatrix} = \mathbf{0} \tag{2.5}$$

can be used as the equations which specify points satisfying the necessary condition for being on the the boundary surface.

Thus there are two possible continuation equations,

$$\mathbf{G}^*(\mathbf{q}, \boldsymbol{\tau}^*) = \begin{bmatrix} \boldsymbol{\Psi} \\ \boldsymbol{\Psi}_{\mathbf{z}^*}^T \boldsymbol{\tau}^* \\ \boldsymbol{\tau}^{*T} \boldsymbol{\tau}^* - 1 \end{bmatrix} = \mathbf{0} \tag{2.6}$$

or

$$\mathbf{G}(\mathbf{q}, \boldsymbol{\tau}) = \begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau} \\ \mathbf{L}^T \mathbf{u} + b \\ \boldsymbol{\tau}^T \boldsymbol{\tau} - 1 \end{bmatrix} = \mathbf{0} \tag{2.7}$$

and the following calculations show for this example that Eqs. 2.6 and 2.7 produce the same output space boundary curves.

The first of Eqs. 2.5 can be used to determine the value of ξ analytically, $\xi = -\tau_3/l_3$. Using $\boldsymbol{\tau}^{*T}\boldsymbol{\tau}^* - 1 = \boldsymbol{\tau}^T\boldsymbol{\tau} + \xi^2 - 1 = 0$ and setting $c = 1 - \xi^2$ implies that $\boldsymbol{\tau}^T\boldsymbol{\tau} - c = 0$, where $c > 0$. If $\boldsymbol{\tau}$ is normalized as $\boldsymbol{\tau} \equiv \left[\tau_1, \tau_2, \tau_3\right]^T \equiv \frac{1}{\boldsymbol{\tau}^T\boldsymbol{\tau}}\boldsymbol{\tau} = \frac{1}{c}\boldsymbol{\tau}$,

$$\boldsymbol{\tau}^T\boldsymbol{\tau} - 1 = 0 \tag{2.8}$$

The second of Eqs. 2.5, $f_{1\mathbf{v}}^T\tau_1 + f_{2\mathbf{v}}^T\tau_2 + f_{3\mathbf{v}}^T\tau_3 = \mathbf{0}$, implies

$$\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\tau} \equiv f_{1\mathbf{v}}^T\tau_1 + f_{2\mathbf{v}}^T\tau_2 + f_{3\mathbf{v}}^T\tau_3 = \mathbf{0} \tag{2.9}$$

Equation 2.4, together with Eqs. 2.8 and 2.9, shows that the subequations of Eq. 2.7 can be derived from those of Eq. 2.6 so that Eq. 2.6 implies Eq. 2.7.

Note that the converse implication can be shown by reversing the argument. Eq. 2.7 contains the equation $\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\tau} = f_{1\mathbf{v}}^T\tau_1 + f_{2\mathbf{v}}^T\tau_2 + f_{3\mathbf{v}}^T\tau_3 = \mathbf{0}$, where $\boldsymbol{\tau} = \left[\tau_1, \tau_2, \tau_3\right]^T$ and $\boldsymbol{\tau}^T\boldsymbol{\tau} - 1 = 0$. Defining $\bar{\xi} = -\tau_3/l_3$ shows that $\tau_3 + l_3\bar{\xi} = 0$. If $\left[\boldsymbol{\tau}^T, \bar{\xi}\right]^T$ is normalized as $\boldsymbol{\tau}^* \equiv \left[\tau_1, \tau_2, \tau_3, \xi\right]^T \equiv \frac{1}{\boldsymbol{\tau}^T\boldsymbol{\tau}+\bar{\xi}^2}\left[\boldsymbol{\tau}^T, \bar{\xi}\right]^T = \frac{1}{1+\bar{\xi}^2}\left[\boldsymbol{\tau}^T, \bar{\xi}\right]^T$,

$$\boldsymbol{\tau}^{*T}\boldsymbol{\tau}^* - 1 = 0 \tag{2.10}$$

$$\tau_3 + l_3\xi = 0 \tag{2.11}$$

and

$$f_{1\mathbf{v}}^T\tau_1 + f_{2\mathbf{v}}^T\tau_2 + f_{3\mathbf{v}}^T\tau_3 = \mathbf{0} \tag{2.12}$$

Thus, Eqs. 2.11 and 2.12, with Eq. 2.4, imply

$$\boldsymbol{\Psi}_{\mathbf{z}^*}^T\boldsymbol{\tau}^* = \mathbf{0}. \tag{2.13}$$

Collecting Eqs. 2.10 and 2.13 and using Eq. 2.4 shows that the components of Eq. 2.6 can be derived from those of Eq. 2.7, so that Eq. 2.7 implies Eq. 2.6.

From this it is clear that Eqs. 2.6 and 2.7 for this example are algebraically equivalent, showing that intersection with plane and boundary criteria commute.

2.3 Intersection Commutes With Boundary Definition

The commutativity result for the serial manipulator example of the previous section can be proven true for general boundary surface calculation. If $n_u = 3$, then define $\mathbf{L} = [l_1, l_2, l_3]^T \in \Re^{n_u}$, which defines the intersection of the three dimensional output space with the plane $\mathbf{L}^T\mathbf{u} + b = 0$. This can be extended for $\mathbf{u}$–spaces with $n_u > 3$ by intersecting with a set of hyperplanes, where $\mathbf{L}$ is a matrix and $\mathbf{b}$ is a vector. The result presented here will be for the case $n_u = 3$, but the extension to $n_u > 3$ is straightforward. As in the previous example, consider the following notation:

$$\mathbf{z} = \left[\mathbf{w}^T, \mathbf{v}^T\right]^T \in \Re^{n_q - n_u}$$
$$\mathbf{z}^* = \left[u_i, \mathbf{z}^T\right]^T \in \Re^{n_q - n_u + 1} \; i = 1,\ 2, \ \text{ or } \ ...\ n_u$$
$$\boldsymbol{\tau}^* = \left[\boldsymbol{\tau}^T, \xi\right]^T \in \Re^{m+1} \text{ where } \boldsymbol{\tau} \in \Re^m \text{ and } \xi \in \Re$$

Next define the following two processes for determining the boundary of the set given in Eq. 2.1.

<u>Process 1</u>: Intersect the set A given by Eq. 2.1 with a plane by appending the equation of the plane to Eq. 1.1. This yields the augmented constraint equations

$$\boldsymbol{\Psi}(\mathbf{q}) = \begin{bmatrix} \boldsymbol{\Phi} \\ \mathbf{L}^T\mathbf{u} + b \end{bmatrix} = \mathbf{0} \tag{2.14}$$

Next, the analytic form of the necessary boundary condition of Eq. 2.3 is applied using $\mathbf{z}^*$ to $\boldsymbol{\Psi}(\mathbf{q}) = \mathbf{0}$; i.e., $\boldsymbol{\Psi}_{\mathbf{z}^*}$ must be row rank deficient, to obtain the one dimensional solution curve of

$$\mathbf{G}^*(\mathbf{q}, \boldsymbol{\tau}^*) = \begin{bmatrix} \boldsymbol{\Psi} \\ \boldsymbol{\Psi}_{\mathbf{z}^*}^T \boldsymbol{\tau}^* \\ \boldsymbol{\tau}^{*T}\boldsymbol{\tau}^* - 1 \end{bmatrix} = \mathbf{0} \tag{2.15}$$

<u>Process 2</u>: Apply the analytic necessary boundary condition of Eq. 2.3 to $\boldsymbol{\Phi}(\mathbf{q}) = \mathbf{0}$, using $\mathbf{z}$ to obtain necessary conditions for ∂A, and then intersect the resulting set with a plane by appending the equation of the plane $\mathbf{L}^T\mathbf{u} + b = 0$. This yields the one dimensional solution curve

$$\mathbf{G}(\mathbf{q}, \boldsymbol{\tau}) = \begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau} \\ \mathbf{L}^T\mathbf{u} + b \\ \boldsymbol{\tau}^T\boldsymbol{\tau} - 1 \end{bmatrix} = \mathbf{0} \tag{2.16}$$

Theorem *Process 1 and Process 2 yield the same set of one dimensional solution* *curves.*

Proof: First consider Process 1: To intersect A prior to using the necessary boundary condition requires that components of $\mathbf{u}$ that are *not* in the reduced two dimensional output set be re-assigned to the intermediate coordinates $\mathbf{w}$. For the case $n_u = 3$, which is considered here, there will be one new intermediate coordinate u_i, where $l_i \neq 0$. Since $\mathbf{L} \neq \mathbf{0}$ without loss of generality, assume that $l_3 \neq 0$ and set $i = 3$.

To expand Eq. 2.15, the required terms are

$$\mathbf{\Psi}_{\mathbf{z}^*} = \begin{bmatrix} \mathbf{\Phi}_{u_3} & \mathbf{\Phi}_{\mathbf{z}} \\ l_3 & \mathbf{0}^T \end{bmatrix}$$

Thus,

$$\mathbf{\Psi}_{\mathbf{z}^*}^T \boldsymbol{\tau}^* = \begin{bmatrix} \mathbf{\Phi}_{u_3}^T \boldsymbol{\tau} + l_3\xi \\ \mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau} + \mathbf{0} \end{bmatrix}$$

The equation $\mathbf{\Psi}_{\mathbf{z}^*}^T \boldsymbol{\tau}^* = \mathbf{0}$ contains a condition similar to one that arises from Process 2, namely $\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau} = \mathbf{0}$, and the additional equation $\mathbf{\Phi}_{u_3}^T \boldsymbol{\tau} + l_3\xi = 0$, which is linear in ξ. From this equation, ξ can be determined as

$$\xi = \frac{1}{l_3}\mathbf{\Phi}_{u_3}^T \boldsymbol{\tau} \tag{2.17}$$

Expanding $\boldsymbol{\tau}^{*T}\boldsymbol{\tau}^* - 1 = 0$ from Eq. 2.15 yields

$$\boldsymbol{\tau}^T\boldsymbol{\tau} = 1 - \xi^2 \tag{2.18}$$

Set $c = 1 - \xi^2$, and show that $c > 0$. First, $\boldsymbol{\tau}^{*T}\boldsymbol{\tau}^* - 1 = 0$ implies that $\xi^2 \leq 1$, so $c \geq 0$. To show $c \neq 0$, suppose $c = 0$. This implies that $\xi^2 = 1$, which implies, using Eq. 2.18, that $\boldsymbol{\tau} = \mathbf{0}$. However inserting $\boldsymbol{\tau} = \mathbf{0}$ into Eq. 2.17 implies $\xi = 0$, which is a contradiction.

With c as above, normalize $\boldsymbol{\tau}$ by setting $\bar{\boldsymbol{\tau}} = \frac{1}{c}\boldsymbol{\tau}$, so that

$$\bar{\boldsymbol{\tau}}^T\bar{\boldsymbol{\tau}} = 1 \tag{2.19}$$

Also note that $\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau} = \mathbf{0}$ implies

$$\mathbf{\Phi}_{\mathbf{z}}^T \bar{\boldsymbol{\tau}} = \mathbf{0} \tag{2.20}$$

Equation 2.14 shows that the first and third components of Eq. 2.16, $\mathbf{\Phi} = \mathbf{0}$ and $\mathbf{L}^T\mathbf{u} + b = 0$, appear exactly as in Eq. 2.15. Note that by setting $\bar{\boldsymbol{\tau}} \equiv \boldsymbol{\tau}$, Eqs. 2.19 and 2.20 show that the remaining components of Eq. 2.16 follow from Eq. 2.15. This completes the proof that Process 1 implies Process 2.

Next consider Process 2: Define

$$\xi = \frac{1}{l_3}\mathbf{\Phi}_{u_3}^T \boldsymbol{\tau} \tag{2.21}$$

Then, $\mathbf{\Phi}_{u_3}^T \boldsymbol{\tau} + l_3\xi = 0$. Setting $\bar{\boldsymbol{\tau}}^* = \left[\boldsymbol{\tau}^T, \xi\right]^T$ and normalizing,

$$\boldsymbol{\tau}^* \equiv \frac{1}{(\bar{\boldsymbol{\tau}}^{*T}\bar{\boldsymbol{\tau}}^*)}\bar{\boldsymbol{\tau}}^* = \frac{1}{1+\xi^2}\left[\boldsymbol{\tau}^T, \xi\right]^T \tag{2.22}$$

Using the definition of $\mathbf{\Psi}$ Eq. 2.14,

$$\frac{1}{1+\xi^2}\begin{bmatrix} \mathbf{\Phi}_{u_3}^T \boldsymbol{\tau} + l_3\xi \\ \mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau} + \mathbf{0} \end{bmatrix} = \frac{1}{1+\xi^2}\begin{bmatrix} \mathbf{\Phi}_{u_3} & \mathbf{\Phi}_{\mathbf{z}} \\ l_3 & \mathbf{0}^T \end{bmatrix}\begin{bmatrix} \boldsymbol{\tau} \\ \xi \end{bmatrix} = \mathbf{\Psi}_{\mathbf{z}^*}^T \boldsymbol{\tau}^* \tag{2.23}$$

Equation 2.14 shows that the first and third components of Eq. 2.16, $\mathbf{\Phi} = \mathbf{0}$ and $\mathbf{L}^T\mathbf{u} + b = 0$, appear exactly as in Eq. 2.15. Equations 2.22 and 2.23 show that the remaining components of Eq. 2.15 follow from Eq. 2.16. This completes the proof that Process 2 implies Process 1.

The result that Process 1 and Process 2 imply each other thus proves that Eq. 2.15 and 2.16 are algebraically equivalent. This shows that the one dimensional boundary curves determined by the two processes are identical. □

The previous theorem proves that the intersection and necessary boundary condition operations can be commuted, to yield identical families of one dimensional

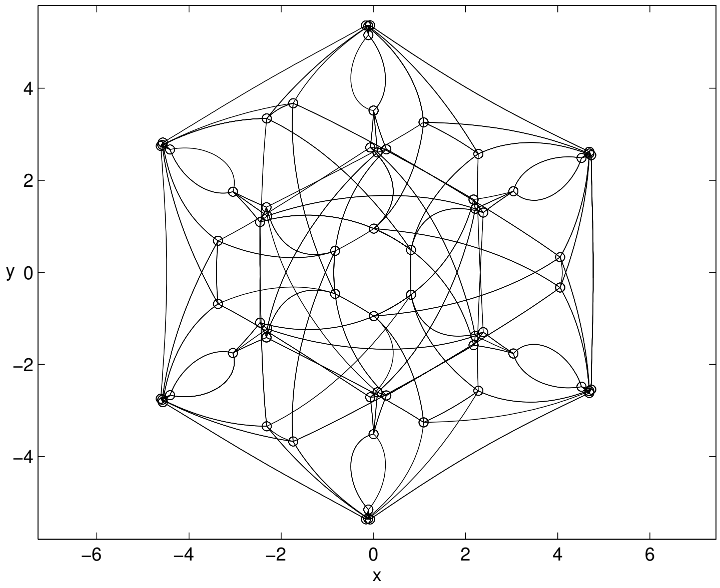

Figure 2.1: Singular Curves at z = 2.23 Meters

boundary curves.

2.4 Numerical Example– Spatial Stewart Platform

Recall the spatial Stewart platform [89] described in Section 1.4.2. In order to compute the boundary of the accessible output set using Process 2, as described in Section 2.3, the condition for row rank deficiency of the sub-Jacobian matrix $\mathbf{\Phi_{w,v}}$ is augmented with the constraint equations $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$. Recall that the necessary condition for points on the boundary of the accessible output set is given by Eq. 2.3.

Since $\partial A \subset \Re^3$, a family of horizontal cutting planes is introduced as $z - b = 0$. For the value $b = 2.23$ meters, the accessible output set is computed using both mathematical models which resulting from the two processes of Section 2.3. The numerical continuation methods presented in Ref. 40 were used to trace one dimensional curves, and two different techniques were used to calculate branches at

Table 2.1: Performance in CPU seconds of Different Models

	Process 1	Process 2
Method of Ref. 40 with Chapter 3	System 4.563e+1 User 9.496e+2 Total 9.952e+2	System 3.776e+1 User 8.780e+2 Total 9.158e+2
Method of Ref. 40 with Ref. 78	System 5.282e+1 User 1.058e+3 Total 1.110e+3	Method does not apply

bifurcation points. The model resulting from Process 1 was computed using the bifurcation point method of both Refs. 78 and the new technique given in Chapter 3 of this thesis. For Process 2 only the method of Chapter 3 applies for calculating bifurcation branch directions.

Table 2.1 contains a summary of the computational times for the various methods when run on a Hewlett Packard 9000 series 755 workstation. Times presented are in seconds and reflect computation of only the singular curves that comprise ∂A in this cutting plane. Note that all singular curves of S in this cutting plane are shown in Fig. 2.1. This table shows that the most efficient computational method is to use the bifurcation method of Chapter 3 with the model of Process 2. This method is 17.5% faster than using the model of Process 1 with the bifurcation method of Ref. 78. Within the same method, the effect of adding the extra equation and variable of Process 1 takes 8.67% more computation time, showing that Process 2 is the faster modelling method. These ratios are problem dependent, but this test problem is fairly complex and one would expect the addition of the extra equation and variable of Process 1 to have a larger effect on smaller problems.

2.5 Conclusions

The technique for reducing the dimension of the boundary of the working domain to a family of one dimensional curves in two dimensional planes is used in Refs. 33, 34, 35, 36, and 40. It is important that the two processes presented, while giving different sets of equations to define one dimensional boundary curves, actually produce the same solution set.

The mathematical model resulting from Process 1 yields the continuation equation for which the method of Qiu, Luh, and Haug [78] applies to compute tangents at bifurcation points. The method presented in Ref. 78 can only be applied only to the continuation equation of Eq. 2.15 which results from Process 1. The results of this chapter show that it is sufficient to formulate the numerical continuation equations using Process 1 and apply the methods of Refs. 40 and 78 to determine tangents at bifurcation points. However, the continuation equation, Eq. 2.16, from Process 2 has one less variable and one less equation than in Process 1. As shown in Section 2.4, this results in computational savings. Thus, it is suggested to use the numerical continuation techniques of Ref. 40 and bifurcation branch computation from Chapter 3, which apply to both models of Processes 1 and 2.

CHAPTER 3
REDUCED DIMENSION METHOD FOR EFFICIENTLY DETERMINING ALL BRANCHES AT BIFURCATION POINTS

A numerical method for obtaining all tangent vectors for a class of bifurcation points of continuation curves that arise in workspace analysis of manipulators and mechanisms is presented in Qiu, Luh, and Haug [78]. For the case in which the dimension of the output set of the workspace is greater than two, it is shown in Chapter 2 that there are two equivalent formulations for the continuation curves. The two formulations produce the same solution set, yet differ in that one has an additional equation and an additional variable. The formulation with the additional variable and equation is the one for which the bifurcation branch determination method of Ref. 78 applies. This section gives an efficient extension to the method of Ref. 78 which applies to both formulations. A new formulation for the tangents and a proof that certain matrices arising in Ref. 78 are invertible is the basis for a simplified and more efficient numerical algorithm for their computation. Second order necessary conditions for tangents are employed to form a system of second order polynomial equations that is solved using HOMPACK [99]. The analytic criteria for the continuation curves given in Refs. 33, 34, 35, 40, and 41, together with the extension of this section and the numerical methods in Refs. 40 and 97, provides a systematic method for mapping boundaries of manipulator workspaces.

3.1 Motivation

A variety of workspace formulations are possible to determine physical properties of mechanisms and manipulators. These include accessible output sets [40, 49], dexterous accessible output sets [41], operational envelopes [34, 35], and domains of interference [33, 34]. These formulations all have in common a collection of equations that govern the kinematics of the mechanism and additional equations that determine either performance requirements or the shape of a working body. These equations, assuming slack variables or other methods have been used to reduce inequality equations to equalities, define the constraint equations $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$, Eq. 1.1.

Note that if $n_u = 3$ then ∂A is a surface in $\Re^3$. A practical approach used to map this boundary in Ref. 40 is to intersect it with a family of planes and then find the corresponding one dimensional curves in each plane. In Chapter 2 it was shown that intersecting the set defined by Eq. 1.4 with a family of planes is equivalent to first intersecting the solution set of Eq. 1.2 with a family of planes and then applying the analytical necessary condition to determine the boundary of the sets in each plane. The method of first intersecting the solution set of Eq. 1.2 with a family of planes is the one for which the method of Ref. 78 for determining tangents at bifurcation points applies. However, this method also results in an additional equation and variable in the continuation calculation.

If a method were available to calculate branches at bifurcation points for the set of Eq. 1.4, intersected with a family of planes, then the continuation calculation can be done without first adding the intersection equation to the set of equations $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$ as described in Chapter 2. This would avoid unnecessarily inflating both the dimension of $\boldsymbol{\tau}$ and the number of continuation equations. The purpose of this chapter is to develop such a method. This chapter also presents a new formulation for finding tangents that is more efficient than previous methods and applies in cases of output space dimension two or greater.

3.2 Finding all Solution Branches at a Bifurcation Point

Extracting the conditional equations defining the set from Eq. 1.4 and adding a cutting plane equation $\mathbf{L}^T\mathbf{u} + d$ yields the continuation equation

$$\mathbf{G}(\mathbf{q}, \boldsymbol{\tau}) = \begin{bmatrix} \boldsymbol{\Phi}(\mathbf{q}) \\ \mathbf{L}^T\mathbf{u} + d \\ \boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q})\boldsymbol{\tau} \\ \boldsymbol{\tau}^T\boldsymbol{\tau} - 1 \end{bmatrix} = \mathbf{0} \tag{3.1}$$

In order to include the cutting plane equation in bifurcation calculation, but not in the constraint equations, consider adding it to the necessary condition for the boundary; i.e.,

$$\begin{bmatrix} \mathbf{L}^T\mathbf{u} + d \\ \boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q})\boldsymbol{\tau} \end{bmatrix} = \mathbf{0}$$

where $\boldsymbol{\Phi}(\mathbf{q}) = \mathbf{0}$ and $\boldsymbol{\tau}^T\boldsymbol{\tau} - 1 = 0$ are to be included as additional equations, yielding

$$\mathbf{G}(\mathbf{q}, \boldsymbol{\tau}) = \begin{bmatrix} \boldsymbol{\Phi}(\mathbf{q}) \\ \begin{bmatrix} \mathbf{L}^T\mathbf{u} + d \\ \boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q})\boldsymbol{\tau} \end{bmatrix} \\ \boldsymbol{\tau}^T\boldsymbol{\tau} - 1 \end{bmatrix} = \mathbf{0} \tag{3.2}$$

As presented in Ref. 97, when the matrix $\mathbf{G}_{\mathbf{x}}(\mathbf{q}, \boldsymbol{\tau})$, where $\mathbf{x} = \left[\mathbf{q}^T, \boldsymbol{\tau}^T\right]^T$, has full row rank a continuation method can be used to map the boundary curves.

However, points are encountered at which $\mathbf{G_x}(\mathbf{q},\boldsymbol{\tau})$ is not of full row rank. These points are called bifurcation points. In Ref. 40, the case in which $\mathbf{G_x}(\mathbf{q},\boldsymbol{\tau})$ is row rank deficient one is treated and Ref. 78 gives an improved scheme for the special case in which $\mathbf{G_x}(\mathbf{q},\boldsymbol{\tau})$ is row rank deficient and $\boldsymbol{\Phi}_\mathbf{z}(\mathbf{q})$ is row rank deficient two. From computational experience with real applications, these possibilities occur often. For the case in which $\boldsymbol{\Phi}_\mathbf{z}(\mathbf{q})$ is row rank deficient two, there is a family of values of $\boldsymbol{\tau}$ such that $\boldsymbol{\Phi}_\mathbf{z}^T\boldsymbol{\tau} = \mathbf{0}$ and $\boldsymbol{\tau}^T\boldsymbol{\tau} - 1 = 0$. It is a significant task to determine which of the $\boldsymbol{\tau}$ values from the family produce solutions branches of $\mathbf{G}(\mathbf{q},\boldsymbol{\tau}) = \mathbf{0}$ that vary in the $\mathbf{q}-$components.

As noted in Section 3.1, the purpose of this chapter is to extend the method of Ref. 78 to determine the values of $\boldsymbol{\tau}$ and the tangents to the continuation curves for the case in which a cutting plane has been introduced to reduce the continuation process to tracing a family one dimensional curves in a family of two dimensional planes, without the extra component of $\boldsymbol{\tau}$ and extra equation that result from adding the cutting plane equation to the constraint equations. The results presented here also apply to the case in which no cutting plane is needed, and the new algebraic method for computing tangents also represents a computational advantage over the method of Ref. 78. Removing the cutting plane equation and related terms from the following development is straight forward and all proofs presented here are both applicable and simplified by this reduction.

3.2.1 Partitioning of the Generalized Coordinates

When using the necessary condition for a boundary curve, there is not always a clear distinction, from physical or other points of view, that allows selecting input and intermediate coordinates. When no natural partitioning of the coordinates exists, the following mathematical partitioning is used. Such a partition is necessary for computations in Section 3.2.2, and the existence of such a partition is guaranteed by the fact that the constraint equation of Eq. 1.1, and thus the kinematic formulation for the underlying mechanism should be composed of independent equations, except at certain critical points or surfaces in the operating space of the mechanism.

If the process of partitioning $\mathbf{q}$ into $\mathbf{u}, \mathbf{w}$, and $\mathbf{v}$ is not natural, define $\mathbf{z}$ as the vector generalized coordinates other than the output coordinates $\mathbf{u}$. The following process will then be used to partition $\mathbf{z}$. If $\mathbf{u}^* \in A$, then there is a $\mathbf{z}^*$ such that $\mathbf{\Phi}(\mathbf{u}^*, \mathbf{z}^*) = \mathbf{0}$. If $\mathbf{\Phi_q}(\mathbf{u}^*, \mathbf{z}^*)$ has full row rank; i.e., if rank $\mathbf{\Phi_q}(\mathbf{u}^*, \mathbf{z}^*) = m$, for $m \leq n_z$, then there exist projections $\mathbf{v} = \mathbf{Sz}$ and $\mathbf{w} = \mathbf{Tz}$, where $\mathbf{S}$ and $\mathbf{T}$ are Boolean matrices with a single unit value in each row, such that $\mathbf{w}$ is of dimension $m - n_u$, $\mathbf{v}$ is of dimension $n_z - n_w$, and $|\mathbf{\Phi_y}(\mathbf{u}^*, \mathbf{v}^*, \mathbf{w}^*)| \neq \mathbf{0}$, where $\mathbf{y} = \left[\mathbf{u}^T, \mathbf{w}^T\right]^T$. The goal is to find $\mathbf{w}$ from the components of $\mathbf{z}$ such that when Gaussian elimination is performed on $\mathbf{\Phi_q}(\mathbf{u}^*, \mathbf{z}^*)$, with pivoting not done on the columns associated with $\mathbf{u}$, $\mathbf{w}$ will be selected as the first $m - n_u$ components of the pivoted result from $\mathbf{z}$. By the implicit function theorem, there exists a continuously differentiable function $\mathbf{f}(\mathbf{v})$ and neighborhoods $\mathbf{Y}$ and $\mathbf{V}$ of $\mathbf{y}^*$ and $\mathbf{v}^*$, respectively, with $\mathbf{y} = \mathbf{f}(\mathbf{v})$ such that $\mathbf{\Phi}(\mathbf{f}(\mathbf{v}), \mathbf{v}) = \mathbf{0}$, for all $\mathbf{y} \in \mathbf{Y}$ and $\mathbf{v} \in \mathbf{V}$. It will be assumed in the calculations below that a partitioning of the generalized coordinate $\mathbf{q}$ has been determined such that for $\mathbf{y} = \left[\mathbf{u}^T, \mathbf{w}^T\right]^T$, $\mathbf{\Phi_y}(\mathbf{q})$ is invertible.

The general context for this chapter includes the case in which the solution set for the mapping of boundary segments is of dimension greater than two, and a family of cutting planes is introduced. Addition of the cutting plane, as in Eq. 3.1, reduces the number of degrees of freedom of the mechanism by restricting the output to a plane. This requires that the number of input coordinates in $\mathbf{v}$ be reduced and the number of intermediate coordinates $\mathbf{w}$, be increased. Even with this change, the dimension of $\mathbf{z}$ remains the same. Thus, the row rank deficiency condition for boundary curves remains the same. However, note that the partitioning described above may need to be modified to reflect this change. The selection of the cutting plane has an influence on the partitioning of $\mathbf{z} = \left[\mathbf{w}^T, \mathbf{v}^T\right]^T$. While details of this influence are presented in Section 3.2.2, assume that the partitioning has been done and that $\bar{\mathbf{y}} = \left[\mathbf{u}^T, \bar{\mathbf{w}}^T\right]^T$, where $\bar{\mathbf{w}} = \left[\mathbf{w}^T, v_i\right]^T$ and $\bar{\mathbf{v}} = [v_1, ..., v_{i-1}, v_{i+1}, ..., v_{n_v}]^T$. With these definitions,

$$\mathbf{G}_{\mathbf{x}}(\mathbf{q}, \boldsymbol{\tau}) = \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}}(\mathbf{q}) & \boldsymbol{\Phi}_{\bar{\mathbf{v}}}(\mathbf{q}) & \mathbf{0} \\ \begin{bmatrix} \left[\mathbf{L}^T, \mathbf{0}\right] \\ (\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q})\boldsymbol{\tau})_{\bar{\mathbf{y}}} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ (\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q})\boldsymbol{\tau})_{\bar{\mathbf{v}}} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q}) \end{bmatrix} \\ \mathbf{0} & \mathbf{0} & 2\boldsymbol{\tau}^T \end{bmatrix} \tag{3.3}$$

3.2.2 Finding All Solutions for $\boldsymbol{\tau}$ at a Bifurcation Point

The case considered here is such that $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau})$ is row rank deficient and $\boldsymbol{\Phi}_{\mathbf{z}}(\mathbf{q}^*)$ is row rank deficient two. As previously noted, the results also hold for the case in which the cutting plane equation, $\mathbf{L}^T\mathbf{u} - b = 0$, is not present. For this case, there exist vectors $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ such that

$$\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q}^*)\boldsymbol{\tau}_i = \mathbf{0} \tag{3.4a}$$

$$\boldsymbol{\tau}_i^T\boldsymbol{\tau}_j = \delta_{ij}, \quad i, j = 1, 2 \tag{3.4b}$$

Every unit vector $\boldsymbol{\tau}$ in $\ker(\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q}^*))$ can be represented as a linear combination of the orthonormal vectors $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$, in the form

$$\boldsymbol{\tau} \equiv \boldsymbol{\tau}(\phi) = \sin(\phi)\boldsymbol{\tau}_1 + \cos(\phi)\boldsymbol{\tau}_2 \tag{3.5}$$

where $\frac{-\pi}{2} \leq \phi < \frac{3\pi}{2}$. Any $\boldsymbol{\tau}$ of this form satisfies $\mathbf{G}(\mathbf{q}^*, \boldsymbol{\tau}) = \mathbf{0}$. The goal is to find values $\left[\mathbf{q}^{*T}, \boldsymbol{\tau}^T\right]^T$ from which there are non-trivial solution branches; i.e., branches that vary in the $\mathbf{q}^*$ components. These branches occur at bifurcation points and this requires that $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau})$ is row rank deficient.

The matrix $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau})$ is row rank deficient if and only if there exists a non-zero vector $\mathbf{c} = \left[\mathbf{c}_1^T, \mathbf{c}_2^T, c_3\right]^T$, where $\mathbf{c}_1 \in \Re^m, \mathbf{c}_2 \in \Re^{n_z+1}$, and $c_3 \in \Re$, such that

$$\mathbf{c}^T \mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}) = \mathbf{0} \tag{3.6}$$

Denoting $\boldsymbol{\Phi}_{\mathbf{z}} \equiv \boldsymbol{\Phi}_{\mathbf{z}}(\mathbf{q}^*)$, $\boldsymbol{\Phi}_{\bar{\mathbf{y}}} \equiv \boldsymbol{\Phi}_{\bar{\mathbf{y}}}(\mathbf{q}^*)$, and $\boldsymbol{\Phi}_{\bar{\mathbf{v}}} \equiv \boldsymbol{\Phi}_{\bar{\mathbf{v}}}(\mathbf{q}^*)$, and expanding Eq. 3.6 using Eq. 3.3,

$$\mathbf{c}_1^T \boldsymbol{\Phi}_{\bar{\mathbf{y}}} + \mathbf{c}_2^T \begin{bmatrix} \left[\mathbf{L}^T, \mathbf{0}\right] \\ \left(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}\right)_{\bar{\mathbf{y}}} \end{bmatrix} = \mathbf{0} \tag{3.7a}$$

$$\mathbf{c}_1^T \boldsymbol{\Phi}_{\bar{\mathbf{v}}} + \mathbf{c}_2^T \begin{bmatrix} \mathbf{0} \\ \left(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}\right)_{\bar{\mathbf{v}}} \end{bmatrix} = \mathbf{0} \tag{3.7b}$$

$$\mathbf{c}_2^T \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\Phi}_{\mathbf{z}}^T \end{bmatrix} + 2c_3 \boldsymbol{\tau}^T = \mathbf{0} \tag{3.7c}$$

If $\mathbf{c}_2^T = \left[c_{21}, \mathbf{c}_{22}^T\right]$, then Eqs. 3.7a, 3.7b, and 3.7c can be rewritten as

$$\mathbf{c}_1^T \boldsymbol{\Phi}_{\bar{\mathbf{y}}} + \left(\left[c_{21}, -\mathbf{c}_{22}^T\right]\right) \begin{bmatrix} \left[\mathbf{L}^T, \mathbf{0}\right] \\ \left(\boldsymbol{\Phi}_{\mathbf{z}}^T(-\boldsymbol{\tau})\right)_{\bar{\mathbf{y}}} \end{bmatrix} = \mathbf{0} \tag{3.8a}$$

$$\mathbf{c}_1^T \boldsymbol{\Phi}_{\bar{\mathbf{v}}} + \left(\left[c_{21}, -\mathbf{c}_{22}^T\right]\right) \begin{bmatrix} \mathbf{0} \\ \left(\boldsymbol{\Phi}_{\mathbf{z}}^T(-\boldsymbol{\tau})\right)_{\bar{\mathbf{v}}} \end{bmatrix} = \mathbf{0} \tag{3.8b}$$

$$\left(\left[c_{21}, -\mathbf{c}_{22}^T\right]\right)\begin{bmatrix}\mathbf{0}\\ \boldsymbol{\Phi}_{\mathbf{z}}^T\end{bmatrix} + 2c_3(-\boldsymbol{\tau}^T) = \mathbf{0} \tag{3.8c}$$

which is equivalent to $\left[\mathbf{c}_1^T, c_{21}, -\mathbf{c}_{22}^T, c_3\right]^T \mathbf{G_x}(\mathbf{q}^*, -\boldsymbol{\tau}) = \mathbf{0}$. However this is equivalent to showing that $\mathbf{G_x}(\mathbf{q}^*, -\boldsymbol{\tau})$ is row rank deficient. This proves that if $\boldsymbol{\tau}$ is a solution, then $-\boldsymbol{\tau}$ is also a solution, which restricts the search range on ϕ to $\frac{-\pi}{2} \leq \phi < \frac{\pi}{2}$. It will be shown that there are a finite number of solutions for $\boldsymbol{\tau}$. Thus, it is possible to choose $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ such that $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_1)$ has full row rank. Requiring $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_1)$ to be full row rank excludes the value $\phi = -\pi/2$ from the search range, so $\cos(\phi) \neq 0$. This allows the introduction of

$$\lambda \equiv \lambda(\phi) = tan(\phi) = \frac{\sin(\phi)}{\cos(\phi)}. \tag{3.9}$$

The problem may then be restated as follows.

Problem 3.2.1 *If $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_1)$ has full row rank, find all ϕ or $\lambda(\phi)$, $-\pi/2 < \phi < \pi/2$, such that $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ is row rank deficient.*

A necessary and sufficient condition for $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ to be row rank deficient is provided by Eqs. 3.7a, 3.7b, and 3.7c. Multiplying Eq. 3.7c on the right by $\boldsymbol{\tau}$ and using the fact that $\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\tau} = \mathbf{0}$ shows that $c_3 = 0$. Thus, from Eq. 3.7c, $\boldsymbol{\Phi}_{\mathbf{z}}\mathbf{c}_{22} = \mathbf{0}$. Since $\boldsymbol{\Phi}_{\mathbf{z}} : \Re^{n_z} \rightarrow \Re^m$ has row rank deficiency two, $\dim(\ker(\boldsymbol{\Phi}_{\mathbf{z}})) = n_z + 2 - m \equiv k$, and there exists an orthonormal basis of $\ker(\boldsymbol{\Phi}_{\mathbf{z}})$, denoted as $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k\}$, such that

$$\mathbf{c}_{22} = d_1\boldsymbol{\alpha}_1 + d_2\boldsymbol{\alpha}_2 + ... + d_k\boldsymbol{\alpha}_k = \boldsymbol{\alpha}\mathbf{d} \tag{3.10}$$

where $\mathbf{d} = [d_1, d_2, ..., d_k]^T$ and $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k]$.

Expanding the second term of Eq. 3.7a

$$\mathbf{c}_1^T \mathbf{\Phi}_{\bar{\mathbf{y}}} + c_{21} \left[\mathbf{L}^T, \mathbf{0}\right] + \mathbf{c}_{22}^T (\mathbf{\Phi}_{\mathbf{z}}^T(\boldsymbol{\tau}))_{\bar{\mathbf{y}}} = \mathbf{0} \tag{3.11}$$

and combining terms yields

$$\left[\mathbf{c}_1^T, c_{21}\right] \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix} + \mathbf{c}_{22}^T (\mathbf{\Phi}_{\mathbf{z}}^T(\boldsymbol{\tau}))_{\bar{\mathbf{y}}} = \mathbf{0} \tag{3.12}$$

Note that the matrix

$$\begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{y}} & \mathbf{\Phi}_{v_i} \\ \left[\mathbf{L}^T, \mathbf{0}\right] & 0 \end{bmatrix} \tag{3.13}$$

is square, of dimension $(m+1) \times (m+1)$, and under the following reasonable conditions will be invertible. Recall that $\mathbf{y}$ was selected so that $\mathbf{\Phi}_{\mathbf{y}}$ has full row rank. Augmenting $\mathbf{y}$ with v_i, to form $\bar{\mathbf{y}} \equiv \left[\mathbf{y}^T, v_i\right]^T$ for some i, $\mathbf{\Phi}_{\bar{\mathbf{y}}} = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{y}} & \mathbf{\Phi}_{v_i} \end{bmatrix}$ has full row rank, since no additional rows were added. To augment this matrix with the row $\begin{bmatrix} \left[\mathbf{L}^T, \mathbf{0}\right] & 0 \end{bmatrix}$ requires that there is some v_i such that $\begin{bmatrix} \mathbf{\Phi}_{v_i} \\ 0 \end{bmatrix}$ is not a linear combination of the columns of $\begin{bmatrix} \mathbf{\Phi}_{\mathbf{y}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}$. Since this condition depends on $\mathbf{L}^T$, the selection of v_i will depend on the cutting plane chosen. In fact, for some $\mathbf{L}^T$, there may be no i such that the matrix of Eq. 3.13 has full row rank. Note also that the nonzero components of $\begin{bmatrix} \left[\mathbf{L}^T, \mathbf{0}\right] & 0 \end{bmatrix}$ correspond to the elements of $\mathbf{u}$ that must be in $\mathbf{y}$ and $\bar{\mathbf{y}}$. Thus the components of $\mathbf{y}$ (selected so that $\mathbf{\Phi}_{\mathbf{y}}$ is invertible) will remain in $\bar{\mathbf{y}}$, and the task is to find the additional component v_i to augment $\mathbf{y}$. Assume that $\mathbf{L}^T$ is such that there is some i such that Eq. 3.13 is invertible, so $\bar{\mathbf{y}} \equiv \left[\mathbf{y}^T, v_i\right]^T$ is set. A practical method for determining the v_i is by carrying out Gaussian elimination on

$$\begin{bmatrix} \mathbf{\Phi}_{\mathbf{q}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}$$

leaving the columns associated with $\mathbf{u}$ fixed and doing column pivoting on all remaining columns. Select $\bar{\mathbf{y}}$ as the first $m+1$ generalized coordinates that correspond to the columns after pivoting, and check that the matrix of Eq. 3.13 is invertible.

From Eq. 3.12 and the fact that the matrix in Eq. 3.13 is invertible,

$$\left[\mathbf{c}_1^T, c_{21}\right] = -\mathbf{c}_{22}^T (\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau})_{\bar{\mathbf{y}}} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \tag{3.14}$$

Substituting $\left[\mathbf{c}_1^T, c_{21}\right]$ from Eq. 3.14 into Eq. 3.7b yields

$$\mathbf{c}_{22}^T \left[-(\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau})_{\bar{\mathbf{y}}} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} + (\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau})_{\bar{\mathbf{v}}} \right] = \mathbf{0} \tag{3.15}$$

Substituting $\mathbf{c}_{22}^T$ from Eq. 3.10 into Eq. 3.15 yields

$$\mathbf{d}^T \boldsymbol{\alpha}^T \left[-(\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau})_{\bar{\mathbf{y}}} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} + (\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau})_{\bar{\mathbf{v}}} \right] = \mathbf{0}. \tag{3.16}$$

Now define

$$\mathbf{H}(\boldsymbol{\tau}(\phi)) \equiv \boldsymbol{\alpha}^T \left[-(\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi))_{\bar{\mathbf{y}}} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} + (\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi))_{\bar{\mathbf{v}}} \right] \tag{3.17}$$

Theorem 3.2.1 *The matrix $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ is row rank deficient of order p, if and only if $\mathbf{H}(\boldsymbol{\tau}(\phi))$ of Eq. 3.17 is row rank deficient of order p.*

Proof: A necessary and sufficient condition for a matrix $\mathbf{T}$ to be row rank deficient of order p is that there exist exactly p mutually independent vectors, $\mathbf{e}_i, i = 1, ..., p$, such that $\mathbf{e}_i^T \mathbf{T} = \mathbf{0}$.

First, assume that $\mathbf{H}(\boldsymbol{\tau}(\phi))$ is row rank deficient of order p. Then there exist independent vectors $\mathbf{d}^{(i)}, i = 1, ..., p$, such that $\mathbf{d}^{(i)^T}\mathbf{H}(\boldsymbol{\tau}(\phi)) = \mathbf{0}$. Evaluate $\mathbf{c}_{22}^{(i)} = \boldsymbol{\alpha}\mathbf{d}^{(i)}$, as in Eq. 3.10. From Eq. 3.14, evaluate $\mathbf{c}_1^{(i)}$ and $c_{21}^{(i)}$ and define $\mathbf{c}^{(i)} = \left[\mathbf{c}_1^{(i)^T}, c_{21}^{(i)}, \mathbf{c}_{22}^{(i)^T}, 0\right]^T$. By the construction preceding this theorem, $\mathbf{c}^{(i)^T}\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi)) = \mathbf{0}$. This shows that the row rank deficiency of $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ is at least p.

To show that the row rank deficiency of $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ is exactly p, suppose that its row rank deficiency is greater than p. Then there exists a vector $\mathbf{c}$ that is not a linear combination of $\mathbf{c}^{(i)}, i = 1, ..., p$, such that $\mathbf{c}^T\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi)) = \mathbf{0}$. The construction preceding this theorem then guarantees the existence of a $\mathbf{d}$ that is not a linear combination of $\mathbf{d}^{(i)}, i = 1, ..., p$, such that $\mathbf{d}^T\mathbf{H}(\boldsymbol{\tau}(\phi)) = \mathbf{0}$, which contradicts the supposition that $\mathbf{H}(\boldsymbol{\tau}(\phi))$ is row rank deficient of order p.

Second, assume that $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ is row rank deficient of order p. Then there exist independent vectors $\mathbf{c}^{(i)}, i = 1, ..., p$, such that $\mathbf{c}^{(i)^T}\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi)) = \mathbf{0}$. The subvectors $\mathbf{c}_{22}^{(i)}, i = 1, ..., p$, are independent, since $c_3^{(i)} = 0$ and if $\mathbf{c}_{22}^{(i)}, i = 1, ..., p$, were not independent, then Eq. 3.14 would imply $\left[\mathbf{c}_1^{(i)^T}, c_{21}^{(i)}\right]^T, i = 1, ..., p$, were also not independent, a contradiction to $\mathbf{c}^{(i)}, i = 1, ..., p$, being independent. Evaluate $\mathbf{d}^{(i)}$ such that $\mathbf{c}_{22}^{(i)} = \boldsymbol{\alpha}\mathbf{d}^{(i)}$. Since $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k\}$ is an orthonormal basis for $\ker(\boldsymbol{\Phi}_{\mathbf{z}})$ and Eq. 3.7c requires $\boldsymbol{\Phi}_{\mathbf{z}}\mathbf{c}_{22}^{(i)} = \mathbf{0}$, then the choice for $\mathbf{d}^{(i)}$ is unique. Equation 3.16 implies, by construction, that $\mathbf{d}^{(i)^T}\mathbf{H}(\boldsymbol{\tau}(\phi)) = \mathbf{0}$, so $\mathbf{H}$ is at least row rank deficient of order p.

To see that $\mathbf{H}$ is exactly row rank deficient of order p, suppose $\mathbf{H}$ is row rank deficient of order greater than p. Then there exists a $\mathbf{d}$ that is not a linear combination of $\mathbf{d}^{(i)}, i = 1, ..., p$, such that $\mathbf{d}^T\mathbf{H}(\boldsymbol{\tau}(\phi)) = \mathbf{0}$. However, set $\mathbf{c}_{22} = \boldsymbol{\alpha}\mathbf{d}$. Since $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k\}$ is an orthonormal basis for $\ker(\Phi_z)$, $\mathbf{c}_{22}$ is not a linear combination of of $\mathbf{c}_{22}^{(i)}, i = 1, ..., p$. Evaluating $\left[\mathbf{c}_1{}^T, c_{21}\right]^T$ from Eq. 3.14 implies that

$\mathbf{c} = \left[\mathbf{c}_1{}^T, c_{21}, \mathbf{c}_{22}{}^T, 0\right]^T$, not a linear combination of $\mathbf{c}^{(i)}, i = 1, ..., p$, is such that $\mathbf{c}^T \mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi)) = \mathbf{0}$. This is a contradiction to $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ being row rank deficient of order p. This completes the proof of Theorem 3.2.1. □

To use this result, substituting Eq. 3.5 into Eq. 3.17 yields

$$\begin{aligned}\mathbf{H}(\boldsymbol{\tau}(\phi)) = \sin(\phi)\boldsymbol{\alpha}^T &\left[-(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}_1)_{\bar{\mathbf{y}}} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}_1)_{\bar{\mathbf{v}}} \right] \\ + \cos(\phi)\boldsymbol{\alpha}^T &\left[-(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}_2)_{\bar{\mathbf{y}}} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}_2)_{\bar{\mathbf{v}}} \right]\end{aligned} \tag{3.18}$$

Defining

$$\mathbf{A} = \boldsymbol{\alpha}^T \left[-(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}_1)_{\bar{\mathbf{y}}} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}_1)_{\bar{\mathbf{v}}} \right] \tag{3.19a}$$

$$\mathbf{B} = \boldsymbol{\alpha}^T \left[-(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}_2)_{\bar{\mathbf{y}}} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}_2)_{\bar{\mathbf{v}}} \right] \tag{3.19b}$$

Eq. 3.18 can be written as

$$\mathbf{H}(\boldsymbol{\tau}(\phi)) = \sin(\phi)\mathbf{A} + \cos(\phi)\mathbf{B} \tag{3.20}$$

Theorem 3.2.2 *The matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *of Eq. 3.19 are square of dimension* $n_{\bar{v}} \times n_{\bar{v}}$.

Proof: By construction, $\mathbf{A}$ and $\mathbf{B}$ both have dimension $(n_z - (m-2)) \times n_{\bar{v}}$. To prove the result, it is necessary to show that $n_z - (m-2) = n_{\bar{v}}$. This will be done by considering two cases. First, if $n_u = 2$, then no cutting plane is introduced and the partitioning of generalized coordinates makes no distinction between n_v and $n_{\bar{v}}$.

Also, the number m of constraint equations is $n_u + n_w$. Thus, for the case $n_u = 2$, $m - 2 = n_w$. Using the fact that $n_z = n_w + n_v$ yields $n_z - (m-2) = (n_w + n_v) - n_w = n_v \equiv n_{\bar{v}}$, which is the desired result.

Second, for the case $n_u \geq 3$, $n_u - 2$ cutting planes are added and the partitioning process shifts $n_u - 2$ coordinates from $\mathbf{v}$ to $\bar{\mathbf{w}}$, so $n_{\bar{v}} = n_v - (n_u - 2)$. This justifies the computation $n_z - (m-2) = (n_w + n_v) - ((n_u + n_w) - 2) = n_v - (n_u - 2) = n_{\bar{v}}$, which completes the proof. $\square$

Theorem 3.2.3 *If* $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_1)$ *has full row rank then* A *is invertible.*

Proof: If $\phi = 0$, then $\boldsymbol{\tau}(\phi) = \boldsymbol{\tau}_1$, $\mathbf{H} = \mathbf{A}$, and $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}(\phi)) = \mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_1)$ has full row rank. Theorem 3.2.1 then implies that $\mathbf{H} = \mathbf{A}$ must also have full row rank. By Theorem 3.2.2, $\mathbf{A}$ is square so it is invertible. This completes the proof. $\square$

Finally, note that the $\boldsymbol{\tau}(\phi)$ being sought satisfies the condition $\mathbf{H}(\boldsymbol{\tau}(\phi)) = \sin(\phi)\mathbf{A} + \cos(\phi)\mathbf{B}$ is row rank deficient. Choosing $\boldsymbol{\tau}_1$ such that $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_1)$ is full row rank, multiplying by $\mathbf{A}^{-1}$, and dividing by $\cos(\phi)$ shows that this is equivalent to requiring that either $\lambda(\phi)\mathbf{I} + \mathbf{BA}^{-1}$ or $\lambda(\phi)\mathbf{I} + \mathbf{A}^{-1}\mathbf{B}$ is row rank deficient, since rank is not changed as a result of multiplication by a non-singular matrix [72]. Since both $\mathbf{BA}^{-1}$ and $\mathbf{A}^{-1}\mathbf{B}$ are square matrices, $\lambda(\phi)$ is an eigenvalue of either $-\mathbf{BA}^{-1}$ or $-\mathbf{A}^{-1}\mathbf{B}$

Note that the dimension of $\mathbf{A}$ and $\mathbf{B}$ is $(n_z + 2 - m) \times (n_{\bar{v}}))$, so the dimension of $\mathbf{BA}^{-1}$ is $(n_z + 2 - m) \times (n_z + 2 - m)$. The dimension of $\mathbf{BA}^{-1}$ is the same as the matrix $\mathbf{BA}^{+}$ derived in Ref. 78, if the cutting plane was not included in the constraint equations. However, there $\mathbf{A}$ and $\mathbf{B}$ have different dimension and $\mathbf{A}^{+}$ is the Moore-Penrose pseudo-inverse of $\mathbf{A}$. This change reflects the fact that the $\mathbf{A}$ and $\mathbf{B}$ are formulated to produce continuation curve tangents consistent with the cutting plane.

3.2.3 Selecting Which Eigenvalue Problem to Use

In the previous section it was proved that there are two eigenvalue problems that yield the values of λ for which the corresponding values of $\boldsymbol{\tau}$ cause $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ to be row rank deficient. Once the values of $\boldsymbol{\tau}$ are found, the $\mathbf{q}$-space tangents to continuation curves must be found. One of the two eigenvalue problems leads to an efficient way to compute these tangents, as the following derivation will show. The other will be used in Section 3.2.5.

A tangent $\boldsymbol{\eta}$ to a continuation curve must satisfy the first order necessary condition $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))\boldsymbol{\eta} = \mathbf{0}$ [6]. In this development, where $\boldsymbol{\Phi}_{\mathbf{z}}$ is row rank deficient two, for each $\boldsymbol{\tau}(\phi)$ there is always a vector that satisfies this first order necessary condition. This vector is associated with a change only in the $\boldsymbol{\tau}$ coordinates. This is expected, since the result in Section 3.2.2 yields all of the solutions for $\boldsymbol{\tau}(\phi)$ at $\mathbf{q}^*$, which are bifurcation points, without tracing continuation curves with $\mathbf{q}^*$ fixed and vary only in the $\boldsymbol{\tau}$ coordinates. A simple proof of the existence and uniqueness of a unit vector satisfying this condition follows.

A vector $\boldsymbol{\eta} = \left[\mathbf{0}, \mathbf{0}, \boldsymbol{\eta}_3^T\right]^T$ satisfies the first order necessary condition for being tangent to the continuation curve at $\mathbf{x} = \left[\mathbf{q}^{*T}, \boldsymbol{\tau}(\phi)^T\right]^T$ if it satisfies $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))\boldsymbol{\eta} = \mathbf{0}$. Using the matrix of Eq. 3.3 and simplifying, this is equivalent to satisfying

$$\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\eta}_3 = \mathbf{0} \tag{3.21a}$$

and

$$2\boldsymbol{\tau}(\phi)^T \boldsymbol{\eta}_3 = 0 \tag{3.21b}$$

The first condition implies that $\boldsymbol{\eta}_3$ is in the null space of $\boldsymbol{\Phi}_{\mathbf{z}}^T$ and the second implies that $\boldsymbol{\eta}_3$ is perpendicular to $\boldsymbol{\tau}(\phi)$. Since the null space of $\boldsymbol{\Phi}_{\mathbf{z}}^T$ is of dimension two and $\boldsymbol{\tau}(\phi) = \cos(\phi)\boldsymbol{\tau}_1 + \sin(\phi)\boldsymbol{\tau}_2$ is, by definition in this null space, the only unit vector choice for $\boldsymbol{\eta}_3$ that satisfies both conditions is

$$\boldsymbol{\eta}_3 = -\sin(\phi)\boldsymbol{\tau}_1 + \cos(\phi)\boldsymbol{\tau}_2 \tag{3.22}$$

This is true, both in the context of this development and that of Ref. 78.

At a bifurcation point, there are at least two independent vectors that satisfy the first order necessary condition. Another vector, $\boldsymbol{\xi} = \left[\boldsymbol{\xi}_1^T, \boldsymbol{\xi}_2^T, \boldsymbol{\xi}_3^T\right]^T$, is sought that has non-zero entries associated with the $\mathbf{q}$ components and that is orthogonal to $\boldsymbol{\eta}$. Thus, $\boldsymbol{\xi}$ is sought such that

$$\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \gamma(\phi))\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}} & \boldsymbol{\Phi}_{\bar{\mathbf{v}}} & \mathbf{0} \\ \begin{bmatrix} \left[\mathbf{L}^T, \mathbf{0}\right] \\ (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi))_{\bar{\mathbf{y}}} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi))_{\bar{\mathbf{v}}} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\Phi}_{\mathbf{z}}^T \end{bmatrix} \\ \mathbf{0} & \mathbf{0} & 2\boldsymbol{\tau}^T(\phi) \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_1 \\ \boldsymbol{\xi}_2 \\ \boldsymbol{\xi}_3 \end{bmatrix} = \mathbf{0} \tag{3.23a}$$

and

$$\boldsymbol{\eta}^T\boldsymbol{\xi} = \left[\mathbf{0}, \mathbf{0}, \boldsymbol{\eta}_3^T\right] \left[\boldsymbol{\xi}_1^T, \boldsymbol{\xi}_2^T, \boldsymbol{\xi}_3^T\right]^T = \boldsymbol{\eta}_3^T\boldsymbol{\xi}_3 = 0 \tag{3.23b}$$

These conditions can be rewritten as

$$\begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix} \boldsymbol{\xi}_1 + \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} \boldsymbol{\xi}_2 = \mathbf{0} \tag{3.24a}$$

$$(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi))_{\bar{\mathbf{y}}}\boldsymbol{\xi}_1 + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi))_{\bar{\mathbf{v}}}\boldsymbol{\xi}_2 + \boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi}_3 = \mathbf{0} \tag{3.24b}$$

$$\boldsymbol{\tau}(\phi)^T\boldsymbol{\xi}_3 = 0 \tag{3.24c}$$

$$\boldsymbol{\eta}_3^T\boldsymbol{\xi}_3 = 0 \tag{3.24d}$$

Equation 3.24a yields

$$\boldsymbol{\xi}_1 = -\begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} \boldsymbol{\xi}_2 \tag{3.25}$$

Using this to eliminate $\boldsymbol{\xi}_1$ from Eq. 3.24b yields

$$\left[-\left(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi)\right)_{\bar{\mathbf{y}}} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} + \left(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi)\right)_{\bar{\mathbf{v}}} \right] \boldsymbol{\xi}_2 = -\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi}_3 \tag{3.26}$$

Let $\hat{\mathbf{E}} \in \Re^{(m-2)\times m}$ be such that $\mathbf{E} \in \Re^{m \times m}$,

$$\mathbf{E} \equiv \begin{bmatrix} \hat{\mathbf{E}} \\ \boldsymbol{\tau}_1^T \\ \boldsymbol{\tau}_2^T \end{bmatrix} \tag{3.27}$$

is an invertible change of variables matrix with $\hat{\mathbf{E}}\boldsymbol{\tau}_1 = \mathbf{0}$ and $\hat{\mathbf{E}}\boldsymbol{\tau}_2 = \mathbf{0}$.

Also let $\hat{\boldsymbol{\beta}} \in \Re^{(m-2)}$, $\beta_1 \in \Re$ and $\beta_2 \in \Re$, be such that $\boldsymbol{\beta} = \left[\hat{\boldsymbol{\beta}}^T, \beta_1, \beta_2\right]^T \in \Re^m$, is the change of variables such that

$$\mathbf{E}^T \boldsymbol{\beta} = \boldsymbol{\xi}_3 \tag{3.28}$$

This change of variables is used to restrict $\boldsymbol{\xi}_3$ so that Eqs. 3.24c and 3.24d are satisfied, by restricting $\boldsymbol{\beta}$ to $\hat{\boldsymbol{\beta}}$. Let $\hat{\boldsymbol{\xi}}_3 \equiv \hat{\mathbf{E}}^T \hat{\boldsymbol{\beta}}$. Since $span\{\boldsymbol{\eta}_3,\ \boldsymbol{\tau}(\phi)\} = span\{\boldsymbol{\tau}_1,\ \boldsymbol{\tau}_2\}$, Eqs. 3.24c and 3.24d are equivalent to

$$\boldsymbol{\xi}_3^T \boldsymbol{\tau}_1 = 0 \tag{3.29a}$$

$$\boldsymbol{\xi}_3^T \boldsymbol{\tau}_2 = 0 \tag{3.29b}$$

Since $\hat{\boldsymbol{\xi}}_3^T \boldsymbol{\tau}_i = \left(\hat{\mathbf{E}}^T \hat{\boldsymbol{\beta}}\right)^T \boldsymbol{\tau}_i = \hat{\boldsymbol{\beta}}^T \hat{\mathbf{E}} \boldsymbol{\tau}_i = \hat{\boldsymbol{\beta}}^T \mathbf{0} = 0,\ i = 1$ and 2, the restriction of $\boldsymbol{\xi}_3$ to $\hat{\boldsymbol{\xi}}_3$ satisfies Eqs. 3.24c and 3.24d.

Using this restriction in Eq. 3.26 reduces the system of Eqs. 3.24 to the single equation

$$\left[-(\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\tau}(\phi))_{\bar{\mathbf{y}}}\begin{bmatrix}\mathbf{\Phi}_{\bar{\mathbf{y}}}\\ \left[\mathbf{L}^T,\mathbf{0}\right]\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{\Phi}_{\bar{\mathbf{v}}}\\ \mathbf{0}\end{bmatrix}+(\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\tau}(\phi))_{\bar{\mathbf{v}}}\right]\boldsymbol{\xi}_2=-\mathbf{\Phi}_{\mathbf{z}}^T\hat{\boldsymbol{\xi}}_3 \tag{3.30}$$

If $\boldsymbol{\tau}(\phi)$ and $\boldsymbol{\xi}_2$ were *a priori* known, then

$$\mathbf{s}=\left[-(\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\tau}(\phi))_{\bar{\mathbf{y}}}\begin{bmatrix}\mathbf{\Phi}_{\bar{\mathbf{y}}}\\ \left[\mathbf{L}^T,\mathbf{0}\right]\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{\Phi}_{\bar{\mathbf{v}}}\\ \mathbf{0}\end{bmatrix}+(\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\tau}(\phi))_{\bar{\mathbf{v}}}\right]\boldsymbol{\xi}_2$$

could be evaluated. When $\mathbf{\Phi}_{\mathbf{z}}^T$, viewed as a linear transformation from $\Re^m$ to $\Re^{n_z}$, is restricted to $\ker(\mathbf{\Phi}_{\mathbf{z}}^T)^{\perp}$, which is what was done by restricting $\boldsymbol{\xi}_3$ to $\hat{\boldsymbol{\xi}}_3$, it induces a one-to-one and onto linear transformation from $\ker(\mathbf{\Phi}_{\mathbf{z}}^T)^{\perp}$ onto $\mathrm{range}(\mathbf{\Phi}_{\mathbf{z}}^T)$ [72]. Thus, for $\mathbf{s}\in\mathrm{range}(\mathbf{\Phi}_{\mathbf{z}}^T)$, there exists a unique $\hat{\boldsymbol{\xi}}_3\in\ker(\mathbf{\Phi}_{\mathbf{z}}^T)^{\perp}$ such that $\mathbf{s}=-\mathbf{\Phi}_{\mathbf{z}}^T\hat{\boldsymbol{\xi}}_3$. Since Eq. 3.30 has n_z equations and $\hat{\boldsymbol{\xi}}_3=\hat{\mathbf{E}}^T\hat{\boldsymbol{\beta}}$ has $n_{\hat{\boldsymbol{\beta}}}=(m-2)$ degrees of freedom, there should be $n_z-(m-2)$ independent equations that must be satisfied. In fact, for arbitrary $\boldsymbol{\tau}(\phi)$ and $\boldsymbol{\xi}_2$, $\mathbf{s}$ is an $n_z\times 1$ vector, which to have $\boldsymbol{\tau}(\phi)$ and $\boldsymbol{\xi}_2$ be parts of the solution to the system of Eq. 3.24, must be part of the $(m-2)$ dimensional space $\mathrm{range}(\mathbf{\Phi}_{\mathbf{z}}^T)$. Thus, there are $n_z-(m-2)$ constraints on $\boldsymbol{\tau}(\phi)$ and $\boldsymbol{\xi}_2$ via Eq. 3.30. Noting that $\{\boldsymbol{\alpha}_i, i=1,...,k\}$ is a basis for the set of all linear functionals $\mathbf{\Psi}$ such that $\mathbf{\Psi}\boldsymbol{\omega}=0$, for all $\boldsymbol{\omega}\in\mathrm{range}(\mathbf{\Phi}_{\mathbf{z}}^T)$, $\mathbf{\Psi}$ can be written as $\mathbf{\Psi}=\mathbf{d}^T\boldsymbol{\alpha}^T$ for some $\mathbf{d}\in\Re^k$. Thus, $\boldsymbol{\alpha}^T$ is the annihilator of $\mathrm{range}(\mathbf{\Phi}_{\mathbf{z}}^T)$ and $\boldsymbol{\alpha}^T\mathbf{\Phi}_{\mathbf{z}}^T=\mathbf{0}\in\Re^{n_z-(m-2)}$ [72]. The vectors $\boldsymbol{\tau}(\phi)$ and $\boldsymbol{\xi}_2$ must satisfy the reduced system,

$$\boldsymbol{\alpha}^T\left[-(\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\tau}(\phi))_{\bar{\mathbf{y}}}\begin{bmatrix}\mathbf{\Phi}_{\bar{\mathbf{y}}}\\ \left[\mathbf{L}^T,\mathbf{0}\right]\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{\Phi}_{\bar{\mathbf{v}}}\\ \mathbf{0}\end{bmatrix}+(\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\tau}(\phi))_{\bar{\mathbf{v}}}\right]\boldsymbol{\xi}_2=\boldsymbol{\alpha}^T\left(-\mathbf{\Phi}_{\mathbf{z}}^T\hat{\boldsymbol{\xi}}_3\right)=\mathbf{0} \tag{3.31}$$

Recalling the definition of Eq. 3.17 and using Eq. 3.20, the previous equation can be written as

$$\mathbf{H}(\boldsymbol{\tau}(\phi))\boldsymbol{\xi}_2 = (\sin(\phi)\mathbf{A} + \cos(\phi)\mathbf{B})\boldsymbol{\xi}_2 = \mathbf{0} \tag{3.32}$$

Noting that if $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}_1)$ has full row rank then, using Theorem 3.2.3, $\mathbf{A}$ is invertible and $cos(\phi) \neq 0$, so Eq. 3.32 can be modified to obtain

$$\frac{1}{cos(\phi)}\mathbf{A}^{-1}(\sin(\phi)\mathbf{A} + \cos(\phi)\mathbf{B})\boldsymbol{\xi}_2 = (\lambda(\phi)\mathbf{I} + \mathbf{A}^{-1}\mathbf{B})\boldsymbol{\xi}_2 = \mathbf{0} \tag{3.33}$$

This shows that finding $\lambda(\phi)$ as an eigenvalue of $-\mathbf{A}^{-1}\mathbf{B}$, as shown in Section 3.2.2, surprisingly yields that the eigenvector associated with $\lambda(\phi)$ will be part of the vector that satisfies the first order necessary condition for being tangent to a continuation curve. The next section details how to construct the rest of this vector and to determine the tangent to the continuation curve by using a second order necessary condition.

3.2.4 Finding All Tangents to Continuation Curves

To find all tangents to continuation curves at bifurcation points, the eigenvalue problem

$$-\mathbf{A}^{-1}\mathbf{B}\boldsymbol{\xi}_2 = \lambda(\phi)\boldsymbol{\xi}_2 \tag{3.34}$$

is used. Multiplying by $\mathbf{A}$ and using Eq. 3.9, Eq. 3.34 becomes

$$(\sin(\phi)\mathbf{A} + \cos(\phi)\mathbf{B})\boldsymbol{\xi}_2 = \mathbf{0}. \tag{3.35}$$

Using Eqs. 3.17 and 3.20, Eq. 3.35 becomes

$$\boldsymbol{\alpha}^T \left[-(\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\tau}(\phi))_{\bar{\mathbf{y}}} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} + (\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\tau}(\phi))_{\bar{\mathbf{v}}} \right] \boldsymbol{\xi}_2 = \mathbf{0} \tag{3.36}$$

As before, defining

$$\mathbf{s} = \left[-(\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi))_{\bar{\mathbf{y}}} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} + (\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi))_{\bar{\mathbf{v}}} \right] \boldsymbol{\xi}_2 \tag{3.37}$$

Eq. 3.36 becomes $\boldsymbol{\alpha}^T \mathbf{s} = \mathbf{0}$. This is equivalent to $\mathbf{s}$ being orthogonal to $\boldsymbol{\alpha}_i^T, i = 1, ..., k$. Since $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k\}$ is an orthonormal basis of $\ker(\mathbf{\Phi}_{\mathbf{z}})$, $\mathbf{s} \in \ker(\mathbf{\Phi}_{\mathbf{z}})^{\perp}$.

Nering [72] has presented a necessary and sufficient condition for the solvability of the linear problem $-\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi}_3 = \mathbf{s}$ namely that $\mathbf{s} \in \ker(\mathbf{\Phi}_{\mathbf{z}})^{\perp}$. By restricting the linear transformation $\mathbf{\Phi}_{\mathbf{z}}^T$ to $\ker(\mathbf{\Phi}_{\mathbf{z}}^T)^{\perp}$, it can be regarded as a one-to-one and onto linear transformation from $\ker(\mathbf{\Phi}_{\mathbf{z}}^T)^{\perp}$ onto $\mathrm{range}(\mathbf{\Phi}_{\mathbf{z}}^T) = \ker(\mathbf{\Phi}_{\mathbf{z}})^{\perp}$ [72]. With this restriction, there exists a unique vector $\boldsymbol{\xi}_3$ such that

$$\boldsymbol{\xi}_3 \in \ker(\mathbf{\Phi}_{\mathbf{z}}^T)^{\perp} \tag{3.38}$$

and

$$\mathbf{s} = -\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi}_3 \tag{3.39}$$

This also shows that

$$\boldsymbol{\tau}(\phi)^T \boldsymbol{\xi}_3 = 0 \tag{3.40}$$

since $\boldsymbol{\tau}(\phi)$ is in $\ker(\mathbf{\Phi}_{\mathbf{z}}^T)$, whereas $\boldsymbol{\xi}_3$ is in $\ker(\mathbf{\Phi}_{\mathbf{z}}^T)^{\perp}$.

From Eqs. 3.37 and 3.39,

$$\left[-(\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi))_{\bar{\mathbf{y}}} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} + (\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi))_{\bar{\mathbf{v}}} \right] \boldsymbol{\xi}_2 = -\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi}_3 \tag{3.41}$$

Define

$$\boldsymbol{\xi}_1 = - \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} \boldsymbol{\xi}_2 \tag{3.42}$$

From Eqs. 3.42, 3.41, and 3.40, the resulting system is formed

$$\begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix} \boldsymbol{\xi}_1 + \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} \boldsymbol{\xi}_2 = \mathbf{0} \tag{3.43a}$$

$$(\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi))_{\bar{\mathbf{y}}} \boldsymbol{\xi}_1 + (\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi))_{\bar{\mathbf{v}}} \boldsymbol{\xi}_2 + \mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi}_3 = \mathbf{0} \tag{3.43b}$$

$$\boldsymbol{\tau}(\phi)^T \boldsymbol{\xi}_3 = 0 \tag{3.43c}$$

The above equations can be written in matrix form as

$$\begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{y}}} & \mathbf{\Phi}_{\bar{\mathbf{v}}} & \mathbf{0} \\ \begin{bmatrix} \left[\mathbf{L}^T, \mathbf{0}\right] \\ (\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi))_{\bar{\mathbf{y}}} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ (\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi))_{\bar{\mathbf{v}}} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ \mathbf{\Phi}_{\mathbf{z}}^T \end{bmatrix} \\ \mathbf{0} & \mathbf{0} & 2\boldsymbol{\tau}^T(\phi) \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_1 \\ \boldsymbol{\xi}_2 \\ \boldsymbol{\xi}_3 \end{bmatrix} = \mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))\boldsymbol{\xi} = \mathbf{0} \tag{3.44}$$

It is clear that $\boldsymbol{\xi} = \left[\boldsymbol{\xi}_1^T, \boldsymbol{\xi}_2^T, \boldsymbol{\xi}_3^T\right]^T$ satisfies the first order necessary condition for a tangent to the continuation curve at $\mathbf{q}^*$, associated with $\boldsymbol{\tau}(\phi)$.

The previous analysis proves the following result.

Theorem 3.2.4 *Let $\boldsymbol{\xi}_2$ be an eigenvector associated with the eigenvalue $\lambda(\phi)$ of the matrix $-\mathbf{A}^{-1}\mathbf{B}$. Evaluate ϕ from $\tan(\phi) = \lambda(\phi)$, as in Problem 3.2.1, and let $\boldsymbol{\tau}(\phi)$ be as in Eq. 3.5. Let $\boldsymbol{\xi}_1 = -\begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} \boldsymbol{\xi}_2$ and let $\boldsymbol{\xi}_3$ be the unique solution of*

$$\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi}_3 = \left[\left(\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi)\right)_{\bar{\mathbf{y}}} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} - \left(\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\phi)\right)_{\bar{\mathbf{v}}} \right] \boldsymbol{\xi}_2 \tag{3.45a}$$

$$\boldsymbol{\tau}_1^T \boldsymbol{\xi}_3 = 0 \tag{3.45b}$$

$$\boldsymbol{\tau}_2^T \boldsymbol{\xi}_3 = 0 \tag{3.45c}$$

Then, $\boldsymbol{\xi} = \left[\boldsymbol{\xi}_1^T, \boldsymbol{\xi}_2^T, \boldsymbol{\xi}_3^T\right]^T$ satisfies the first order necessary condition for a tangent to the continuation curve at $\mathbf{q}^*$, associated with $\boldsymbol{\tau}(\phi)$.

Since $\boldsymbol{\xi}_2$ is an eigenvector, $\boldsymbol{\xi}_2 \neq \mathbf{0}$, and $\boldsymbol{\xi} \neq \mathbf{0}$. Normalize $\boldsymbol{\xi}$ and denote the unit vector obtained also by $\boldsymbol{\xi}$.

Recall that $\boldsymbol{\eta} \equiv \left[\mathbf{0}, \mathbf{0}, \boldsymbol{\eta}_3^T\right]^T$, where $\boldsymbol{\eta}_3 = -\sin(\phi)\boldsymbol{\tau}_1 + \cos(\phi)\boldsymbol{\tau}_2$ also satisfies the first order necessary condition. Equations 3.38 and 3.21a show that $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ are linearly independent. Since the first order necessary condition is linear, any linear combination of $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ will satisfy it. A second order necessary condition is needed to determine the vector $\boldsymbol{\zeta} \in \ker(\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}(\phi)))$ that is tangent to a continuation curve at $(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$. This condition is

$$\boldsymbol{\zeta}^T(\mathbf{e}_i^T\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}(\phi)))_{\mathbf{x}}\boldsymbol{\zeta} = 0, i = 1, 2, ..., p \tag{3.46}$$

where $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_p\}$ spans $\ker(\mathbf{G}_{\mathbf{x}}^T(\mathbf{q}^*, \boldsymbol{\tau}(\phi)))$, and $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ has row rank deficiency $p \geq 1$ [6].

Note that a particular eigenvalue $\lambda(\phi)$ may have algebraic multiplicity greater than one. Since the null space of $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ can be computed once $\boldsymbol{\tau}(\phi)$, hence $\lambda(\phi)$, is known, it is assumed that if p is the algebraic multiplicity of $\lambda(\phi)$, then either p independent vectors satisfying the first order necessary condition at $\left[\mathbf{q}^T, \boldsymbol{\tau}(\phi)^T\right]^T$ are found, or p is also the geometric multiplicity and the above process is used to find p linearly independent eigenvectors $\{\boldsymbol{\xi}_2^{(i)}, i = 1, ..., p\}$, yielding p vectors $\boldsymbol{\xi}^{(i)}$. In practice, the null space of $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ does not need to be computed, since no cases in which the algebraic multiplicity was not equal to the geometric multiplicity have been encountered to date in mechanism and workspace analysis.

Thus, any tangent to a continuation curve at $\mathbf{q}^*$, associated with $\boldsymbol{\tau}(\phi)$, is of the form

$$\boldsymbol{\zeta} = a\boldsymbol{\eta} + \sum_{i=1}^{p} b_i \boldsymbol{\xi}^{(i)} = \begin{bmatrix} \sum_{i=1}^{p} b_i \boldsymbol{\xi}_1{}^{(i)} \\ \sum_{i=1}^{p} b_i \boldsymbol{\xi}_2{}^{(i)} \\ a\boldsymbol{\eta}_3 + \sum_{i=1}^{p} b_i \boldsymbol{\xi}_3{}^{(i)} \end{bmatrix} \tag{3.47}$$

where a and $b_i \in \Re$. Form the system of p equations from the second order necessary condition of Eq. 3.46 and add the normalization constraint $||\boldsymbol{\zeta}||_2^2 = a^2 + \sum_{i=1}^{p} b_i^2 = 1$. This results in $p+1$ quadratic equations in the $p+1$ unknowns $\{a, b_i\}, i = 1, ..., p$. HOMPACK [69, 71, 99] solves such systems using polynomial continuation [21]. Of course, if $p = 1$ the quadratic formula provides an easy solution [40].

Using analytical differentiation of $\mathbf{G}_\mathbf{x}(\mathbf{q}^*, \boldsymbol{\tau})\boldsymbol{\eta}$, taking into consideration the special form of $\boldsymbol{\eta}$, shows that $a = 1$ and $b_i = 0, i = 1, ..., p$, is a solution of this system. This tangent is of no interest, since it corresponds to a continuation curve in which the coordinates are only changing in the $\boldsymbol{\tau}$ components. The solutions sought are the physical space tangents, for which $b_i \neq 0$, for some i.

It is hypothesized that in case $\mathbf{G}_\mathbf{x}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ is row rank deficient one, the second of the two solutions is $a = 0$ and $b \equiv b_1 = 1$, but it is not possible to verify this, in general, by performing analytical differentiation to obtain the higher order derivatives required in the second order necessary condition. In this case, however, since $p = 1$, the first $n_{\bar{y}} + n_{\bar{v}}$ tangent components are uniquely determined by $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ and, in practice, setting $\boldsymbol{\zeta} = \boldsymbol{\xi}$ as the tangent is sufficient for numerical methods used in continuation to converge to a physical space continuation curve. In both of the examples presented in Section 3.4 it is the case that the second of the two solutions is $a = 0$ and $b_1 = 1$.

If $p > 1$, it is necessary to form the system for the second order necessary conditions. This requires finding the vectors $\mathbf{e}_i, i = 1, ..., p$. In the next section it is shown that these vectors can be found from an eigenvalue problem.

3.2.5 Calculation of Tangents when $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau})$ is Row Rank Deficient Greater Than Two

If $\text{rank}\mathbf{G}_{\mathbf{x}}(\mathbf{x}_0) = n_x - 1 - p \equiv k < n_x - 2$, the calculation of tangent vectors as presented in Section 1.3.4 needs to be slightly modified. The assumption in Section 1.3.4 is that $\text{rank}\mathbf{G}_{\mathbf{x}}(\mathbf{x}_0) = n_x - 2$, which is the case $p = 1$ from the previous section. As in Section 1.3.4, the QR factorization of $\mathbf{G}_{\mathbf{x}}(\mathbf{x}_0) = \mathbf{QR}$ can be obtained, where $\mathbf{Q}$ is an orthogonal matrix, i.e., $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ of dimension $(n_x - 1) \times (n_x - 1)$, and $\mathbf{R}$ is upper triangular of dimension $(n_x - 1) \times n_x$. $\mathbf{R}$ has the last p rows zero due to the rank deficiency of $\mathbf{G}_{\mathbf{x}}(\mathbf{x}_0)$. The system of bifurcation equations, determining the source of singular behavior is formed by a modifying Eq. 1.12 to include the last $n_x - k - 1$ columns of $\mathbf{Q}$

$$f^j(\mathbf{x}) \equiv \mathbf{q}_{n_x - j}^T \mathbf{G} = 0,\ j = 1,\ 2,\ \ldots,\ p \tag{3.48}$$

Now the first k rows of $\mathbf{R}$ is $\mathbf{F}_{\mathbf{x}}(\mathbf{x}_0)$, a $k \times n_x$ matrix with full row rank, so there are $n_x - k$ vectors $\boldsymbol{\xi}^i$ satisfying Eq. 1.14. Equation 1.18 becomes a system of second order equations in $\boldsymbol{\alpha}$.

The vectors $\mathbf{e}_i, i = 1, ..., p$, from the second order necessary condition of the previous section are in $\ker(\mathbf{G}_{\mathbf{x}}^T(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ and could be chosen as the last $n_x - k - 1$ columns of $\mathbf{Q}$. In the construction of Section 3.2.2, a vector, $\mathbf{c} = \left[\mathbf{c}_1^T, c_{21}, \mathbf{c}_{22}^T, c_3\right]^T$ is used to characterize the condition that $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ is row rank deficient. Note that $\mathbf{c} \in \ker(\mathbf{G}_{\mathbf{x}}^T(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$, by construction, and it will be shown that the collection of vectors $\mathbf{c}$ associated with $\lambda(\phi)$ can be used to characterize row rank deficiency, in place of the vectors $\mathbf{e}_i$. This $\mathbf{c}$ can be found in multiplicity equal to the algebraic multiplicity of $\lambda(\phi)$, which is equal to the row rank deficiency of $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$.

Using Eq. 3.20 to simplify Eq. 3.16 yields

$$\mathbf{d}^T\mathbf{H}(\boldsymbol{\tau}(\phi)) = \mathbf{d}^T\left(\sin(\phi)\mathbf{A} + \cos(\phi)\mathbf{B}\right) = \mathbf{0} \tag{3.49}$$

Dividing by $\cos(\phi)$ and using $\lambda(\phi) \equiv \tan(\phi)$ transforms this into the following generalized eigenvalue problem with left eigenvector $\mathbf{d}$:

$$\mathbf{d}^T\left(\lambda(\phi)\mathbf{A} + \mathbf{B}\right) = \mathbf{0} \tag{3.50}$$

Multiplying on the right by $\mathbf{A}^{-1}$ and taking the transpose of the equation yields the following standard eigenvalue problem

$$\left(\lambda(\phi)\mathbf{I} + \mathbf{A}^{-1^T}\mathbf{B}^T\right)\mathbf{d} = \mathbf{0} \tag{3.51}$$

Since the eigenvalues of a matrix are the same as for its transpose, the solutions for $\lambda(\phi)$ in the above problem are the same as those of Eq. 3.33. Denote the eigenvectors associated with $\lambda(\phi)$ as $\mathbf{d}^{(i)}, i = 1, ..., p$, where p is the algebraic multiplicity of $\lambda(\phi)$. Recover $\mathbf{c}_{22}^{(i)} = \boldsymbol{\alpha}\mathbf{d}^{(i)}$, and then $\mathbf{c}_1^{(i)}$ and $c_{21}^{(i)}$, as

$$\left[\mathbf{c}^{(i)T}_{\;1}, c^{(i)}_{\;21}\right] = -\mathbf{c}^{(i)T}_{\;22}(\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\tau}(\phi))_{\bar{\mathbf{y}}}\begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \tag{3.52}$$

From the construction of Section 3.2.2, $c_3^{(i)} = 0$, which completes the calculation of $\mathbf{c}^{(i)} \equiv \mathbf{e}_i$.

Using these vectors, the system from the second order necessary condition can be formed. HOMPACK [99] can be used to solve the second order system of polynomial equations, and tangents to the continuation curves can be found from Eq. 3.47.

3.2.6 Efficient Update of $\mathbf{G}_{\mathbf{x}}\left(\mathbf{q}^*, \boldsymbol{\tau}\left(\phi\right)\right)$ and $\left(\mathbf{e}_i^T\mathbf{G}_{\mathbf{x}}\left(\mathbf{q}^*, \boldsymbol{\tau}\left(\phi\right)\right)\right)_{\mathbf{x}}$

It is often necessary to update $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ for different values of ϕ. This is done when searching for some $\boldsymbol{\tau}(\phi)$ such that $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ is full row rank, when forming the matrix components of $\mathbf{A}$ and $\mathbf{B}$, or when verifying that a solution $\boldsymbol{\tau}(\phi)$ is such that $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ is row rank deficient.

If $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ are an orthonormal basis for the null space of $\boldsymbol{\Phi}_{\mathbf{z}}(\mathbf{q})$ and if $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}_1)$ and $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}_1)$ are known then $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, a\boldsymbol{\tau}_1 + b\boldsymbol{\tau}_2)$ can be evaluated for any $\boldsymbol{\tau} = a\boldsymbol{\tau}_1 + b\boldsymbol{\tau}2$, without additional derivative evaluations. This is due to the fact that portions of $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau})$ that depend on $\boldsymbol{\tau}$ do so linearly. Thus, for $\boldsymbol{\tau} = a\boldsymbol{\tau}_1 + b\boldsymbol{\tau}2$, compute

$$
\begin{aligned}
&\mathbf{G}_{\mathbf{x}}(\mathbf{q}, a\boldsymbol{\tau}_1 + b\boldsymbol{\tau}_2) \\
&= \begin{bmatrix}
\boldsymbol{\Phi}_{\bar{\mathbf{y}}}(\mathbf{q}) & \boldsymbol{\Phi}_{\bar{\mathbf{v}}}(\mathbf{q}) & \mathbf{0} \\
\begin{bmatrix} \left[\mathbf{L}^T, \mathbf{0}\right] \\ (\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q})(a\boldsymbol{\tau}_1 + b\boldsymbol{\tau}_2))_{\bar{\mathbf{y}}} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ (\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q})(a\boldsymbol{\tau}_1 + b\boldsymbol{\tau}_2))_{\bar{\mathbf{v}}} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q}) \end{bmatrix} \\
\mathbf{0} & \mathbf{0} & 2(a\boldsymbol{\tau}_1 + b\boldsymbol{\tau}_2)^T
\end{bmatrix} \\
&= \begin{bmatrix}
\boldsymbol{\Phi}_{\bar{\mathbf{y}}}(\mathbf{q}) & \boldsymbol{\Phi}_{\bar{\mathbf{v}}}(\mathbf{q}) & \mathbf{0} \\
\begin{bmatrix} \left[\mathbf{L}^T, \mathbf{0}\right] \\ a(\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q})\boldsymbol{\tau}_1)_{\bar{\mathbf{y}}} + b(\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q})\boldsymbol{\tau}_2)_{\bar{\mathbf{y}}} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ a(\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q})\boldsymbol{\tau}_1)_{\bar{\mathbf{v}}} + b(\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q})\boldsymbol{\tau}_2)_{\bar{\mathbf{v}}} \end{bmatrix} & \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q}) \end{bmatrix} \\
\mathbf{0} & \mathbf{0} & 2a{\boldsymbol{\tau}_1}^T + 2b{\boldsymbol{\tau}_2}^T
\end{bmatrix}
\end{aligned}
\tag{3.53}
$$

By updating only the subcomponents in Eq. 3.53 that depend on $\boldsymbol{\tau}$, the Jacobian of $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau})$ can be efficiently evaluated.

This observation also holds true for second derivatives. To formulate the second order necessary condition for tangents, it is necessary to evaluate $\mathbf{c}^T\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi)))_{\mathbf{x}}$. If $\mathbf{c}$ has a one in the $i-th$ component and zeroes elsewhere, then $\mathbf{c}^T\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\phi)))_{\mathbf{x}}$ is the Hessian matrix of the $i-th$ equation of $\mathbf{G}(\mathbf{x})$. Thus, if $\mathbf{c}$ is such that all non-zero terms correspond to equations from either the $\boldsymbol{\Phi}(\mathbf{q})$ or $\mathbf{L}^T\mathbf{u} - b$ of $\mathbf{G}(\mathbf{x})$, then $\left(\mathbf{c}^T \mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))\right)_{\mathbf{x}}$ is independent of of $\boldsymbol{\tau}(\phi)$.

If $\mathbf{c}$ has non-zero terms corresponding to $\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}$, then $\left(\mathbf{c}^T \mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))\right)_{\mathbf{x}}$ is again linear in $\boldsymbol{\tau}(\phi)$ and can be efficiently updated.

3.3 Numerical Algorithm

Algorithm 3.3.1 *At a bifurcation point* $\mathbf{q}^*$*, with an orthonormal basis* $\{\boldsymbol{\tau}_1, \boldsymbol{\tau}_2\}$ *of the null space of* $\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q}^*)$*,*

(1) If $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_1)$ *is row rank deficient, choose a* ϕ_0 *and set*

$$\boldsymbol{\tau}_1 = \cos(\phi_0)\boldsymbol{\tau}_1 + \sin(\phi_0)\boldsymbol{\tau}_2 \tag{3.54a}$$

$$\boldsymbol{\tau}_2 = -\sin(\phi_0)\boldsymbol{\tau}_1 + \cos(\phi_0)\boldsymbol{\tau}_2 \tag{3.54b}$$

such that $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_1)$ *is not row rank deficient. Note that, since the probability that both* $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_1)$ *and* $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_2)$ *are row rank deficient is essentially zero, if* $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_2)$ *has full row rank, choosing* $\phi_0 = \pi/2$ *will almost always work. This choice saves recomputation of the derivatives* $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_1)$ *and* $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_2)$*. If both* $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_1)$ *and* $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}_2)$ *are row rank deficient, then choose another* ϕ_0 *and use the Jacobian update method of Section 3.2.6.*

(2) Solve for an orthonormal basis $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k\}$ *of the null space of* $\boldsymbol{\Phi}_{\mathbf{z}}(\mathbf{q}^*)$*; i.e., find all* $\boldsymbol{\alpha}_i$ *such that such that* $\boldsymbol{\Phi}_{\mathbf{z}}(\mathbf{q}^*)\boldsymbol{\alpha}_i = \mathbf{0}$ *and form the matrix* $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k]$*.*

(3) Evaluate

$$\mathbf{A} = \boldsymbol{\alpha}^T \left[-(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}_1)_{\bar{\mathbf{y}}} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}_1)_{\bar{\mathbf{v}}} \right] \tag{3.55}$$

$$\mathbf{B} = \boldsymbol{\alpha}^T \left[-(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}_2)_{\bar{\mathbf{y}}} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}_2)_{\bar{\mathbf{v}}} \right] \tag{3.56}$$

(4) Evaluate $\mathbf{A}^{-1}$.

(5) Evaluate $-\mathbf{A}^{-1}\mathbf{B}$.

(6) For the matrix $-\mathbf{A}^{-1}\mathbf{B}$*, find all eigenvalues* λ^j *and associated eigenvectors,* $\boldsymbol{\xi}_2^{j\,(i)}, i = 1, ..., p(j)$*, where* $p(j)$ *is the algebraic multiplicity of* λ^j.

(7) For each eigenvalue, compute $\boldsymbol{\tau}(\lambda^j)$.

(8) For each j *and* $i = 1, ..., p(j)$*, compute*

$$\boldsymbol{\xi}_1^{j\,(i)} = -\begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{v}}} \\ \mathbf{0} \end{bmatrix} \boldsymbol{\xi}_2^{j\,(i)}$$

$\boldsymbol{\xi}_3^{j\,(i)}$ *from Eq. 3.45a.*

(9) Normalize $\boldsymbol{\xi}^{j\,(i)} = \left[\boldsymbol{\xi}_1^{j\,(i)T}, \boldsymbol{\xi}_2^{j\,(i)T}, \boldsymbol{\xi}_3^{j\,(i)T}\right]^T$ *and denote the normalized vector* $\boldsymbol{\xi}^{j\,(i)}$.

(10) For each j*, compute* $\boldsymbol{\eta}_3^j$ *via Eq. 3.22.*

(11) For the matrix $-\mathbf{A}^{-1^T}\mathbf{B}^T$*, find all eigenvalues, ordered the same as* λ^j*, and associated eigenvectors,* $\mathbf{d}^{j\,(i)}, i = 1, ..., p(j)$.

(12) For each j *and* $i = 1, ..., p(j)$*, compute* $\mathbf{c}^{j\,(i)} = \left[\mathbf{c}_1^{j\,(i)^T}, c_{21}^{j\,(i)}, \mathbf{c}_{22}^{j\,(i)^T}, 0\right]^T$ *from*

$$\mathbf{c}_{22}^{j\,(i)} = \boldsymbol{\alpha}\mathbf{d}^{j\,(i)}$$

$$\left[\mathbf{c}_1^{j\,(i)^T}, c_{21}^{j\,(i)}\right] = -\mathbf{c}_{22}^{j\,(i)^T} (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau}(\lambda^j))_{\bar{\mathbf{y}}} \begin{bmatrix} \boldsymbol{\Phi}_{\bar{\mathbf{y}}} \\ \left[\mathbf{L}^T, \mathbf{0}\right] \end{bmatrix}^{-1}$$

(13) For each j, solve the $p(j)+1$ order quadratic system using HOMPACK

$$||\boldsymbol{\zeta}||_2^2 = a^2 + \sum_{i=1}^{p} b_i^2 = 1$$

$$\boldsymbol{\zeta}^T(\mathbf{c}^{j\,(i)}\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\tau}(\lambda^j)))_{\mathbf{x}}\boldsymbol{\zeta} = 0 \; \textit{for } i = 1, 2, ..., p$$

for a and $b_i, i = 1, ..., p(j)$, where

$$\boldsymbol{\zeta} = a\boldsymbol{\eta}^j + \sum_{i=1}^{p} b_i \boldsymbol{\xi}^{j\,(i)} = \begin{bmatrix} \sum_{i=1}^{p} b_i \boldsymbol{\xi}_1^{j\,(i)} \\ \sum_{i=1}^{p} b_i \boldsymbol{\xi}_2^{j\,(i)} \\ a\boldsymbol{\eta}_3^j + \sum_{i=1}^{p} b_i \boldsymbol{\xi}_3^{j\,(i)} \end{bmatrix}$$

(14) For each j, exclude the solution for a and $b_i, i = 1, ..., p(j)$, where $a = 1$ and $b_i = 0, i = 1, ..., p(j)$. The remaining solutions for a and $b_i, i = 1, ..., p(j)$, yield tangents to continuation curves at $\mathbf{q}^*$, associated with $\boldsymbol{\tau}(\lambda^j)$, as

$$\boldsymbol{\zeta}^j = a\boldsymbol{\eta}^j + \sum_{i=1}^{p} b_i \boldsymbol{\xi}^{j\,(i)}$$

3.4 Numerical Examples

Two examples are presented here. The first is a performance comparison using the spatial Stewart platform accessible output set as developed in Section 1.4.2. The second is an illustration of the need to solve the second order system of equations for a spatial serial manipulator that has bifurcation points with $p = 2$ from Eq. 3.46.

3.4.1 Spatial Stewart Platform

Recall the development of the spatial Stewart platform constraint equations from Section 1.4.2. In order to compute the boundary of the accessible output set, as described in Section 1.4, the condition for row rank deficiency of the sub-Jacobian matrix $\boldsymbol{\Phi}_{\mathbf{z}}$ is augmented with the constraint equations $\boldsymbol{\Phi}(\mathbf{q}) = \mathbf{0}$. Recall that the necessary condition for points on the boundary of the accessible output set is given by Eq. 1.4.

Table 3.1: Performance in CPU Seconds of Different Models

	Model 1	Model 2
New method	System 0.09 User 0.98 Total 1.07	System 0.09 User 0.84 Total 0.93
Method of Ref. 78	System 0.76 User 8.88 Total 9.64	Method does not apply

Since ∂A is a surface in $\Re^3$, a family of horizontal cutting planes is introduced, as $z-b=0$. As in Chapter 2, denote by Model 1 the mathematical model that results from augmenting the equation $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$ with the cutting plane equation and then applying the necessary condition for the boundary. Model 2 results from applying the necessary condition for the boundary to $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$ and then appending the cutting plane equation, $z - b = 0$.

For the value $b = 3.42$ meters, the boundary of the accessible output set is computed using both mathematical models. Fig. 3.1 gives the singular curves for this cutting plane. The numerical continuation methods presented in Ref. 40 were used to trace one dimensional curves. Branches at bifurcation points were computed using both the method of Ref. 78 and the method presented in this chapter. The solution associated with Model 1 can be computed using both bifurcation point methods. However, for Model 2 only the method presented in this paper applies.

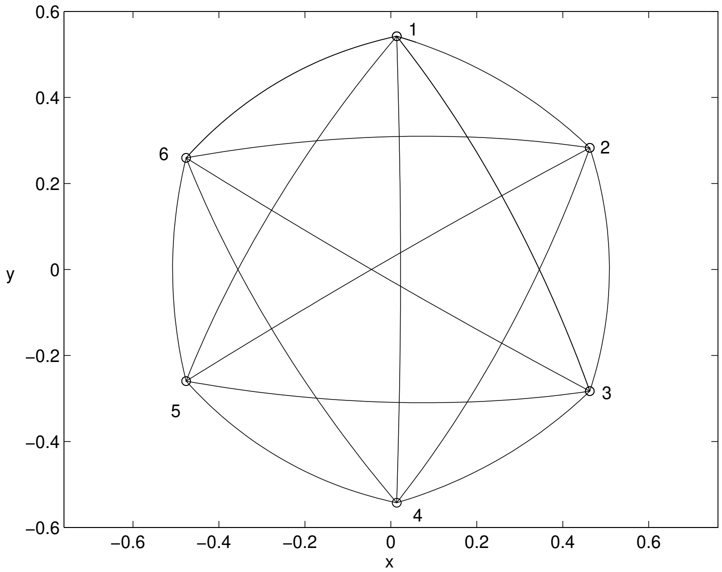

Figure 3.1: Singular Curves at z = 3.42 Meters

Table 3.1 contains a summary of the computational times for the various methods, when run on a Hewlett Packard 9000 series 755 workstation. The computation times shown are the time required for computation of all branches at bifurcation point number four in Fig. 3.1. This table shows that the most efficient computational method is to use the bifurcation method of this paper with Model 2. For this example, the new method enables branch computation at bifurcation points more than ten times faster than previously possible.

Note that the times given for Model 1 with the method of Ref. 78 differ slightly from those presented there. This is because the timing there was performed without computing the solution of $\boldsymbol{\xi}_3$ at each branch. For this example, it happens that $\boldsymbol{\xi}_3 = \mathbf{0}$ at each branch; however, this should be computationally verified in general. The timing presented here reflects the verification that $\boldsymbol{\xi}_3 = \mathbf{0}$ at each branch.

3.4.2 Spatial Serial Manipulator

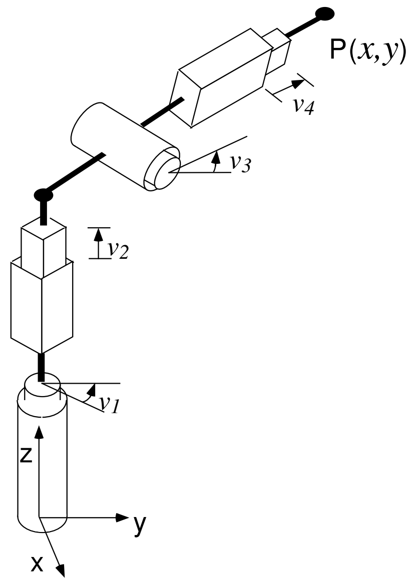

Figure 3.2: Four Degree of Freedom Spatial Serial Manipulator

The four degree of freedom spatial serial manipulator shown in Fig. 3.2 can be modelled using the Denavit-Hartenberg representation method [14, 84] with the following constraint equation [106]:

$$\mathbf{\Phi}\left(\mathbf{q}\right) \equiv \begin{bmatrix} x - (\cos(v_1)\cos(v_3)v_4 + 30\cos(v_1)) \\ y - (\sin(v_1)\cos(v_3)v_4 + 30\sin(v_1)) \\ z - (\sin(v_3)v_4 + v_2) \end{bmatrix} \tag{3.57}$$

with generalized coordinate $\mathbf{q} = [x, y, z, v_1, v_2, v_3, v_4]^T$.

As presented in Yeh and Abdel-Malek [106], unilateral constraints $20 \leq v_2 \leq 50$, $-\pi/4 \leq v_3 \leq \pi$, and $10 \leq v_4 \leq 20$ are imposed on the actuator inputs. Actuator input v_1 is unrestricted. These unilateral constraints can be enforced by defining modified input functions,

$$v_2 = 35 + 15\sin(v_2^*)$$

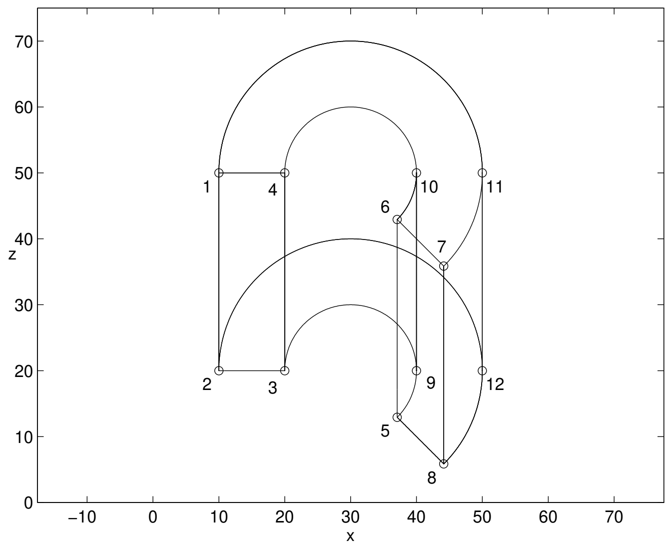

Figure 3.3: Singular Curves for $y = 0$

$$v_3 = \frac{3\pi}{8} + \frac{5\pi}{8}\sin(v_3^*)$$
$$v_4 = 15 + 5\sin(v_4^*)$$

and replacing generalized coordinate v_i with v_i^* for $i = 2, 3$, and 4.

The accessible output set boundary was computed for cutting plane $y = 0$ and is shown in Fig. 3.3. A cutaway view of the three dimensional accessible output set, showing boundary and interior singular surfaces is shown in Fig. 3.4 as presented in Ref. 106.

For the bifurcation point number 4 in Fig. 3.3, the generalized coordinate is $[20.0, 0.0, 50.0, 0.0, \pi/2, \pi/2, -\pi/2]^T$. The eigenvalues of $\mathbf{A}^{-1}\mathbf{B}$ from Eq. 3.34 are 0.194380, 0.194380, and -5.144555. The value of $\boldsymbol{\tau}$ associated with the repeated eigenvalue is $[-1.0, 0.0, 0.0]^T$ and the other is $[0.0, 0.0, 1.0]^T$. The two tangents associated with the repeated eigenvalue, determined using the procedure of Section 3.2.4 and satisfying the first order necessary condition of Eq. 3.44, are unit vectors with

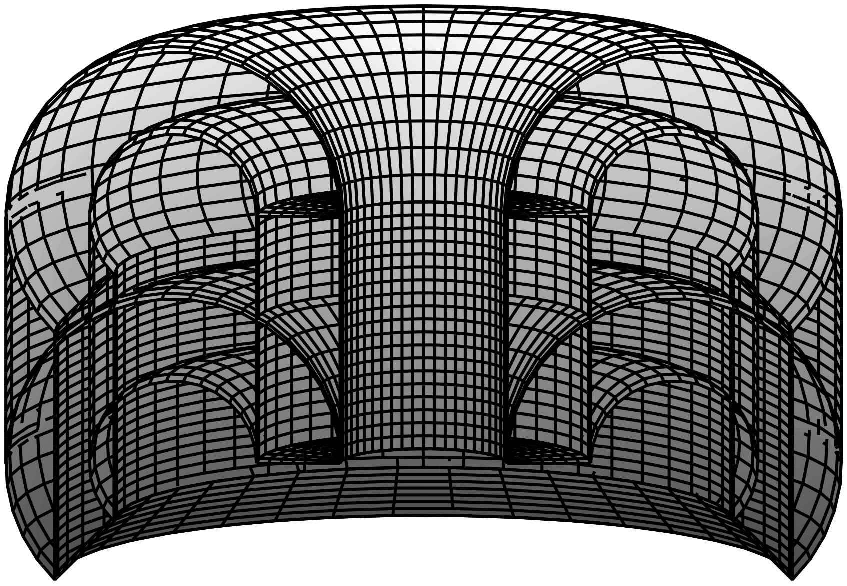

Figure 3.4: Cutaway View of Singular Surfaces

a one in the fifth or sixth entry and zeroes elsewhere. The third tangent associated with change in the $\boldsymbol{\tau}$ components is a unit vector with a one in the tenth entry. The matrix $\mathbf{G}_{\mathbf{x}}^T$ is row rank deficient two, and the vectors found in the null space are unit vectors with a one in the sixth or seventh entry. The resulting second order system, as in Step (14) of the numerical algorithm, is formed from these vectors and the second derivatives of $\mathbf{G}(\mathbf{x})$

$$\begin{bmatrix} -0.0004b_1b_2 + 0.0004b_1^2 + 0.0004 * a^2 - 30.0ab_1 \\ 0.0005b_2^2 - 0.0004b_1b_2 + 39.27ab_2 - 0.0005a^2 \\ a^2 + b_1^2 + b_2^2 - 1.0 \end{bmatrix} = \mathbf{0}$$

The solutions are

$$[a, b_1, b_2] = [1.0, 0.000013, 0.000013]\,, \; [0.0, 1.0, 0.0]\,, \;\; \text{and} \;\; [0.00351, 0.958, -0.286]$$

as given by HOMPACK. Note that the tangent resulting from the first solution is not of interest, as it has components that vary only in the $\boldsymbol{\tau}$ coordinates.

The non-repeated eigenvalue corresponds to a simple bifurcation point, with physical space tangent as a unit vector with a one in the seventh entry and the matrix $\mathbf{G_x}$ is row rank deficient one.

Bifurcation points numbered 1–4 in Fig. 3.3 are of the type detailed above. These all have a $\boldsymbol{\tau}(\phi)$ value such that $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ is row rank deficient two, resulting in two physical space tangents. For the same generalized coordinate $\mathbf{q}$, there is another choice of $\boldsymbol{\tau}(\phi)$ such that $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ is row rank deficient one, resulting in one additional physical space tangent. This illustrates the ability of the methods developed in this chapter to handle this new class of multiple bifurcation points.

Bifurcation points numbered 5–8 are of the type for which $\boldsymbol{\Phi}_\mathbf{z}$ is row rank deficient two, but for the resulting $\boldsymbol{\tau}(\phi)$, $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\tau}(\phi))$ is row rank deficient one. Bifurcation points numbered 9–12 are simple bifurcation points.

CHAPTER 4
SPECIAL PURPOSE NUMERICAL CONTINUATION METHOD

Numerical methods for mapping boundary surfaces associated with various properties of mechanisms and manipulators have been presented in Refs. 40, 41, 49, and 103. In all of these presentations the underlying continuation method was one of general purpose for solving for the zero level set solution of $\mathbf{G} : \Re^{n_x} \mapsto \Re^{n_x - 1}$,

$$\mathbf{G}(\mathbf{x}) = \mathbf{0} \tag{4.1}$$

In this chapter a dense output algorithm that is applicable for general numerical continuation methods is presented and the special structure of the implicit boundary surface equations

$$\mathbf{G}(\mathbf{q}, \gamma) = \begin{bmatrix} \begin{bmatrix} \mathbf{\Phi} \\ \mathbf{L}^T \mathbf{u} + d \end{bmatrix} \\ \Phi_{\mathbf{z}}^T \gamma \\ \gamma^T \gamma - 1 \end{bmatrix} = \mathbf{0} \tag{4.2}$$

is exploited to provide better convergence and stepsize control for this class of problems. As an additional feature, this optimized continuation method can provide additional information about the mechanism being analyzed.

4.1 Background on Numerical Continuation

This section presents some general background on numerical continuation followed by specific information on the numerical continuation package PITCON.

4.1.1 General Background

Methods for solving the the general problem $\mathbf{G}(\mathbf{x}) = \mathbf{0}$ for $\mathbf{G} : \Re^{n_x} \mapsto \Re^{n_x - 1}$ could solve the related ODE initial value problem, obtained by differentiating Eq. 4.1,

$$\mathbf{G_x}(\mathbf{x})\dot{\mathbf{x}} = \mathbf{0}; \qquad \mathbf{x}(\mathbf{0}) = \mathbf{x_0}$$

However, such solution methods ignore the attractive contractive properties that the solution curve has relative to Newton-type iterative methods, since it is the set of zero points of $\mathbf{G}(\mathbf{x})$ [6]. Thus, general continuation methods for solving $\mathbf{G}(\mathbf{x}) = \mathbf{0}$ rely on predicting a new solution point, as is done for ODE methods, and then correcting the predicted value with Newton type iteration, as is done for solving nonlinear systems of equations.

For the case of manifold boundary calculation using Eq. 4.2, it is sought to further improve the prediction and enhance the contractive properties of the continuation method. This can be done by improving the correction direction, thus taking fewer correction steps, or by modifying the correction procedure to take advantage of the form of Eq. 4.2. Methods for such enhancements are presented in Sections 4.3.1 and 4.3.2. When the goal is computational efficiency, there is a trade-off regarding accuracy and refinement in discretizing the continuation curve. If large steps are taken, then mapping of the curve may take a small amount of time, but the discretization may not be refined enough to give an accurate representation. The goal of Section 4.2.1 is to develop a fast output method that is an effective compromise between large steps and refined curves.

The book by Allgower and Georg, "Numerical Continuation Methods: An introduction" [6] is a comprehensive resource for general purpose numerical continuation algorithms. The computer code PITCON [80] is a software package for solving general nonlinear systems of the form of Eq. 4.1. Since PITCON is used in Refs. 40, 41, 61 and others for numerical continuation based implementations of workspace analysis and provides a general framework for implementing the enhancements of Sections 4.2, 4.3.1, and 4.3.2, a brief introduction to PITCON follows.

4.1.2 PITCON

As noted in Section 1.3.4, Eq. 1.10 can be used to uniquely determine tangents to a continuation curve. PITCON actually solves the system

$$\begin{bmatrix} \mathbf{G_x}(\mathbf{x}) \\ \mathbf{e}^{iT} \end{bmatrix} \mathbf{T} = \mathbf{e}^{n_x} \tag{4.3}$$

where the natural basis vector, $\mathbf{e}^i$, is the vector with a one in the $i-$th term and zeroes elsewhere. The solution $\mathbf{T}$ is then normalized via

$$\mathbf{h} = \sigma \frac{\mathbf{T}}{||\mathbf{T}||_2} \tag{4.4}$$

where $\sigma = \pm 1$ is selected to determine the direction of the tangent. Note that the vector $\mathbf{h}$ computed in this manner satisfies Eq. 1.10.

PITCON can be viewed as a predictor–corrector process where the prediction, $\hat{\mathbf{x}} = \mathbf{x}_0 + h\mathbf{h} \subset \Re^{n_x}$, is done for a computed step length h in the direction $\mathbf{h}$ and correction is done in the subspace $\hat{\mathbf{x}} + (\text{span } \mathbf{e}^{i_k})^{\perp} \subset \Re^{n_x}$ specified by the following coordinate splitting involving the natural basis vectors $\mathbf{e}^1, \ldots \mathbf{e}^{n_x}$ of $\Re^{n_x}$:

$$\Re^{n_x} = \left(\text{span } \mathbf{e}^{i_k}\right)^{\perp} \bigoplus \text{span } \mathbf{e}^{i_k} \tag{4.5}$$

where $\left(\mathbf{e}^{i_k}\right)^T \mathbf{h} \neq 0$ for some chosen index $i_k, 1 \leq i_k \leq n_x$. The correction step is implemented as finding the value of $\mathbf{x}$ that satisfies the augmented system

$$\bar{\mathbf{G}}(\mathbf{x}) \equiv \begin{bmatrix} \mathbf{G}(\mathbf{x}) \\ \mathbf{e}^{i_k T}(\mathbf{x}-\hat{\mathbf{x}}) \end{bmatrix} = \mathbf{0} \tag{4.6}$$

Applying the Newton method to $\bar{\mathbf{G}}(\mathbf{x}) = \mathbf{0}$ results in the linear system

$$\begin{bmatrix} \mathbf{G}_{\mathbf{x}}(\mathbf{x}) \\ \mathbf{e}^{i_k T} \end{bmatrix} \Delta\mathbf{x} = -\bar{\mathbf{G}}(\mathbf{x}) \tag{4.7}$$

with update $\mathbf{x} = \mathbf{x} + \Delta\mathbf{x}$. From either Eq. 4.6 or Eq. 4.7 if can be seen the correction is done so the i_k component of $\mathbf{x}$ is fixed. Observe the similarity to the linear system of Eq. 4.3.

The coordinate splitting of Eq. 4.5 is natural for systems where $\mathbf{x}$ can be partitioned into a state vector $\boldsymbol{\chi} \in \Re^n$ and a parameter $\lambda \in \Re$, to form $\mathbf{x} = \left[\boldsymbol{\chi}^T, \lambda\right]^T$. In this case, continuation can be done to determine the resulting states as λ varies by choosing $\mathbf{e}^{i_k} = \mathbf{e}^{n_x}$. Using the natural basis vectors for splitting is also the best choice when computing *target points.* A target point $\mathbf{x}$ is a point for which the component $x_i = \mathbf{e}^{i^T}\mathbf{x}$ is sought to have a prescribed value. If the continuation process has determined values of x_i close to the prescribed value, then the system of Eq. 4.6 is solved for $\mathbf{x}$, while holding the value of x_i at its prescribed value [80].

Existence, uniqueness, and regularity for all of the above computations is discussed in detail in Ref. 80.

The previous discussion contains all of the details necessary to understand the methodology behind PITCON, except stepsize control. In selecting an appropriate value of h, a measure of the convergence behavior of the corrector iterations is needed. Rheinboldt [80] presents a detailed method for stepsize control, which is only briefly summarized here. The idea is that there is some radius of convergence of the corrector process at each of the computed points and it is desired to extrapolate these radii to the next point to be determined [80]. Den Heijer and Rheinboldt [43] show that using convergence radii of past computed points provides insufficient information to determine a lower bound of the maximal convergence radius at the next point. This negative result shows that only general inference about the convergence radius can be determined. However, this information can be used to determine a heuristic for the steplength algorithm. The convergence quality assessment used by Rheinboldt [80] is based on tracking the changes in the errors during correction, keeping the number of correction iterations required to converge in a specified range, estimating the curvature of the solution relative to the size of the step, and tracking relative growth and absolute size of the step.

4.2 Dense Output

When mapping solution curves, large step sizes may lead to a discretization such that the continuation curve is not refined enough to give an accurate representation. Even when enough data points are present to capture desired information about the solution, spacing between data points may not be optimal for graphical output or additional analysis. ODE solvers deal with this by the use of dense output algorithms. These algorithms, when there is a natural parameterization of the solution such as time or arclength, output data points corresponding to a fixed increment of the parameter, producing evenly spaced data points [30]. These are often based on embedding a lower order method in the algorithm and efficiently using data from the higher order method to give fast approximates of evenly spaced points. Numerical continuation methods differ from ODE methods in the ability to verify acceptability of computed points by checking the magnitude of $\mathbf{G}(\mathbf{x})$. This section presents a dense output scheme for numerical continuation that can be added to PITCON. It takes advantage of information available at execution time that is not typically available for post processing. It can interpolate evenly spaced data points and correct these to the continuation curve if desired. This dense output method complements the improved stepsize method of Section 4.4, to efficiently produce a refined output discretization.

4.2.1 Cubic Hermite Interpolation

After PITCON or a general numerical continuation code reports acceptance of a point, the tangent satisfying Eq. 4.3 or Eq. 1.10 is computed. The tangent at the previous point has been stored and is used in the steplength algorithm to obtain a measure of the curvature of the solution curve. Thus, with the coordinate values and associated tangents from the current and previous data points, there is enough information to form the cubic Hermite interpolating polynomial. This polynomial can be formed for each of the coordinates and used to interpolate any solution point in the current step. It can also be used to extrapolate the next predicted point.

If s is arclength and $f(s)$ is the actual function through a coordinate, then the divided difference form of the cubic Hermite interpolating polynomial at s on interval $[a, b]$ is [9]

$$H_2(s) = f(a) + (s-a)f'(a) + (s-a)^2 f[a,a,b] + (s-a)^2(s-b)f[a,a,b,b]$$

where

$$f[a,b] = \frac{f(b)-f(a)}{b-a}$$
$$f[a,a,b] = \frac{f[a,b]-f'(a)}{b-a}$$
$$f[a,a,b,b] = \frac{f'(b)-2f[a,b]+f'(a)}{(b-a)^2}$$

The cubic Hermite polynomial can be efficiently evaluated from the nested multiplication form of the divided difference formula [9].

The derivative with respect to arclength

$$H_2'(s) = f'(a) + (s-a)\left(2f[a,a,b] + ((s-a) + 2(s-b))\, f[a,a,b,b]\right)$$

yields approximation of the component of the tangent corresponding to the coordinate that was interpolated. Note that the cost of evaluating $H_2'(s)$ is only four additional multiplications and three additions over the cost of evaluating $H_2(s)$.

PITCON reports the pseudo-arclength of the continuation curve, computed as the sum of the Euclidean lengths of each of the computed secants. This can be used as the arclength parameter for the interpolating polynomial, but of more practical use for workspace analysis and boundary surface calculation is the pseudo-arclength of the curve through $\mathbf{u}$ coordinate vector output points. This choice will allow the *output* $\mathbf{u}$–space to be discretized on an evenly space grid. This choice requires that if no change in the $\mathbf{u}$–space coordinates occurs for an accepted solution curve point, that the interpolation not be done since $(b-a) = 0$. However, this is desired because only points yielding changes in the output $\mathbf{u}$–space will be used in graphical representation of the boundary surface, and in this case interpolation is not necessary.

Once all coordinates are interpolated for each desired grid point $\bar{\mathbf{x}}$, which falls in the current interval $[a, b]$, the magnitude of the error $||\mathbf{G}(\bar{\mathbf{x}})||$ can be calculated. This can be corrected by performing a Newton type iteration on the system of Eq. 4.6. In fact, PITCON has the factored matrix from Eq. 4.7 available in memory and it can be used to perform modified Newton iteration, where the Jacobian is not re-evaluated at each step, and only the right hand side is updated at each iteration. This modified Newton iteration, using the available factored Jacobian, works well when PITCON takes small steps relative to desired interpolation spacing, or if the interpolation error is small.

However, this modified Newton method was tried for several examples and was found to perform poorly in many cases for two reasons. The first reason is that when the predicted step length is large the factored matrix has a Jacobian that is evaluated at end point b of the interval over which interpolation is done. There can be substantial difference in this Jacobian, compared to the value at end point a of the interpolating interval. The second apparent reason is that the choice of the natural basis vector $\mathbf{e}^{i_k}$ used in PITCON, while not interfering with the correction procedure, is not the best choice for correcting the interpolated coordinates to the *nearest* point on the curve being mapped. If an approximate tangent to the interpolated point, computed using the derivative of the cubic Hermite interpolating polynomial is used in place of the $\mathbf{e}^{i_k}$ of Eq. 4.7, the correction process updates the interpolated coordinates in a subspace that is orthogonal to the curve being mapped. See Section 4.3.2 for details. This is thus is the best choice for finding the *nearest* point. Correcting the interpolated points in this manner nearly preserves the even spacing that was used while interpolating, as can be seen in the example of Section 4.2.3.

4.2.2 Adding Dense Output to Continuation Codes

In order to use the interpolation procedure of the previous section the control sequence for the continuation code needs to be modified. In Fig. 4.1, a modified control flow diagram is presented. It uses five additional control flow variables. The variable **outstep** is the desired fixed arclength step size. Variables **curpoint** and **totalarc** are used to track the arclength, when **curpoint** the value at the last interpolated point and **totalarc** is the total arclength to the last computed continuation point. Integer variables **idimout** and **intrpdim** control what type of interpolation is done. If **intrpdim** is zero, no interpolation is done and if **intrpdim** is n_x, then correction is done on interpolated values. For any other value of **intrdim** between

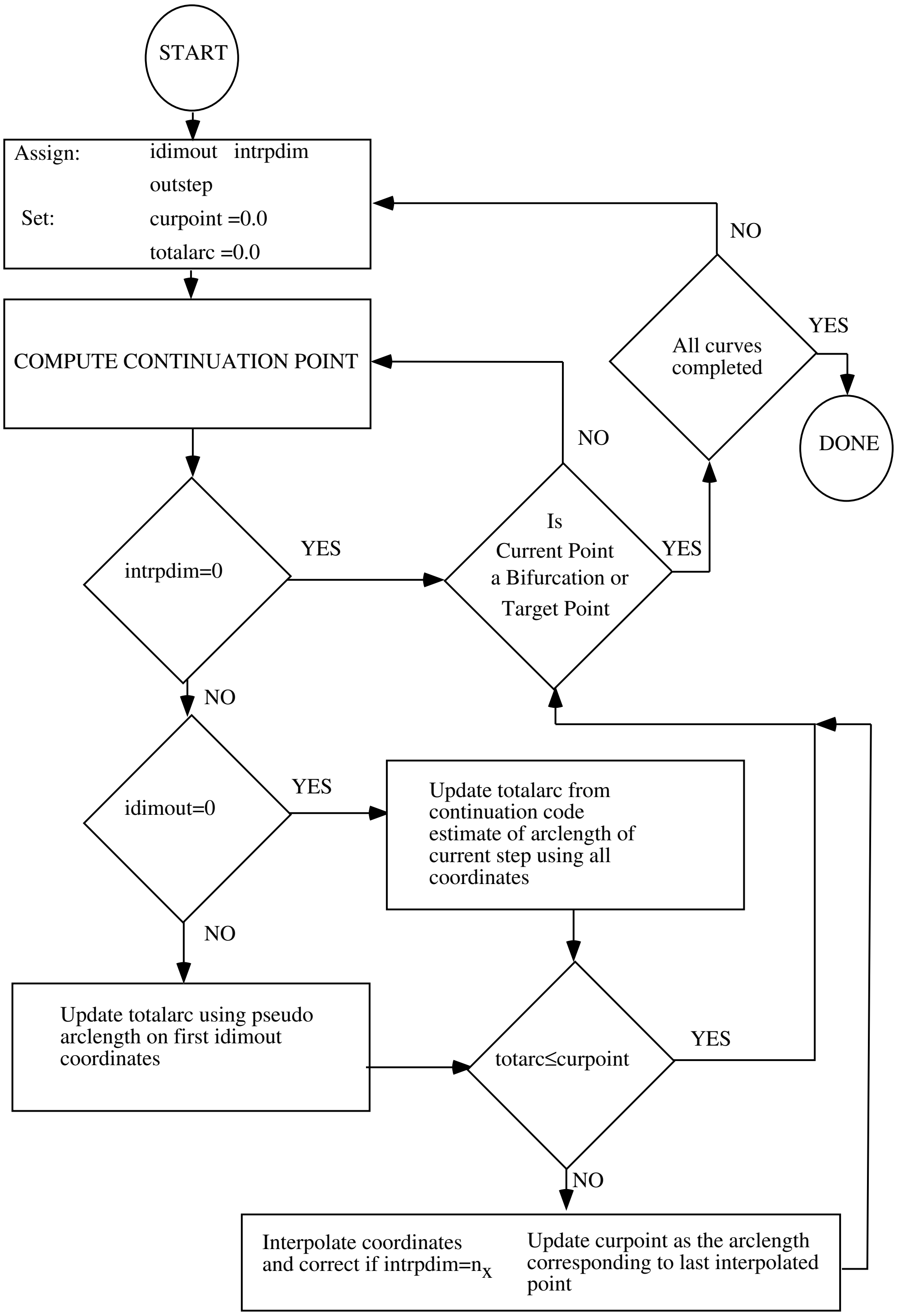

Figure 4.1: Continuation Flow Diagram with Dense Output

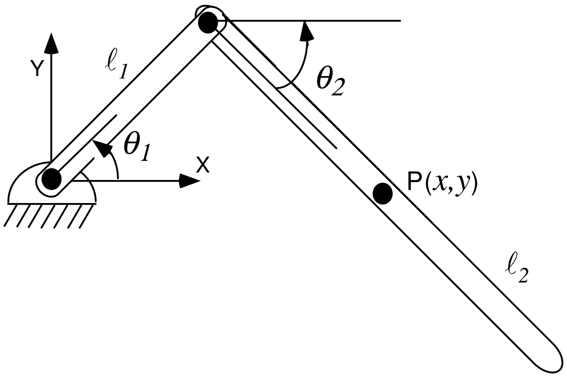

Figure 4.2: Two Bar Manipulator

zero and n_x, interpolation will be done on the first **intrpdim** components of $\mathbf{x}$. If **idimout** is zero, then the arclength estimate from the continuation code using all components of $\mathbf{x}$ will be used. Otherwise, the first **idimout** components of $\mathbf{x}$ will be used to compute the arclength estimate.

4.2.3 Numerical Example–Two Bar Manipulator

In order to provide a challenging test for the dense output algorithm, an example with large curvature in the $\mathbf{u}$–space, but where the numerical continuation can still take large steps, is desired. A straight line with small steps would not illustrate the effectiveness of the interpolation method. The open chain structure of the two-bar manipulator shown in Fig. 4.2 reduces dependencies among the coordinates and the numerical continuation code converges with relatively large steps, compared to curvature. The workspace boundary is a circle and thus provides a challenging curve for the interpolation routine to approximate.

The continuation equations for the two-bar manipulator are

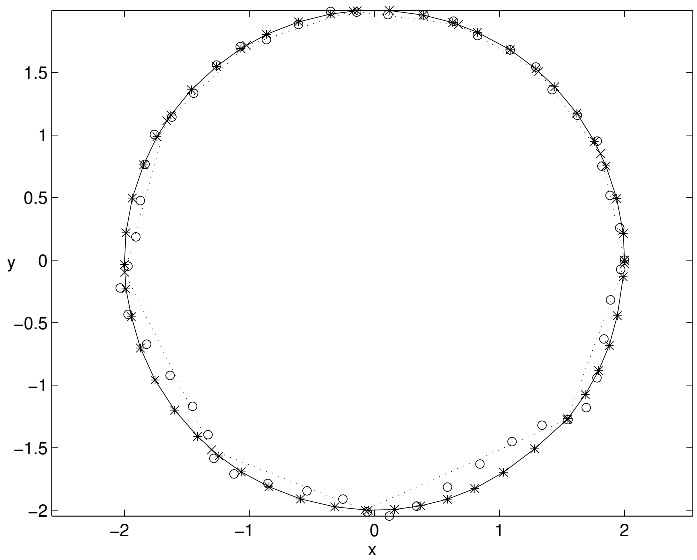

Figure 4.3: Interpolation for Accessible Output Set of Two Bar Manipulator

$$\mathbf{G}(\mathbf{x}) = \begin{bmatrix} x - \ell_1 cos(\theta_1) - \frac{\ell_2}{2.0} cos(\theta_2) \\ y - \ell_1 sin(\theta_1) - \frac{\ell_2}{2.0} sin(\theta_2) \\ \ell_1 sin(\theta_1)\tau_1 - \ell_1 cos(\theta_1)\tau_2 \\ \frac{\ell_2}{2.0} sin(\theta_2)\tau_1 - \frac{\ell_2}{2.0} cos(\theta_2)\tau_2 \\ \tau_1^2 + \tau_2^2 - 1 \end{bmatrix} \tag{4.8}$$

where the vector $\mathbf{x} = [x, y, \theta_1, \theta_2, \tau_1, \tau_2]^T$. Bar lengths were set at $\ell_1 = 1.0$ and $\ell_2 = 2.0$ for the calculations described below.

The boundary of the workspace in the $\mathbf{u} = [x, y]^T$–space is shown in Fig. 4.3. The points marked by x's are the original points on the boundary, as computed by PITCON. The points marked by $\circ$'s are the initial cubic Hermite interpolated points, and the corrected interpolated points are marked with $*$'s. The interpolated points were computed to be spaced by 0.25 units in the output space. From Fig. 4.3, it can be seen that the piecewise linear curve formed by connecting the initial cubic Hermite interpolated points produces a better boundary representation than the dotted piecewise linear line through the PITCON calculated boundary curves. These points are computed with very little additional computation time as shown in Table 4.1. The method of augmenting the Jacobian in Eq. 4.7 with the Hermite tangent was used to correct the interpolated values. The computation time approximately doubles to correct the interpolated points to a tolerance of less than 1.0e-6 on $||\mathbf{G}(\mathbf{x})||_2$. This correction time, however, is about half that required to have PITCON compute using a maximum stepsize selected to have *on average* the output space point printed on a grid of 0.25 pseudo-arclength between them. Because PITCON does stepsize control on the norm of the entire coordinate $\mathbf{x}$, the maximum step was found by trial and error, which yielded on average 0.25 pseudo-arclength of the output coordinate. The computation times in Table 4.1 are in CPU seconds on a Hewlett Packard HP9000/755 computer. The timings presented are actually for the computation of three times around the boundary curve. This is necessary to get appropriate resolution on the CPU clock. Much of the computation cost reduction of the dense output method over using PITCON iteration with small steps, is due to the ability to have fast computation of Hermite tangent approximations that require no linear system solutions of the form of Eq. 4.3.

Figure 4.4 gives a close up view of a portion of the boundary curve from Fig. 4.3. Points labeled with +'s are added to show the correction steps that were taken from each interpolated point. Notice how constant spacing is nearly preserved after correction.

4.3 Special Purpose Numerical Continuation Method- Theory and Ideas

Two possible enhancements to the solution procedure used by PITCON are

Table 4.1: Performance in CPU Seconds of Discretization of Two Bar Accessible Output Set

Compute Method	Syst. Time	User Time	Total Time
Regular PITCON Steps Only	0.01	0.27	0.28
Add Hermite Interpolation of Output	0.01	0.28	0.29
Add Newton Correction	0.01	0.55	0.56
PITCON with Corresponding Small Step	0.01	1.02	1.03

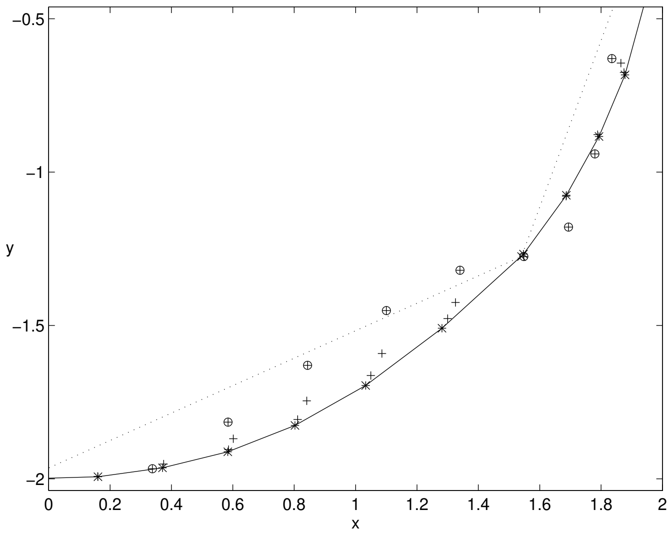

Figure 4.4: Interpolation Detail for Accessible Output Set of Two Bar Manipulator

suggested here.

4.3.1 Changing Correction Procedure

Close observation of the subequations of Eq. 4.2, reveals that information about the action of the mechanism or attached body geometry is contained only in the equations $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$. If a cutting plane is necessary for reduction of the boundary curve to one-dimensional sets, assume that $\mathbf{\Phi}(\mathbf{q})$ has been augmented with the cutting plane equation $\mathbf{L}^T\mathbf{u} - b = 0$, as the grouping in Eq. 4.2 indicates. For the remainder of this discussion, reference to $\mathbf{\Phi}(\mathbf{q})$ implies this augmented collection of equations.

One modification to the correction procedure is to consider a sequence of steps in which correction at each step is done based on a portion of the subequations of Eq. 4.2. First consider the subequation $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$. It is shown that when correcting a predicted estimate $\hat{\mathbf{x}} = \left[\hat{\mathbf{q}}^T, \hat{\boldsymbol{\tau}}^T\right]^T$ of a point to the solution curve $\mathbf{G}(\mathbf{x}) = \mathbf{0}$, more information about the choice of stepsize will be obtained by first correcting $\hat{\mathbf{q}}$ to $\hat{\mathbf{q}}^{(k)}$, so that $\mathbf{\Phi}(\hat{\mathbf{q}}^{(k)}) = \mathbf{0}$, and then approaching the boundary, i.e., $\mathbf{G}(\hat{\mathbf{x}}^{(k^*)}) = \mathbf{0}$, along the sub-manifold $\mathbf{\Phi}(\hat{\mathbf{q}}^{(k^*)}) = \mathbf{0}$.

Denoting $\mathbf{y} = \left[\mathbf{u}^T, \mathbf{w}^T\right]^T$, the correction to $\mathbf{\Phi}(\hat{\mathbf{q}}^{(k)}) = \mathbf{0}$ will, when possible, be done using the square Jacobian iteration matrix of $\mathbf{\Phi}_{\mathbf{y}}(\hat{\mathbf{q}})$, by solving

$$\mathbf{\Phi}_{\mathbf{y}}(\hat{\mathbf{q}})\Delta\mathbf{y}^{(k)} = -\mathbf{\Phi}(\hat{\mathbf{y}}^{(k)}, \hat{\mathbf{v}}); \qquad \hat{\mathbf{y}}^{(k+1)} = \hat{\mathbf{y}}^{(k)} + \Delta\mathbf{y}^{(k)} \tag{4.9}$$

Why does this make sense? First, by predicting a solution and then solving to the correct $\hat{\mathbf{y}}^{(k)}$ that matches the predicted $\hat{\mathbf{v}}$, actually uses the input control coordinate $\mathbf{v}$ to determine the configuration of the mechanism. Second, while following the boundary curve, $\mathbf{\Phi}_{\mathbf{z}}$ is required to be row rank deficient; thus failing to satisfy a necessary condition for the implicit function theorem to imply that there exists some function $\mathbf{z} = \mathbf{f}(\mathbf{u})$ such that $\mathbf{\Phi}\left(\mathbf{u}, \mathbf{f}\left(\mathbf{u}\right)\right)$. In fact, since $\mathbf{z} = \left[\mathbf{w}^T, \mathbf{v}^T\right]^T$, $\mathbf{\Phi}_{\mathbf{v}}$ is a matrix with as many rows as $\mathbf{\Phi_z}$, but fewer columns. It is guaranteed also to be row rank deficient, when $\mathbf{\Phi_z}$ is, again failing a necessary condition for the implicit function theorem to imply that there exists some function $\mathbf{v} = \bar{\mathbf{f}}(\mathbf{u}, \mathbf{w})$ such that $\mathbf{\Phi}(\mathbf{u}, \mathbf{w}, \bar{\mathbf{f}}(\mathbf{u}, \mathbf{w})) = \mathbf{0}$ for all $\mathbf{u} \in \mathbf{U}$, $\mathbf{w} \in \mathbf{W}$ and $\mathbf{v} \in \mathbf{V}$ neighborhoods of boundary curve points. However, by appropriate partitioning of the generalized coordinate, as was described in Section 3.2.1, $\mathbf{\Phi_y}(\hat{\mathbf{q}})$ can be selected so that it has full row rank, in fact is invertible as long as $\hat{\mathbf{q}}$ is not too poor an estimate of a mechanism and shape configuration. Then, by the implicit function theorem, there exists $\mathbf{y} = \mathbf{f}(\mathbf{v})$ where $\mathbf{\Phi}(\mathbf{y}, \mathbf{f}(\mathbf{v})) = \mathbf{0}$ for all $\mathbf{y} \in \mathbf{Y}$ and $\mathbf{v} \in \mathbf{V}$ neighborhoods of boundary curve points. Thus, if $\hat{\mathbf{v}}$ is close enough to a feasible mechanism input, the Newton iteration of Eq. 4.9 should converge. Third, from previous numerical experience that demonstrates boundary curves can be parameterized with the input coordinates, usually only one or two of the input coordinates are changing along a 2 or 3 dimensional output space boundary curve. It is expected, for the solution $\hat{\mathbf{q}}^{(k)}$ obtained from the iteration of Eq. 4.9, that the mechanism will be configured close to the boundary curve and it remains only to find, within tolerance, the non-zero vector $\boldsymbol{\tau}$ such that Eq. 1.3 holds. The linearity of the $\mathbf{\Phi_z}^T\boldsymbol{\tau}$ can be exploited by finding $\boldsymbol{\tau}$ to solve a least squares type problem or to find the smallest eigenvalue and associated eigenvector of matrices related to $\mathbf{\Phi_z}$.

The least squares solution [9] for $\boldsymbol{\tau}$ from the following system of linear equations can be used to update $\hat{\boldsymbol{\tau}}$

$$\begin{bmatrix} \mathbf{\Phi_z}^T(\hat{\mathbf{q}}^{(k)}) \\ \hat{\boldsymbol{\tau}}^T \end{bmatrix} \boldsymbol{\tau} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \tag{4.10}$$

Note that because $\hat{\boldsymbol{\tau}}$ provides an estimate of $\boldsymbol{\tau}$ this system does not need to be solved for the collection $\mathbf{e}_i,\ i = 1,\ \ldots,\ m$ as was done in Eq. 1.7.

The least squares solution of Eq. 4.10 is the solution $\boldsymbol{\tau}$ of

$$(\boldsymbol{\Phi}_{\mathbf{z}}\boldsymbol{\Phi}_{\mathbf{z}}^T + \hat{\boldsymbol{\tau}}\hat{\boldsymbol{\tau}}^T)\boldsymbol{\tau} = \hat{\boldsymbol{\tau}} \tag{4.11}$$

Since the solution will not exactly satisfy $\boldsymbol{\tau}^T\boldsymbol{\tau} = 1$, the solution may need to be normalized.

Another possible update for $\hat{\boldsymbol{\tau}}$ is to select $\boldsymbol{\tau}$ as the eigenvector corresponding to the *smallest* eigenvalue of the matrix $\boldsymbol{\Phi}_{\mathbf{z}}\boldsymbol{\Phi}_{\mathbf{z}}^T$. Note that, while this matrix is symmetric, it is expected to be nearly singular and $\boldsymbol{\tau}$ will correspond to the smallest eigenvalue.

If this solution procedure is adopted, where eigenvalues of $\boldsymbol{\Phi}_{\mathbf{z}}\boldsymbol{\Phi}_{\mathbf{z}}^T$ are computed, this yields an extra item of information that is of interest for certain dexterous workspace questions. In Haug, Wang, and Wu [41], it is noted that the smallest eigenvalue of $\boldsymbol{\Phi}_{\mathbf{z}}\boldsymbol{\Phi}_{\mathbf{z}}^T$ should be tracked for a sweep of an orientation parameter. Computation of this result yields one such point in the sweep.

4.3.2 Changing Augmenting Vector

As mentioned in Section 4.1.2, the workspace coordinates do not have a natural partitioning into a state vector with one parameter as $\mathbf{x} = \left[\boldsymbol{\chi}^T, \lambda\right]^T$, where it makes sense to use the natural basis vector $\mathbf{e}^{n_x}$ as the augmenting vector in the system Eq. 4.7.

The basic problem to be solved is finding $\mathbf{x} = \hat{\mathbf{x}} + \Delta\mathbf{x}$ such that $\mathbf{G}(\mathbf{x}) = \mathbf{0}$. This can be rewritten as

$$\mathbf{0} \equiv \mathbf{G}(\hat{\mathbf{x}} + \Delta\mathbf{x}) = \mathbf{G}(\hat{\mathbf{x}}) + \mathbf{G}_{\mathbf{x}}(\hat{\mathbf{x}})\Delta\mathbf{x} + \ldots \tag{4.12}$$

where the Taylor expansion is used. A solution $\Delta\mathbf{x} = \mathbf{s}$ to the underdetermined system

$$\mathbf{G}_{\mathbf{x}}(\hat{\mathbf{x}})\mathbf{s} = -\mathbf{G}(\hat{\mathbf{x}}) \tag{4.13}$$

satisfies Eq. 4.12 to first order.

Since on regular boundary curve points the Jacobian, $\mathbf{G}_{\mathbf{x}}$, is a $\Re^{n_x-1} \times \Re^{n_x}$ full row rank matrix, there exists a unique $\mathbf{h}$ such that $\mathbf{G}_{\mathbf{x}}\mathbf{h} = \mathbf{0}$ and $\mathbf{h}^T\mathbf{h} = 1$ [6]. The Moore-Penrose inverse, $\mathbf{G}_{\mathbf{x}}^{+} \equiv \mathbf{G}_{\mathbf{x}}^{T}\left(\mathbf{G}_{\mathbf{x}}\mathbf{G}_{\mathbf{x}}{}^{T}\right)^{-1}$ [6], provides a solution $\mathbf{s} = \mathbf{G}_{\mathbf{x}}^{+}\left(-\mathbf{G}(\hat{\mathbf{x}})\right)$ to the system Eq. 4.13.

Using the definition of the Moore-Penrose inverse,

$$\begin{bmatrix} \mathbf{G}_{\mathbf{x}} \\ \mathbf{h}^T \end{bmatrix} \begin{bmatrix} \mathbf{G}_{\mathbf{x}}^{+} & \mathbf{h} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{\mathbf{x}}\mathbf{G}_{\mathbf{x}}^{+} & \mathbf{G}_{\mathbf{x}}\mathbf{h} \\ \mathbf{h}^T\mathbf{G}_{\mathbf{x}}^{+} & \mathbf{h}^T\mathbf{h} \end{bmatrix} = \mathbf{I} \tag{4.14}$$

so

$$\begin{bmatrix} \mathbf{G}_{\mathbf{x}} \\ \mathbf{h}^T \end{bmatrix} \begin{bmatrix} \mathbf{G}_{\mathbf{x}}^{+} & \mathbf{h} \end{bmatrix} \begin{bmatrix} -\mathbf{G}(\hat{\mathbf{x}}) \\ 0 \end{bmatrix} = \mathbf{I} \begin{bmatrix} -\mathbf{G}(\hat{\mathbf{x}}) \\ 0 \end{bmatrix} \tag{4.15}$$

implies

$$\begin{bmatrix} \mathbf{G}_{\mathbf{x}} \\ \mathbf{h}^T \end{bmatrix} \mathbf{G}^{+}\left(-\mathbf{G}(\hat{\mathbf{x}})\right) \equiv \begin{bmatrix} \mathbf{G}_{\mathbf{x}} \\ \mathbf{h}^T \end{bmatrix} \mathbf{s} = \begin{bmatrix} -\mathbf{G}(\hat{\mathbf{x}}) \\ 0 \end{bmatrix} \tag{4.16}$$

which yields

$$\mathbf{h}^T\mathbf{s} = 0 \tag{4.17}$$

Since the null space of $\mathbf{G}_{\mathbf{x}}$ is span $\mathbf{h}$, solutions for $\Delta\mathbf{x}$ must be of the form

$$\Delta\mathbf{x} = \alpha\mathbf{h} + \mathbf{s} \tag{4.18}$$

Equation 4.17 shows that $\mathbf{h}$ and $\mathbf{s}$ are orthogonal, implying

$$||\Delta\mathbf{x}||_2^2 = \alpha^2 + ||\mathbf{s}||_2^2 \tag{4.19}$$

and it is clear that the shortest correction distance $\Delta\mathbf{x}$ occurs when $\alpha = 0$. Solving for $\Delta\mathbf{x}$ from the system

$$\begin{bmatrix} \mathbf{G}_{\mathbf{x}} \\ \mathbf{h}^T \end{bmatrix} \Delta\mathbf{x} = \begin{bmatrix} -\mathbf{G}(\hat{\mathbf{x}}) \\ 0 \end{bmatrix} \tag{4.20}$$

thus produces the best correction update.

The tangent $\mathbf{h}$ is readily available and can be used in place of $\mathbf{e}^{i_k}$ of Eq. 4.7. This solution procedure is essentially equivalent to applying the Newton method in solving

$$\hat{\mathbf{G}}(\mathbf{x}) \equiv \begin{bmatrix} \mathbf{G}(\mathbf{x}) \\ \mathbf{h}^T(\mathbf{x} - \hat{\mathbf{x}}) \end{bmatrix} = \mathbf{0} \tag{4.21}$$

Note that $\mathbf{h}$ is not independent of the sequence of iterates $\mathbf{x}$. If the standard Newton method were applied, this would require updating $\mathbf{h}$ at each step. The modified Newton method where either $\mathbf{h}$ or $\mathbf{h}$ and $\mathbf{G}_{\mathbf{x}}$ are held fixed, should still converge fast enough relative to the additional time required to determine $\mathbf{h}$ at each step. If $||\mathbf{h}||_\infty > 0.98$, it is fast enough to use the iteration scheme of Eq. 4.7, since evaluating the residual term ${\mathbf{e}^{i_k}}^T(\mathbf{x} - \hat{\mathbf{x}}) = (\Delta\mathbf{x})^{i_k}$ is faster than $\mathbf{h}^T(\mathbf{x} - \hat{\mathbf{x}})$.

When correcting a coordinate predicted by the Hermite dense output interpolation it is not necessarily desired to have the smallest Δx update. In this case it is sought to minimize the change in the output coordinate, i.e., the norm of the components of Δx corresponding to the $\mathbf{u}$-space coordinates. If $\mathbf{h} = \left[{\mathbf{h}_{\mathbf{u}}}^T, {\mathbf{h}_{\mathbf{z}}}^T, {\mathbf{h}_{\boldsymbol{\mathcal{T}}}}^T\right]^T$ and $\mathbf{s} = \left[{\mathbf{s}_{\mathbf{u}}}^T, {\mathbf{s}_{\mathbf{z}}}^T, {\mathbf{s}_{\boldsymbol{\mathcal{T}}}}^T\right]^T$, then this restricted measure of Δx, expressed as a function of α, is $f(\alpha) = ||\alpha\mathbf{h}_{\mathbf{u}} + \mathbf{s}_{\mathbf{u}}||_2^2$. The necessary condition for $f(\alpha)$ to be minimum is that $\frac{\partial f(\alpha)}{\partial \alpha} = 0$. If $\mathbf{h}_{\mathbf{u}} \neq \mathbf{0}$ this condition implies that

$$\alpha = \frac{\mathbf{h}_\mathbf{u}^T \mathbf{s}_\mathbf{u}}{\mathbf{h}_\mathbf{u}^T \mathbf{h}_\mathbf{u}} \tag{4.22}$$

If $\mathbf{h}_\mathbf{u} = \mathbf{0}$, then choose $\alpha = 0$. Thus to achieve the smallest change in the output coordinate select α as above and $\Delta x = \alpha \mathbf{h} + \mathbf{s}$.

4.4 Special Purpose Numerical Continuation Method- Stepsize Selection

While the prediction and correction procedure for differential-algebraic equations is similar to the process of general numerical continuation methods, the step size acceptance criteria differs in that the differential equation solver needs to assess if the predicted solution corresponds to a correct amount of time, arclength or an other parameter that should have elapsed for the selected step, whereas the continuation method is simply concerned with finding a future candidate solution on the curve $\mathbf{G}(\mathbf{x}) = \mathbf{0}$. This requires that the differential equation solver use some method of checking its solution after the solution is computed, typically imbedded methods of higher orders, or some PECE method. However, the continuation method can check, after the solution is computed, if it satisfies the condition $\mathbf{G}(\mathbf{x}) = \mathbf{0}$ and a quick estimate of arclength will determine if the solution is close enough to the desired predicted value. Continuation methods evaluate the change from the predicted value to the corrected solution, or track the number of correction steps needed, and this is used to determine the size of the next prediction step. The benefits of being able to check the accuracy of the solution, simply by measuring $\|\mathbf{G}(\mathbf{x})\|$ during correction iteration, are weakened by the fact that convergence is not guaranteed for a given step size h; thus several linear system solutions and Jacobian calculations may be computed before a step is rejected. More information about the acceptablitity of a chosen step size h, before iterating on the system $\mathbf{G}(\mathbf{x}) = \mathbf{0}$, could improve computational performance by reducing the amount of rejected steps.

For the case of mapping the manifold boundary, using Eq. 4.2, additional information can also be obtained for step size control from partitioning the variables. This information will be further enhanced by solving the continuation equations using the enhancements of Section 4.3.

Since information about the acceptablitity of the size of the current step must be considered in determination of acceptance of the length of the current and future prediction directions, any information obtained during the correction procedure is valuable. The generalized coordinate $\mathbf{q}$ has all of the information about the mechanism configuration and motion. Thus, the iteration process of Eq. 4.9 will capture information about the acceptability of the prediction $\hat{\mathbf{q}}$ with regard to both difficulty in convergence and the amount of correction that occurs between prediction $\hat{\mathbf{q}}$ and iterates $\hat{\mathbf{q}}^{(k)}$. Since Eq. 4.9 iterates on only a logically selected portion of the continuation equation, the current stepsize can be modified, if convergence on this portion is slow, before the larger full system will need to be formulated and solved.

The code PITCON uses a factor to estimate the convergence quality of each accepted step. The number of correction steps is one of the parameters on which this convergence quality factor depends. The information and heuristic used in PITCON's convergence quality and related stepsize selection is based on sound theory and is well tested [80]. This stepsize selection routine can be used for the method of the previous section, if a value for the number of correction steps can be obtained. Since the full system is not being iterated on in the method of the previous section, this value must be based on the sum of any full system iterations, J, and a factor relating the number of iteration of Eq. 4.9 or the magnitude of the change of components in this iteration procedure.

If K is the number of iterations of Eq. 4.9 and $J = 0$, because the $\boldsymbol{\tau}$ was found by solution rather than by iteration or $J = 1$ because the $\boldsymbol{\tau}$ term was found by one full system iteration following the K iterations of Eq. 4.9, then choose the value of the number of correction iterations as expected by PITCON to be $K + J - 1$. If the iteration procedure of Eq. 4.9 concluded, at a point that required $J > 1$ full system iterations, assume that the coordinate found by the Eq. 4.9 iteration was relatively far from the boundary, hence the chosen stepsize is on the large side, and set the value for PITCON as $\alpha K + J - 1$ where $\alpha > 1$, $\alpha = 1.2$ is suggested.

If the iteration process was aborted in the solution of Eq. 4.9, then the stepsize has already been reduced and the PITCON iteration count will be based on the totals previously described for the iteration process of the accepted stepsize.

4.5 Special Purpose Numerical Continuation Method- Examples and Performance

The two bar manipulator of Fig. 4.2 and the planar Stewart platform of Fig. 1.3 are used to test the methods in previous sections of this chapter. Note that these two examples represent very different mechanical systems. The two bar manipulator is an open chain mechanism whereas the planar Stewart platform is a closed loop mechanism. The planar Stewart platform has bifurcation points on the accessible output set boundary, but and two bar manipulator does not. These differences make the combination of these two problems a good test collection for these numerical continuation methods.

4.5.1 Planar Stewart Platform

The accessible output set was calculated using both regular PITCON steps, which fix the i_kth coordinate value in correction using $\mathbf{e}^{i_k}$ in Eq. 4.7, and the method using the computed tangent in place of $\mathbf{e}^{i_k}$, as in Eq. 4.20. In both cases, the stepsize control

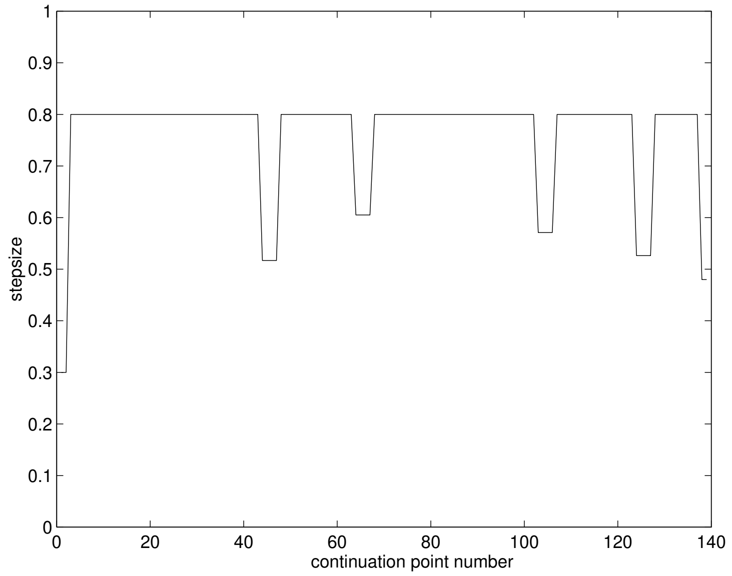

Figure 4.5: Stepsize Selection for PITCON basis vector augmentation

algorithm used is as described in Ref. 80, where iteration counts for correction are passed to PITCON. Due to the nature of the bifurcation points and scaling of the problem, a maximum step size of 0.8 units was imposed to force computation of at least one continuation point on curves separated by bifurcation points. If this is not done, large steps could be permitted by the step size algorithm, such that the starting step leaving a bifurcation point may initially jump past the next bifurcation point, and the the process becomes one only of determining the bifurcation points.

As a measure of performance of the correction iteration procedure for the different ways of augmenting the Jacobian of Eq. 4.7, the selected stepsize at each accepted continuation point can be graphed versus continuation point number. Figure 4.5 presents this comparison for correction using regular PITCON basis vector, $\mathbf{e}^{i_k}$ augmentation. Figure 4.6 presents this comparison for correction augmentation using the tangent vector. As shown in Fig. 4.6, good convergence of the corrector step

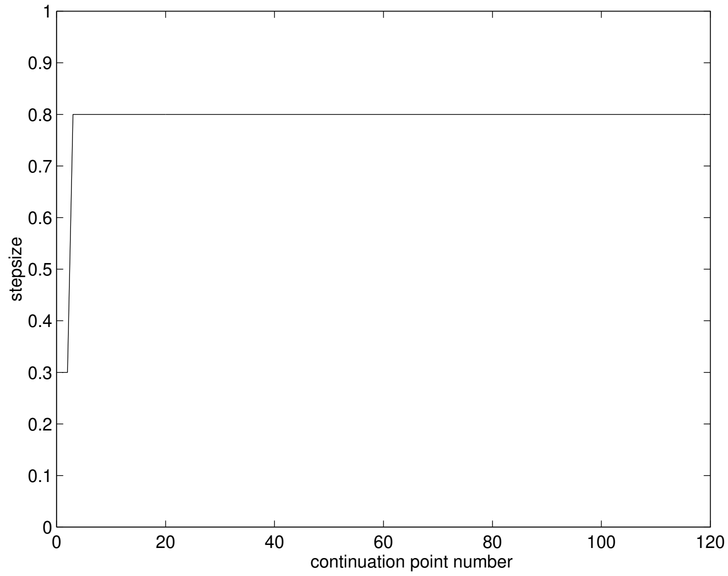

Figure 4.6: Stepsize Selection for PITCON tangent vector augmentation

increases the stepsize to the maximum on the second step and the convergence rate stays appropriately high enough for maximum stepsize selection for the entire solution procedure. As Fig. 4.5 shows, the regular PITCON basis vector creates slow convergence, which in turn causes a reduction in the stepsize. When comparing Figs. 4.5 and 4.6, note that the total number of calculated continuation points is different, the step length reductions require 20 additional points to be computed in order to complete the accessible output set when using PITCON basis vector augmentation.

4.5.2 Two Bar Manipulator

Comparison similar to that of the Section 4.5.1 was done for the two bar manipulator of Fig. 4.2. Here, three schemes for replacing the basis vector $\mathbf{e}^{i_k}$ of Eq. 4.7 are used. Two of the schemes are as described in Section 4.5.1 and the third is one in which the components of the tangent corresponding to the input coordinates are selected for the augmenting vector and the remaining entries are set to zero. A maximum stepsize

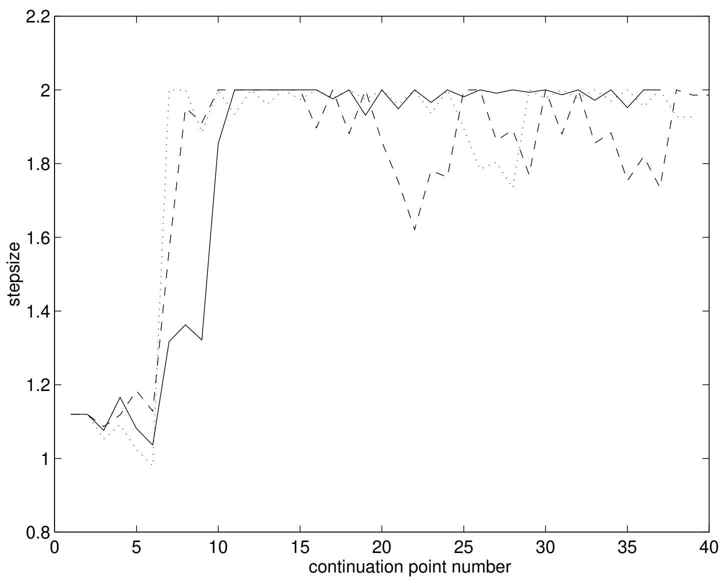

Figure 4.7: Stepsize for Various Augmentations with Two Bar Manipulator

of 2.0 units was selected for these comparisons. The boundary calculation was set to complete after tracing the outer boundary curve three times by setting the a target point $\theta_1 = 6\pi$ since the starting value $\theta_1 = 0$ was used.

In Fig. 4.7, it can be seen that the use of the PITCON basis vector causes difficulty in convergence and stepsize. The dashed line is the stepsize selection when using the PITCON augmenting basis vectors. The solid line corresponds to using the tangent as augmenting vector. The dotted line is the one that corresponds to selecting only the input $\mathbf{v}$–space components from the tangent for the augmenting vector. This performs better than the natural basis vector, but not as good as using the tangent to augment Eq. 4.7. Observe from Fig. 4.7 that 40 continuation points were needed to complete the mapping when the PITCON natural basis augmenting vector was used, and only 36 points were required for the tangent as augmenting vector case.

CHAPTER 5
ANALYSIS OF BARRIERS TO CONTROL OF MANIPULATORS WITHIN ACCESSIBLE OUTPUT SETS

Barriers to output control of manipulators, both in the interior and at the boundary of accessible output sets, are analyzed using first and second order Taylor approximations of the output in selected directions as functions of manipulator input. The formulation is valid for both planar and spatial manipulators, with open chain and closed loop structures, and accounts for the effects of unilateral constraints on the range of admissible control inputs. Criteria defining curves and surfaces associated with singular output control of manipulators are extended to define normals to such curves and surfaces. It is shown that output velocity in the direction normal to such curves and surfaces must be zero, so they are barriers to velocity control in the associated manipulator configuration. Second order Taylor expansion of normal output with respect to input parameters yields quantitative information regarding barriers to output position control. Definiteness properties of the resulting quadratic approximation define directions of admissible and inadmissible outputs. Algorithms for automatically computing the associated quadratic forms and eigenvalues that determine their definiteness properties are presented and illustrated using planar examples.

5.1 Introduction

Analytical criteria for defining the boundary of accessible output sets involve rank deficiency of Jacobian matrices associated with kinematic constraints that describe admissible motion of the manipulator. As shown by Haug, Luh, Adkins, and Wang [40] and in literature cited therein, the exterior boundaries of such accessible output sets are included in a family of singular curves or surfaces, many of which are interior to such sets. Earlier studies by Tsai and Waldron [93] and Nielsen, Canudas deWit, and Hagander [73] have shown that such interior curves and surfaces are often associated with restrictions on motion control. To date, however, no analytical criteria have been presented that characterize the nature of output restrictions that exist in crossing such singular curves and surfaces. The purpose of this chapter is to develop, numerically implement, and illustrate such criteria.

As before, consider a manipulator whose generalized coordinates comprise the vector $\mathbf{q} \equiv \left[\mathbf{u}^T, \mathbf{w}^T, \mathbf{v}^T\right]^T$. Recall that the kinematics of the mechanism is defined by m independent holonomic constraints of the form of Eq. 1.1; i.e.,

$$\boldsymbol{\Phi}\left(\mathbf{q}\right) = \mathbf{0} \tag{5.1}$$

where the vector function $\boldsymbol{\Phi}\left(\mathbf{q}\right)$ now is assumed to have at least three continuous derivatives. Recall the accessible output set is the set of outputs $\mathbf{u}$ for which there are $\mathbf{v}$ and $\mathbf{w}$ satisfying Eq. 5.1; i.e.,

$$A = \left\{\mathbf{u} \in \Re^{n_u} : \boldsymbol{\Phi}\left(\mathbf{u},\ \mathbf{w},\ \mathbf{v}\right) = \mathbf{0},\ \text{for some } \mathbf{v} \textit{ and } \mathbf{w}\right\}$$

For analytic convenience in characterizing boundaries of accessible output sets, the subvector of input and intermediate generalized coordinates $\mathbf{z} \equiv \left[\mathbf{w}^T, \mathbf{v}^T\right]^T$ is defined and an augmented vector $\mathbf{x} \equiv \left[\mathbf{q}^T, \boldsymbol{\tau}^T\right]^T$ is introduced, where the vector $\boldsymbol{\tau}$ is introduced to characterize row rank deficiency of the sub-Jacobian matrix $\boldsymbol{\Phi}_{\mathbf{z}}$.

As shown by Haug, Luh, Adkins, and Wang [40] and in references cited therein, a necessary condition for curves or surfaces in the boundary of the accessible output set, denoted ∂A, is

$$\mathbf{G}^*(\mathbf{x}) \equiv \begin{bmatrix} \mathbf{\Phi}(\mathbf{q}) \\ \mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\tau} \\ \boldsymbol{\tau}^T \boldsymbol{\tau} - 1 \end{bmatrix} = \mathbf{0} \tag{5.2}$$

That is, $\partial A \subset S \equiv \{\mathbf{u} \in \Re^{n_u} : \mathbf{G}^*(\mathbf{x}) = \mathbf{0}\}$.

5.2 Barriers to Output Velocity Normal to S

The function $\mathbf{G}^*(\mathbf{x})$ in Eq. 5.2 is a mapping from $\Re^{n_x}$ to $\Re^{n_x-(n_u-1)}$, where n_x is the dimension of the vector $\mathbf{x}$ and n_u is the dimension of the vector $\mathbf{u}$. If the rank of $\mathbf{G}^*_{\mathbf{x}}(\mathbf{x}_0)$ is $n_x - n_u + 1$ at a point $\mathbf{x}_0$ that satisfies Eq. 5.2, then a candidate trajectory in the boundary of the accessible output set is characterized as a set of points of the form $\mathbf{x} = \mathbf{x}(s)$ that satisfies Eq. 5.2, for all s in some neighborhood of $s = 0$. Such a trajectory can be selected with a unit tangent vector $\mathbf{x}'$ at $s = 0$ that satisfies

$$\mathbf{G}^*_{\mathbf{x}}(\mathbf{x}_0)\, \mathbf{x}' = \mathbf{0} \tag{5.3a}$$

$$\mathbf{x}'^T \mathbf{x}' = 1 \tag{5.3b}$$

In particular, the tangent $\mathbf{u}'$ to S in the output space satisfies

$$\mathbf{\Phi}_{\mathbf{u}}(\mathbf{q}_0)\, \mathbf{u}' + \mathbf{\Phi}_{\mathbf{z}}(\mathbf{q}_0)\, \mathbf{z}' = \mathbf{0} \tag{5.4}$$

which is obtained by taking the derivative of Eq. 5.1 with respect to s.

Multiplying both sides of Eq. 5.4 on the left by $\boldsymbol{\tau}_0^T$, the subcomponent of $\mathbf{x}_0$ satisfying Eq. 5.2, and using the second component of Eq. 5.2 yields

$$\boldsymbol{\tau}_0^T \mathbf{\Phi}_{\mathbf{u}}(\mathbf{q}_0)\, \mathbf{u}' = \mathbf{0} \tag{5.5}$$

Since this relationship holds for all tangents $\mathbf{u}'$ to S that pass through point $\mathbf{x}_0$, the normal $\mathbf{n}_0$ to S at $\mathbf{x}_0$ is

$$\mathbf{n}_0 \equiv \mathbf{n}\left(\mathbf{x}_0\right) = \mathbf{\Phi}_{\mathbf{u}}^T\left(\mathbf{q}_0\right)\boldsymbol{\tau}_0 \tag{5.6}$$

In order to obtain a basis for the tangent space to S at $\mathbf{x}_0$, a set of $n_u - 1$ orthonormal vectors that are orthogonal to $\mathbf{n}_0$ is defined by

$$\begin{aligned} \mathbf{u}_i'^T\mathbf{n}_0 = 0, \quad i = 1, \ldots, n_u - 1 \\ \mathbf{u}_i'^T\mathbf{u}_j' = \delta_{ij}, \quad i, j = 1, \ldots, n_u - 1 \end{aligned} \tag{5.7}$$

In order to investigate barriers to normal movement across S, a normal movement functional is defined as

$$\psi \equiv \mathbf{n}_0^T\left(\mathbf{u} - \mathbf{u}_0\right) \tag{5.8}$$

where $\mathbf{n}_0$ and $\mathbf{u}_0$ are fixed. Taking the time derivative of ψ and using Eq. 5.6 yields the normal output velocity,

$$\dot{\psi} = \mathbf{n}_0^T\dot{\mathbf{u}} \quad = \boldsymbol{\tau}_0^T\mathbf{\Phi}_{\mathbf{u}}\left(\mathbf{q}_0\right)\dot{\mathbf{u}} \tag{5.9}$$

Similarly, taking the time derivative of both sides of Eq. 5.1 yields

$$\mathbf{\Phi}_{\mathbf{u}}\left(\mathbf{q}_0\right)\dot{\mathbf{u}} + \mathbf{\Phi}_{\mathbf{z}}\left(\mathbf{q}_0\right)\dot{\mathbf{z}} = \mathbf{0} \tag{5.10}$$

which must hold for all kinematically admissible velocities $\dot{\mathbf{u}}$ and $\dot{\mathbf{z}}$, in particular for all finite input velocities. Multiplying both sides of Eq. 5.10 on the left by $\boldsymbol{\tau}_0^T$ yields

$$\boldsymbol{\tau}_0^T\mathbf{\Phi}_{\mathbf{u}}\left(\mathbf{q}_0\right)\dot{\mathbf{u}} = \mathbf{0} \tag{5.11}$$

Taken with Eq. 5.9, this shows that

$$\dot{\psi} = 0 \tag{5.12}$$

for all input velocities $\dot{\mathbf{v}}$.

This shows that in the configuration $\mathbf{q}_0^T \equiv \left[\mathbf{u}_0^T, \mathbf{w}_0^T, \mathbf{v}_0^T\right]^T$ satisfying Eq. 5.2, with $\mathbf{u}_0 \in S$, it is impossible to achieve a nonzero normal output velocity to S with a finite input velocity. Thus, an output trajectory at $\mathbf{u}_0$ in the configuration $\mathbf{q}_0$ of the manipulator that is specified with a nonzero normal component of velocity $\dot{\mathbf{u}}_0$ cannot be achieved with a finite input velocity $\dot{\mathbf{v}}_0$

5.3 Barriers to Output Movement Normal to S

In light of the foregoing, a second order expansion of the normal output functional ψ is required to obtain information regarding restrictions on movement across S. For notational convenience, the subvector of output and intermediate generalized coordinates $\mathbf{y} \equiv \left[\mathbf{u}^T, \mathbf{w}^T\right]^T$ is defined. Since these coordinates must be uniquely determined as functions of input coordinates for controllable manipulators, the governing constraint equations for the manipulator are presumed to satisfy the condition

$$|\mathbf{\Phi}_{\mathbf{y}}(\mathbf{q})| \neq 0 \tag{5.13}$$

The implicit function theorem [91] thus implies the existence of a three times continuously differentiable vector function $\mathbf{f}(\mathbf{v})$ such that

$$\mathbf{y} = \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \end{bmatrix} = \mathbf{f}(\mathbf{v}) \equiv \begin{bmatrix} \mathbf{f}^{\mathbf{u}}(\mathbf{v}) \\ \mathbf{f}^{\mathbf{w}}(\mathbf{v}) \end{bmatrix} \tag{5.14}$$

identically satisfies the constraint equations of Eq. 5.1.

Defining $\mathbf{n}_0^* = \left[\mathbf{n}_0^T,\ \mathbf{0}^T\right]^T$, the normal movement functional of Eq. 5.8 may be written in the form,

$$\psi = \mathbf{n}_0^T(\mathbf{u} - \mathbf{u}_0) \equiv \mathbf{n}_0^{*T}(\mathbf{y} - \mathbf{y}_0) \tag{5.15}$$

Since $\mathbf{u}$ and $\mathbf{y}$ are functions of $\mathbf{v}$, from Eq. 5.14,

$$\psi_{\mathbf{v}} = \mathbf{n}_0^{*T} \mathbf{y}_{\mathbf{v}} \tag{5.16}$$

Differentiating Eq. 5.1 with respect to $\mathbf{v}$ yields

$$\mathbf{\Phi}_{\mathbf{y}} \mathbf{y}_{\mathbf{v}} = -\mathbf{\Phi}_{\mathbf{v}} \tag{5.17}$$

which, as a result of Eq. 5.13, uniquely determines $\mathbf{y}_{\mathbf{v}}$.

Differentiating the i-th component of Eq. 5.16 with respect to v_j yields

$$\psi_{v_i v_j} = \mathbf{n}_0^{*T} \mathbf{y}_{v_i v_j} \tag{5.18}$$

Multiplying the i-th column of Eq. 5.17 on the left by a constant vector $\boldsymbol{\lambda}^T$ and differentiating with respect to v_j,

$$\boldsymbol{\lambda}^T \mathbf{\Phi}_{\mathbf{y}} \mathbf{y}_{v_i v_j} = -\left\{ \left[\left(\boldsymbol{\lambda}^T \mathbf{\Phi}_{\mathbf{y}} \right) \mathbf{y}_{v_i} \right]_{v_j} + \left(\boldsymbol{\lambda}^T \mathbf{\Phi}_{\mathbf{y}} \mathbf{y}_{v_i} \right)_{\mathbf{y}} \mathbf{y}_{v_j} + \boldsymbol{\lambda}^T \mathbf{\Phi}_{v_i v_j} + \boldsymbol{\lambda}^T \mathbf{\Phi}_{v_i \mathbf{y}} \mathbf{y}_{v_j} \right\} \tag{5.19}$$

where partial derivative of a product with respect to a variable implies only partial differentiation with respect to explicit dependence of the functions in the product on that variable. Setting

$$\mathbf{\Phi}_{\mathbf{y}}^T \boldsymbol{\lambda} = \mathbf{n}_0^* \tag{5.20}$$

uniquely determines $\boldsymbol{\lambda}$, due to Eq. 5.13. Equation 5.18 thus reduces to

$$\psi_{v_i v_j} = -\left\{ \left[\left(\boldsymbol{\lambda}^T \mathbf{\Phi} \right)_{\mathbf{y}} \mathbf{y}_{v_i} \right]_{v_j} + \left[\left(\boldsymbol{\lambda}^T \mathbf{\Phi} \right)_{\mathbf{y}} \mathbf{y}_{v_i} \right]_{\mathbf{y}} \mathbf{y}_{v_j} + \left(\boldsymbol{\lambda}^T \mathbf{\Phi} \right)_{v_i v_j} + \left(\boldsymbol{\lambda}^T \mathbf{\Phi} \right)_{v_i \mathbf{y}} \mathbf{y}_{v_j} \right\} \tag{5.21}$$

Note that Eq. 5.17 determines each vector $\mathbf{y}_{v_i}$ of first derivatives of $\mathbf{y}$ by factorization of $\mathbf{\Phi}_{\mathbf{y}}$ and repeated back substitution with the column $\mathbf{\Phi}_{v_i}$. In contrast, only a single solution of Eq. 5.20 for $\boldsymbol{\lambda}$ is required to determine all second derivatives of

$\mathbf{y}$ with respect to components of $\mathbf{v}$. This is a well known result of design sensitivity analysis [38].

Note that, from Eq. 5.2,

$$\mathbf{\Phi}_{\mathbf{z}}^{T}\boldsymbol{\tau} = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{w}}^{T} \\ \mathbf{\Phi}_{\mathbf{v}}^{T} \end{bmatrix} \boldsymbol{\tau} = 0 \tag{5.22}$$

In particular,

$$\mathbf{\Phi}_{\mathbf{w}}^{T}\boldsymbol{\tau} = 0 \tag{5.23}$$

From Eqs. 5.6, 5.20, and 5.23,

$$\mathbf{\Phi}_{\mathbf{y}}^{T}\boldsymbol{\lambda} = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{u}}^{T}\boldsymbol{\tau} \\ \mathbf{\Phi}_{\mathbf{w}}^{T}\boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{u}}^{T}\boldsymbol{\tau} \\ 0 \end{bmatrix} = \mathbf{\Phi}_{\mathbf{y}}^{T}\boldsymbol{\tau} \tag{5.24}$$

Due to Eq. 5.13, this implies that $\boldsymbol{\lambda} = \boldsymbol{\tau}$. Thus, no new equations need to be solved to determine the second derivatives of ψ in Eq. 5.21.

In a special case that often occurs in practice, the constraint function in Eq. 5.1 is of the form

$$\mathbf{\Phi}\left(\mathbf{y},\ \mathbf{v}\right) = \mathbf{\Psi}\left(\mathbf{y}\right) + \mathbf{\Lambda}\left(\mathbf{v}\right) \tag{5.25}$$

In this case, Eq. 5.21 reduces to

$$\psi_{v_i v_j} = -\left\{ \left[\left(\boldsymbol{\lambda}^{T}\mathbf{\Psi}\right)_{\mathbf{y}} \mathbf{y}_{v_i} \right]_{\mathbf{y}} \mathbf{y}_{v_j} + \left(\boldsymbol{\lambda}^{T}\mathbf{\Lambda}\right)_{v_i v_j} \right\} \tag{5.26}$$

Linearizing the constraint equations of Eq. 5.1, in terms of $\mathbf{y}$ and $\mathbf{v}$, yields $\mathbf{\Phi}_{\mathbf{y}}\left(\mathbf{q}_0\right)\delta\mathbf{y} + \mathbf{\Phi}_{\mathbf{v}}\left(\mathbf{q}_0\right)\delta\mathbf{v} = 0$. Due to the condition of Eq. 5.13, $\delta\mathbf{y}$ may be determined, as a linear function of $\delta\mathbf{v}$, as

$$\delta\mathbf{y} \equiv \begin{bmatrix} \delta\mathbf{u} \\ \delta\mathbf{w} \end{bmatrix} = -\mathbf{\Phi}_{\mathbf{y}}^{-1}\left(\mathbf{q}_0\right)\mathbf{\Phi}_{\mathbf{v}}\left(\mathbf{q}_0\right)\delta\mathbf{v} \equiv \begin{bmatrix} \mathbf{B} \\ \mathbf{C} \end{bmatrix} \delta\mathbf{v} \tag{5.27}$$

Of particular interest is the relationship between input and output variations, $\delta\mathbf{v}$ and

$\delta\mathbf{u}$; i.e.,

$$\delta\mathbf{u} = \mathbf{B}\delta\mathbf{v} \tag{5.28}$$

Interest centers on normal movement across S defined by Eq. 5.8, or equivalently Eq. 5.15, for variations $\delta\mathbf{v}^*$ that yield output variation $\delta\mathbf{u}^*$ normal to S. Requiring that $\delta\mathbf{u}^*$ be orthogonal to the basis of the tangent space, $\mathbf{u}'_i$, $i = 1, \ldots, n_u - 1$, of Eq. 5.7 and using Eq. 5.28 yields the condition

$$\mathbf{u}'^T_i \delta\mathbf{u}^* = \mathbf{u}'^T_i \mathbf{B}\delta\mathbf{v}^* = 0, \quad i = 1, \ldots, n_u - 1 \tag{5.29}$$

All variations $\delta\mathbf{v}^*$ satisfying Eq. 5.29 yield normal variations $\delta\mathbf{u}^*$ to S. Defining the matrix

$$\mathbf{U}' \equiv \left[\mathbf{u}'_1, \ldots, \mathbf{u}'_{n_u-1}\right] \tag{5.30}$$

the orthogonality conditions of Eq. 5.29 for admissible $\delta\mathbf{v}^*$ can be stated in terms of a basis $\{\mathbf{s}_i\}$ of the null space of the matrix $\mathbf{U}'^T\mathbf{B}$; i.e., $\mathbf{U}'^T\mathbf{B}\mathbf{s}_i = 0$,

$$i = 1, \ldots, n_v - n_u + 1 \equiv n_s \tag{5.31}$$

Thus, all vectors $\delta\mathbf{v}^*$ satisfying Eq. 5.29 may be written in terms of this basis as

$$\delta\mathbf{v}^* = \sum_{i=1}^{n_s} \alpha_i \mathbf{s}_i = [\mathbf{s}_1, \ldots, \mathbf{s}_{n_s}] \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_{n_s} \end{bmatrix} \equiv \mathbf{S}\boldsymbol{\alpha} \tag{5.32}$$

where a defines a parameterization of the subspace of input variations satisfying Eq. 5.29.

Since first order terms in the Taylor expansion of $\mathbf{y}$ in terms of variations $\delta\mathbf{v}^*$ are zero, the second order Taylor expansion of $\mathbf{y}$ is

$$\Delta\psi = \frac{1}{2}\delta\mathbf{v}^{*T}\psi_{\mathbf{vv}}\delta\mathbf{v}^* \tag{5.33}$$

where the Hessian matrix $\psi_{\mathbf{vv}}$ is

$$\psi_{\mathbf{vv}} = \left[\psi_{v_i v_j}\right] \tag{5.34}$$

Substituting from Eq. 5.32 into Eq. 5.33 yields

$$\Delta\psi = \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{S}^T \psi_{\mathbf{vv}} \mathbf{S}\boldsymbol{\alpha} \tag{5.35}$$

Since arbitrary values of the vector a yield normal output variations across S, definiteness properties of the quadratic form $\mathbf{S}^T \psi_{\mathbf{vv}} \mathbf{S}$ of Eq. 5.35 determine whether it is possible to control output normal to S. If this quadratic form is positive definite; i.e., if all its eigenvalues are positive [37], then movement only in the positive sense of the normal vector $\mathbf{n}_0$ of Eq. 5.6 is possible. Conversely, if the quadratic form is negative definite; i.e., if all its eigenvalues are negative, only normal movement opposite to the sense of $\mathbf{n}_0$ is possible. If the quadratic form has both positive and negative eigenvalues, then it is possible to control output in both directions normal to S. Finally, if the quadratic form has one or more zero eigenvalues, then a higher order Taylor expansion of $\mathbf{y}$ in the eigenspace of the matrix corresponding to zero eigenvalues of the quadratic form is required to draw conclusions regarding restrictions on controlled output across S.

It should be noted that even in cases for which the quadratic form of Eq. 5.35 has both positive and negative eigenvalues, output control difficulties across segments of S should be expected. In particular, the magnitude of tangential variations in output along S will be of first order in input variations $\delta\mathbf{v}$. In contrast, output variations normal to S will be of second order in $\delta\mathbf{v}$. As observed by Nielsen, deWit, and Hagander [73] and shown analytically in Section 2, this behavior precludes certain regimes of velocity control and will almost certainly lead to a requirement for very large actuator force or torque inputs to control motion that is required to cross segments of S, when the manipulator is in the associated singular configuration. Nevertheless, the foregoing criteria make a distinction between segments of S that are barriers to output position control, as opposed to segments that only yield control difficulties.

5.4 Examples

The criteria for barrier analysis presented in Sections 2 and 3 have been implemented in an experimental computer program that uses automatic differentiation to evaluate derivatives needed [27, 40]. Two planar examples are analyzed in this section to illustrate use of the method. A spatial Stewart platform is analyzed in detail in a companion paper [62].

In the figures presented here, numbered circles are bifurcation points that identify distinct segments of S that they separate. Solid curves are segments of S across which output is restricted when the manipulator is in the associated critical configuration. Dotted curves are segments of S, interior to A, that can be crossed. Arrows indicate normal directions in which the quadratic form of Eq. 5.35 is negative definite. Therefore the manipulator can control output only opposite to the directions indicated; i.e., the arrows represent directions of restricted output.

5.4.1 Planar Manipulator with Redundant Input

The planar manipulator shown in Fig. 5.1 [40] is comprised of three bodies. The angles θ_1, θ_2, and θ_3 are control inputs and the x and y coordinates of point P are output coordinates; i.e., $\mathbf{u} = [x, y]^T$. No intermediate coordinates are required in description of this manipulator. Unilateral constraints on the input angles for this manipulator are $-\frac{\pi}{3} = \theta_i^{\min} \le \theta_i \le \theta_i^{\max} = \frac{\pi}{3}$, $i = 1, 2, 3$. Defining the input angles θ_i

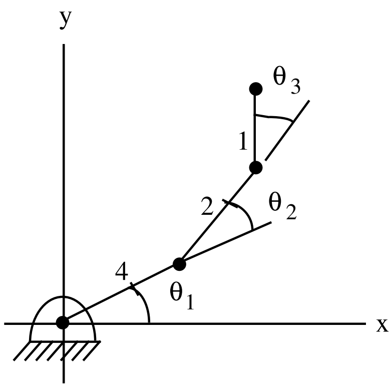

Figure 5.1: Planar Manipulator with Redundant Input

as functions of new input coordinates v_i by the relations $\theta_i = \frac{\pi}{3}\sin v_i$, $i = 1, 2, 3$, the unilateral constraints are automatically satisfied. In this case, $\mathbf{z} = \mathbf{v} = [v_1, v_2, v_3]^T$, and the constraint equations are

$$\mathbf{\Phi}(\mathbf{u}, \mathbf{z}) = \begin{bmatrix} x - 4\cos\theta_1 - 2\cos(\theta_1+\theta_2) - \cos(\theta_1+\theta_2+\theta_3) \\ y - 4\sin\theta_1 - 2\sin(\theta_1+\theta_2) - \sin(\theta_1+\theta_2+\theta_3) \end{bmatrix} = \mathbf{0} \tag{5.36}$$

where θ_i is a function of v_i. Note that the constraint function $\mathbf{\Phi}(\mathbf{u}, \mathbf{z})$ is of the special form given in Eq. 5.25.

The sub-Jacobian of the constraint function of Eq. 5.36 with respect to $\mathbf{v}$ is the 2×3 matrix

$$\mathbf{\Phi_v} = \frac{\pi}{3}\begin{bmatrix} \{4\mathrm{s}(\theta_1) + 2\mathrm{s}(\theta_1+\theta_2) + \mathrm{s}(\phi)\}\,\mathrm{c}(v_1) & \{2\mathrm{s}(\theta_1+\theta_2) + \mathrm{s}(\phi)\}\,\mathrm{c}(v_2) & \mathrm{s}(\phi)\mathrm{c}(v_3) \\ \{-4\mathrm{c}\theta_1 + 2\mathrm{c}(\theta_1+\theta_2) + \mathrm{c}(\phi)\}\,\mathrm{c}(v_1) & \{-2\mathrm{c}(\theta_1+\theta_2) + \mathrm{c}(\phi)\}\,\mathrm{c}(v_2) & -\mathrm{c}(\phi)\mathrm{c}(v_3) \end{bmatrix} \tag{5.37}$$

where $\phi = \theta_1 + \theta_2 + \theta_3$, $\mathrm{s}(\bullet) = \sin(\bullet)$ and $\mathrm{c}(\bullet) = \cos(\bullet)$ Thus, the 3×3 matrix $\psi_{\mathbf{vv}}$ of Eq. 5.26 is computed as $-\left(\boldsymbol{\lambda}^T\mathbf{\Phi}\right)_{v_i v_j}$ component-wise, or as $-\left(\boldsymbol{\lambda}^T\mathbf{\Phi_v}\right)_{v_j}$ column-wise, where $\boldsymbol{\lambda} = \boldsymbol{\tau}$ is a byproduct of the solution of Eq. 5.2 in the continuation process along S [40].

Segments comprising S, computed by Haug, Luh, Adkins, and Wang [40], for

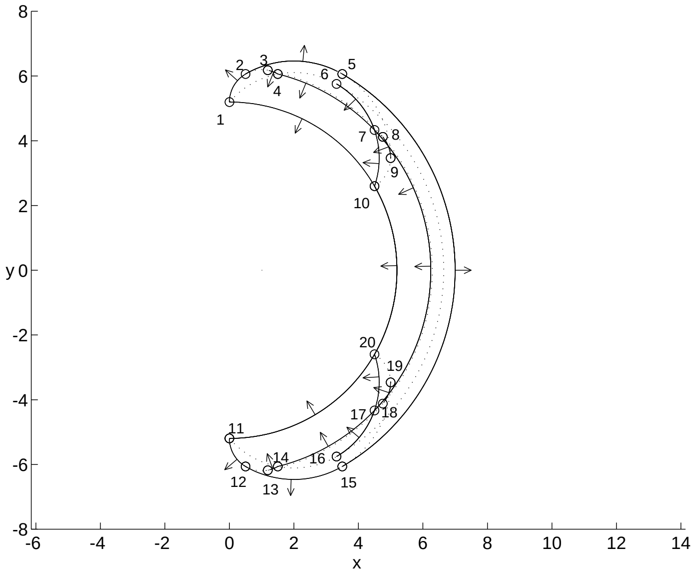

Figure 5.2: S for Planar Manipulator

this manipulator are shown in Fig. 5.2. All curves in Fig. 5.2 represent constraints on nonzero normal output velocity control when the manipulator is in the associated singular configuration. As shown in Fig. 5.2, all segments of the outer boundary of A are absolute restrictions to motion and several curves in S that are interior to A are absolute restrictions to motion when the manipulator is in the associated singular configuration. In each of these configurations the manipulator has two inputs θ_i at either an upper or lower limit. These limiting angles provide restriction to control of motion, since changing the input value a small amount from either a maximum or minimum value will cause motion in a direction opposite to the normal to the curve at such a point, as shown in Fig. 5.3.

An example is the configuration of the mechanism on the curve between points 7 and 10 in Fig. 5.3, with generalized coordinates

$$\mathbf{q} = [4.6402,\ 3.2926,\ 1.5708,\ -0.8383,\ 1.5708]^T$$

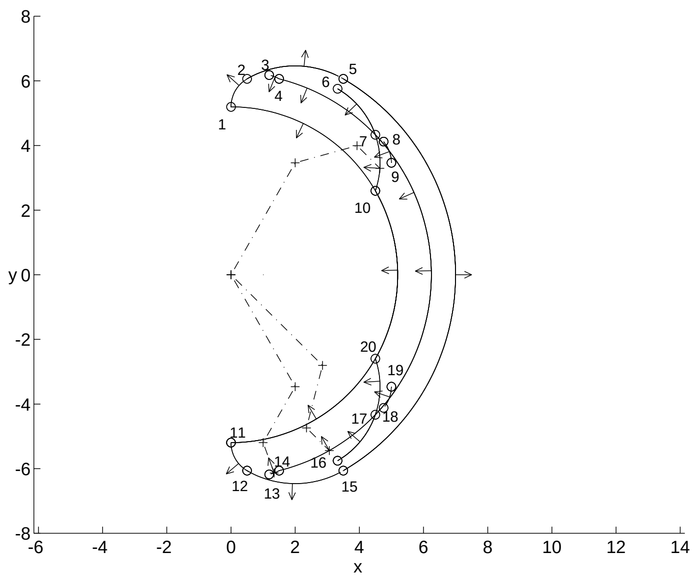

Figure 5.3: Planar Manipulator Output Restrictions and Several Associated Singular Configurations

This is a point at which $v_1 = v_3 = \frac{\pi}{2}$, so θ_1 and θ_3 are at their upper limits. Motion along the curve is caused by changing v_2. From Fig. 5.3, it can be seen that decreasing θ_1 or θ_3 will result in the working point of the manipulator moving to the right. This motion argument can be made on each of the solid curves interior to A in Fig. 5.2.

5.4.2 Planar Stewart Platform

Recall the planar Stewart platform of Section 1.4.1. The generalized coordinate vector is $\mathbf{q} = \left[\mathbf{u}^T, \mathbf{w}^T, \mathbf{v}^T\right]^T = [x,\ y,\ \phi,\ v_1,\ v_2,\ v_3]^T$ and the constraint equations

$$\boldsymbol{\Phi}\left(\mathbf{u}, \mathbf{z}\right) \equiv \begin{bmatrix} \ell_1^2 - (x - \cos\phi + 1)^2 - (y - \sin\phi)^2 \\ \ell_2^2 - (1 - x + \cos\phi)^2 - (y - \sin\phi)^2 \\ \ell_3^2 - (x + \cos\phi - 2)^2 - (y + \sin\phi)^2 \end{bmatrix} = \mathbf{0} \tag{5.38}$$

are of the special form of Eq. 5.26. Using the functions $\ell_i(v_i)$, $i = 1, 2, 3$ from Eq. 1.21 the sub-Jacobian matrix of the constraints of Eq. 5.38 with respect to $\mathbf{v} = [v_1, v_2, v_3]^T$ is

$$\mathbf{\Phi_v} = \begin{bmatrix} 0.5858\ell_1 \cos v_1 & 0 & 0 \\ 0 & 0.5858\ell_2 \cos v_2 & 0 \\ 0 & 0 & 0.7320\ell_3 \cos v_3 \end{bmatrix} \tag{5.39}$$

and, with respect to $\mathbf{y} = [x,\ y,\ \phi]^T$, it is

$$\mathbf{\Phi_y} = \begin{bmatrix} -2(x - \cos\phi + 1) & -2(y - \sin\phi) & -2\,(x+1)\sin\phi + 2y\cos\phi \\ +2(1 - x + \cos\phi) & -2(y - \sin\phi) & 2\,(1-x)\sin\phi + 2y\cos\phi \\ -2(x - 2 + \cos\phi) & -2(y + \sin\phi) & 2\,(x-2)\sin\phi - 2y\cos\phi \end{bmatrix}$$

The elements of the Hessian of ψ are thus determined by Eq. 5.26 and these sub-Jacobians.

The actuator limits given above are set so that colinearity of the upper platform and any actuator cannot occur. This yields the curves comprising S, determined by Haug, Luh, Adkins, and Wang [40], shown in Fig. 5.4. With these actuator limits, the analysis method of Section 3 shows that interior curves interior to A are not absolute restrictions on motion even though they are restrictions on nonzero normal output velocity. Thus, the only curves in S that cannot be crossed comprise the boundary of A, as shown in Fig. 5.4.

Modifying the bounds on length of actuator 3 to $\ell_3^{\min} = 1 \leq \ell_3 \leq 3 = \ell_3^{\max}$ allows colinearity of actuator 1 and the upper platform to occur. This colinearity is known to be a singular configuration [40, 41]. For this modified planar Stewart platform, some interior curves in S shown in Fig. 5.5 are absolute restrictions on output control when the manipulator is in the associated singular configuration. The bifurcation points shown in Fig. 5.5 have been labeled for ease of correlation with Fig. 5.4. On the interior curves across which output motion is restricted, the manipulator is in a singular configuration at which the first actuator has minimum length. This is the physical reason why these curves represent absolute restrictions on motion. Increasing the length of actuator 1 moves the working point in one direction, but since its length

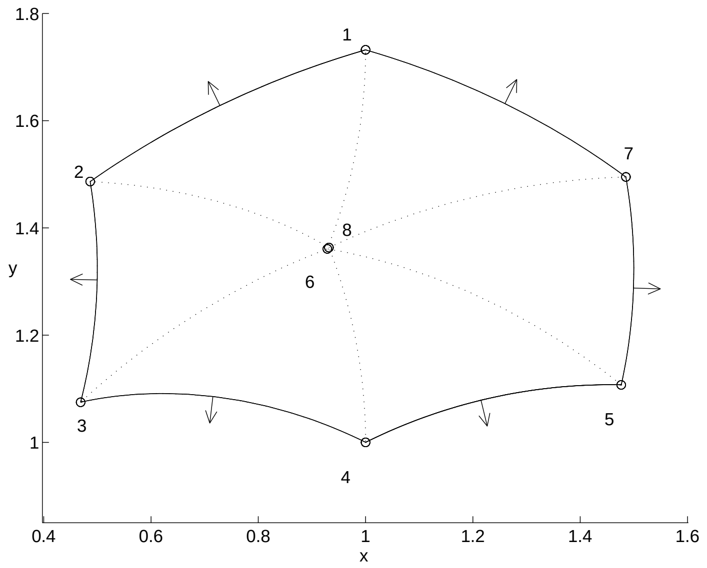

Figure 5.4: S for Basic Planar Stewart Platform

cannot be decreased, motion on only one side of the candidate segment can occur. Figure 5.6 illustrates this situation.

In addition to these results, along segment 7' to 7" in Fig. 5.5, the quadratic form $\mathbf{S}^T \psi_{\mathbf{vv}} \mathbf{S}$ of Eq. 5.35 has a negative eigenvalue and a zero eigenvalue. Whenever a segment of S changes from a curve that can be crossed to a curve with restricted motion, the quadratic form of Eq. 5.35 must, at the intersection, have a zero eigenvalue. However, the quadratic form has a zero eigenvalue on the entire segment 7' to 7". This has an implication on analysis of the character of this curve. Since this curve is associated with a zero eigenvalue, there is a direction (defined by an eigenvector of $\mathbf{S}^T \psi_{\mathbf{vv}} \mathbf{S}$) in which no information on the sign of $\Delta\psi$ in Eq. 5.35 is given. For this direction, higher order derivatives of ψ with respect to $\mathbf{v}$ would be required to analytically assess the restriction on motion.

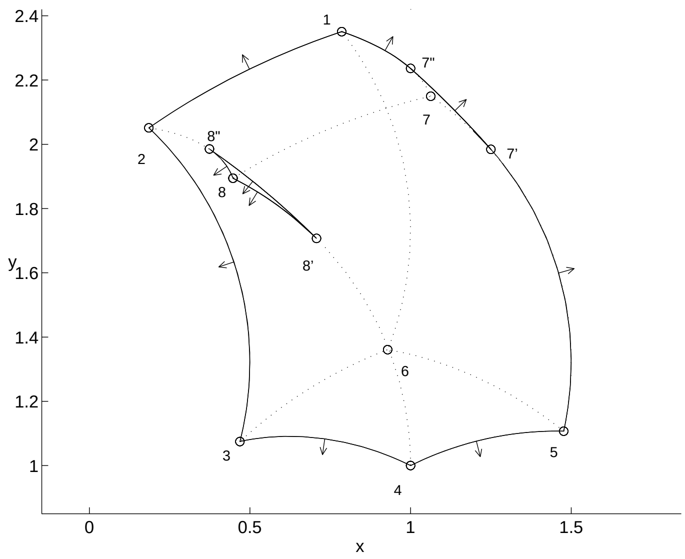

Figure 5.5: S for Modified Planar Stewart Platform

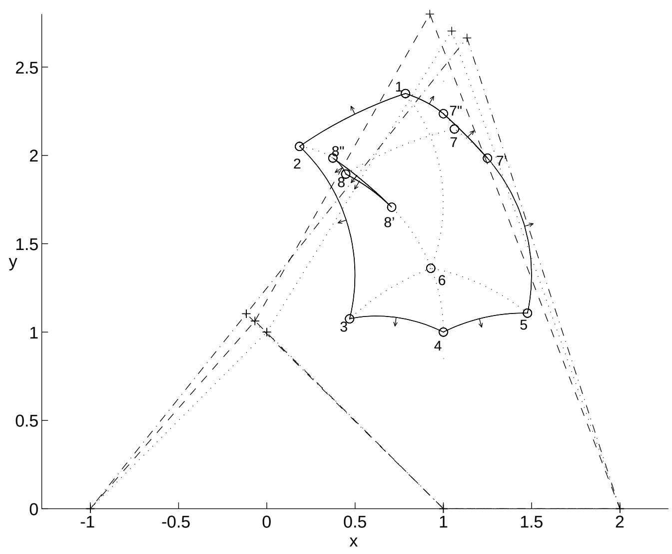

Figure 5.6: S Modified Planar Stewart Platform with Typical Configurations for Curves 8-8' (Dotted), 8'8" (Dash-Dot), and 8"-8 (Dashed)

CHAPTER 6
DOMAINS OF OPERATION AND INTERFERENCE

Domains associated with one or more working bodies of a mechanism or manipulator are defined that characterize the range of operation of the bodies and interference that may occur between the bodies. The equations defining domains are derived in terms of equations defining the kinematics of the mechanism and the geometry of the working bodies. The criteria presented are applied to both planar and spatial mechanism and manipulator examples using numerical methods from previous chapters.

6.1 Analytical Conditions For Working Domains And Boundaries

Maintaining consistency with work done in this area, the notation differs slightly from that of Section 1.2. In order to define working domains in bodies that move with an underlying mechanism, whose configuration is defined by a generalized coordinate $\mathbf{q}$, the shape of the domain of the working bodies is parameterized by a vector $\boldsymbol{\mu}$. Defining the n_z-vector $\mathbf{z} = \left[\mathbf{q}^T, \boldsymbol{\mu}^T\right]^T$, the n_u-vector $\mathbf{u}$ to a working point on a working body is given analytically in the form $\mathbf{u} = \mathbf{g}\left(\mathbf{z}\right)$, where the vector function $\mathbf{g}(\mathbf{z})$ is twice continuously differentiable. The kinematic constraint equations for the mechanism and equations involving parameterization of the shapes of working bodies, which may account for inequalities with slack variables [49] to define domains, are written in the form $\boldsymbol{\Phi}\left(\mathbf{z}\right) = \mathbf{0}$, where the m-vector of functions $\boldsymbol{\Phi}\left(\mathbf{z}\right)$ is twice continuously differentiable. The system of $n_u + m$ equations that define points in a domain specified by the associated criteria is

$$\boldsymbol{\Psi}(\mathbf{u}, \mathbf{z}) \equiv \begin{bmatrix} \mathbf{u} - \mathbf{g}(\mathbf{z}) \\ \boldsymbol{\Phi}(\mathbf{z}) \end{bmatrix} = \mathbf{0} \tag{6.1}$$

Thus, the domain of interest is the set

$$D = \{\mathbf{u} \in \Re^{n_u} : \boldsymbol{\Psi}(\mathbf{u}, \mathbf{z}) = \mathbf{0},\ for\ some\ \mathbf{z}\} \tag{6.2}$$

6.2 Operational Envelopes

A point P in the domain of the working body is located in space by the vector $\mathbf{u} = [x,\ y,\ z]^T = \mathbf{r}(\mathbf{q}) + \mathbf{A}(\mathbf{q})\mathbf{s}'^P$. In the present case, however, it is intended that point P is free to vary over the domain of the working body. A typical point P in the domain of the working body can be defined, relative to the body-fixed x'-y'-z' reference frame, as a function of a parameter vector $\boldsymbol{\mu}$, $\mathbf{s}'^P = \mathbf{f}(\boldsymbol{\mu})$. The vector $\boldsymbol{\mu}$ may be comprised of independent parameters, or it may be required to satisfy an equation of the form $\mathbf{B}(\boldsymbol{\mu}) = \mathbf{0}$. This equation is combined with purely kinematic constraint equations in $\mathbf{q}$, to form the composite set of constraint equations $\boldsymbol{\Phi}(\mathbf{q},\ \boldsymbol{\mu}) \equiv \boldsymbol{\Phi}(\mathbf{z}) = \mathbf{0}$. The operational envelope, denoted OE, of the working body is defined as the set of all points in space that can be occupied by some point P on the working body, for some admissible value of the generalized coordinate vector $\mathbf{q}$. Conditions for points in space that are occupied by points in the domain of the working body, as it moves under the influence of kinematic constraints defining the underlying mechanism, are

$$\boldsymbol{\Psi}(\mathbf{u},\ \mathbf{z}) \equiv \begin{bmatrix} \mathbf{u} - \mathbf{r}(\mathbf{q}) - \mathbf{A}(\mathbf{q})\mathbf{f}(\boldsymbol{\mu}) \\ \boldsymbol{\Phi}(\mathbf{q},\ \boldsymbol{\mu}) \end{bmatrix} = \mathbf{0} \tag{6.3}$$

where the convention that $\mathbf{z} = \left[\mathbf{q}^T,\ \boldsymbol{\mu}^T\right]^T$ has been used. Thus, the operational envelope of the working body is $OE = \{\mathbf{u} : \boldsymbol{\Psi}(\mathbf{u},\ \mathbf{z}) = \mathbf{0}\ for\ some\ \mathbf{z}\}$.

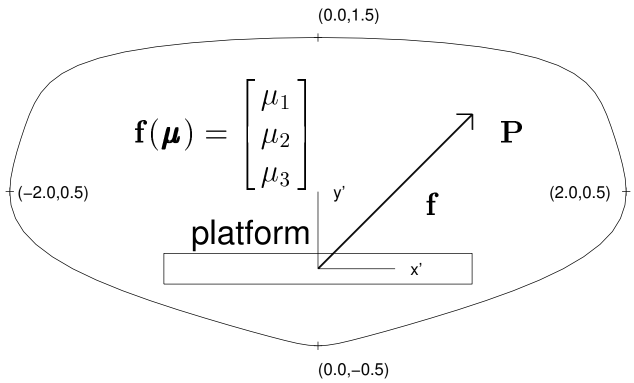

Figure 6.1: Projection Dome

6.2.1 Interference of a Planar Stewart Platform with Dome and Room

Consider the operational envelope of the projection dome of Fig. 6.1 that is attached to the planar Stewart platform of Fig. 1.3 to form a driving simulator. The generalized coordinate vector is $\mathbf{q} = [x,\ y,\ v_1,\ v_2,\ v_3,\ \phi]^T$ and the constraint equations for the manipulator are given in Eq. 1.22. The equation for the boundary of the dome is $\mu_1^2+(2+1.3(\mu_2-0.5)^3)^2((\mu_2-0.5)^2-1) = 0$. The domain of the dome is thus characterized by the inequality $\mu_1^2+(2+1.3(\mu_2-0.5)^3)^2((\mu_2-0.5)^2-1) \leq 0$. To represent the domain of the dome in equation form, a slack variable μ_3 is introduced and the equation for the domain is

$$B\left(\boldsymbol{\mu}\right) = \mu_1^2 + (2 + 1.3(\mu_2 - 0.5)^3)^2((\mu_2 - 0.5)^2 - 1) + \mu_3^2 = 0 \tag{6.4}$$

A typical point in the domain of the dome is thus given, in the platform body-fixed x'-y' frame, as $\mathbf{s}'^P = [\mu_1,\ \mu_2]^T \equiv \mathbf{f}\left(\boldsymbol{\mu}\right)$. The equation defining the operational envelope of the domain of the dome of the planar Stewart platform is

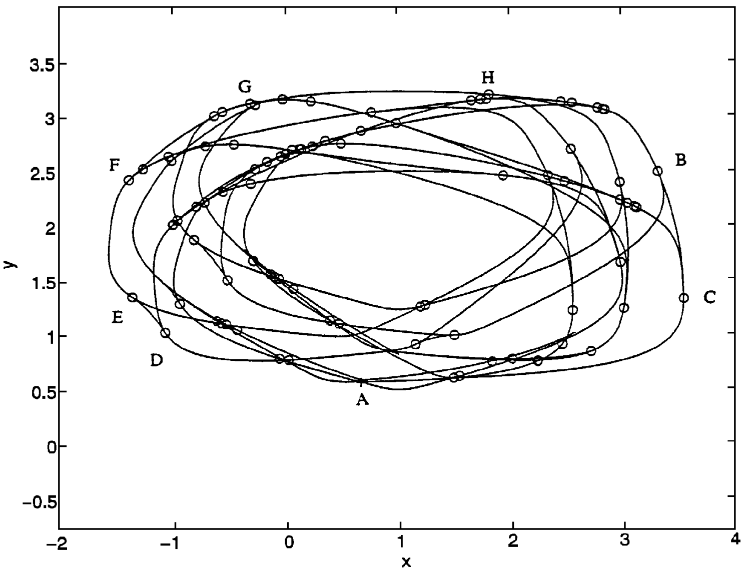

Figure 6.2: Singular Boundary Curves

$$\Psi(\mathbf{u},\mathbf{q},\boldsymbol{\mu}) \equiv \begin{bmatrix} u_1 - x - \mu_1\cos\phi + \mu_2\sin\phi \\ u_2 - y - \mu_1\sin\phi - \mu_2\cos\phi \\ \mu_1^2 + (2 + 1.3(\mu_2 - 0.5)^3)^2((\mu_2 - 0.5)^2 - 1) + \mu_3^2 \\ \ell_1^2 - (x - \cos\phi + 1)^2 - (y - \sin\phi)^2 \\ \ell_2^2 - (1 - x + \cos\phi)^2 - (y - \sin\phi)^2 \\ \ell_3^2 - (x + \cos\phi - 2)^2 - (y + \sin\phi)^2 \end{bmatrix} = \mathbf{0} \qquad (6.5)$$

To obtain a point interior to OE, the mechanism legs were set to their maximum length, which determines the values of x, y and ϕ. The values $\mu_1 = 0.0$ and $\mu_2 = 0.5$, and $\mu_3 = 2.0$ were chosen, and the search direction was selected as $\mathbf{c} = [0,1]^T$. Searching in this direction, from the configuration

$$\left[\mathbf{u}^{0^T},\ \mathbf{q}^{0^T},\ \boldsymbol{\mu}^{0^T}\right]^T = \left[1.0,\ \sqrt{3},\ 1.0,\ \sqrt{3},\ \tfrac{\pi}{2},\ \tfrac{\pi}{2},\ \tfrac{\pi}{2},\ 0,\ 0,\ 0.5,\ 2\right]^T$$

using the method of Section 1.3, yielded the following point on OE:

$$\mathbf{x}^* = \left[\mathbf{u}^{*T}, \mathbf{q}^{*T}, \boldsymbol{\mu}^{*T}\right]^T = [1.0, 3.232, 1.0, 1.732, 1.571, 1.571, 1.571, 0., 0., 1.5, 0.]^T$$

Starting from $\mathbf{x}^*$, the boundary is traced using the method outlined in Section 1.3. The resulting set of boundary segments is shown in Fig. 6.2. The outer curves define the boundary of the operational envelope, OE. In Fig. 6.2, several points of interest are marked. There are points at which the projections of boundary curves cross, but the curves computed do not intersect in the higher dimensional space. An example of such points is point A. Points such as these pose a problem when attempting to calculate the boundary of the operational envelope without also calculating the interior singular curves. Many segments on the boundary of the operational envelope arise from tracing the shape of the dome with the platform in a fixed configuration. Examples of this behavior are segments EF and GH. There are, however boundary segments that result from moving the platform and the dome, sweeping out a curve from one configuration of the platform to another. Examples of such segments are BC and DE. On segment BC the mechanism is moving by changing the third leg from its maximum to its minimum length, hence v_3 changes from $\pi/2$ to $-\pi/2$. On segment DE the mechanism is moving by changing the second leg from its minimum to its maximum length, hence v_2 changes from $-\pi/2$ to $\pi/2$.

6.2.2 Spatial Stewart Platform

Attached to the six degree of freedom motion base is a projection dome for wide-field visual display. This projection dome is modelled as a body of revolution about the vertical axis of the cross section shape, Fig. 6.1, used in the previous planar example. Actual data from the Iowa Driving Simulator's Stewart Platform motion base [62] are used in the computations described below.

Prescribing a typical point P in the dome, shown in Fig. 6.1, as a function of a vector $\boldsymbol{\mu} = [\mu_1, \mu_2, \mu_3, \mu_4]^T$

$$\mathbf{s}'^P = \mathbf{f}(\boldsymbol{\mu}) \equiv [\mu_1, \mu_2, \mu_3]^T \tag{6.6}$$

which is subject to an equation that defines the domain of the projection dome,

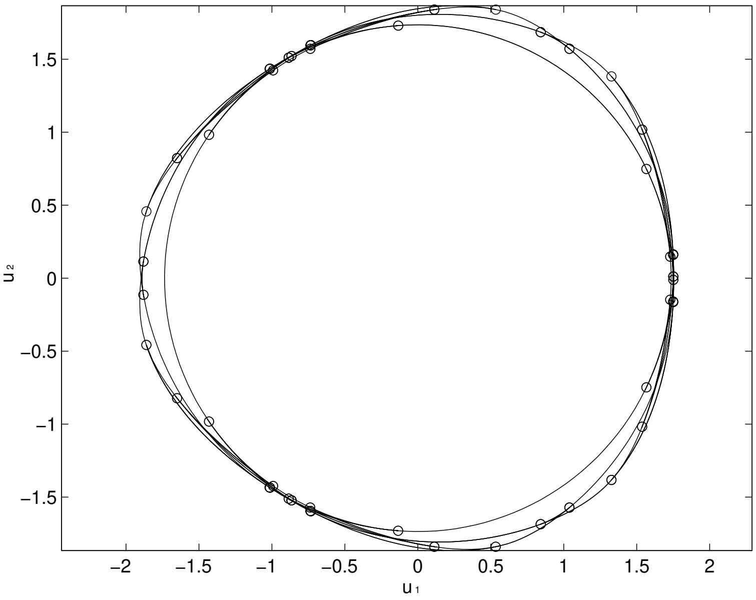

Figure 6.3: Singular Curves at $u_3 = 4.7$ Meters

$$\mathbf{B}(\boldsymbol{\mu}) = \mu_1^2 + \mu_2^2 + (2 + 1.3(\mu_3 - 0.5)^3)^2((\mu_3 - 0.5)^2 - 1) + \mu_4^2 \tag{6.7}$$

where μ_1, μ_2, and μ_3 are spatial coordinates in the dome and μ_4 is a slack variable. The vector $\mathbf{u} = [u_1, u_2, u_3]^T$ of output coordinate from the origin of the global reference frame to point P in the projection dome is

$$\mathbf{u} = \mathbf{r}(\mathbf{q}) + \mathbf{A}(\alpha, \boldsymbol{\beta})\mathbf{s}'^P \tag{6.8}$$

Since ∂OE is a surface in $\Re^3$, horizontal cutting planes $u_3 - b = 0$ are added to define a family of one dimensional generators for ∂OE. Due to the necessary, but not sufficient, conditions used in the analytical criteria for determining ∂OE, the computed candidate curves for ∂OE contain some interior segments. These sets are shown in Figs. 6.3, 6.4, and 6.5 for three different cutting planes.

The points marked by circles are bifurcation points where curves branch in the **x**-space. As seen in Fig. 6.3 some of the outer boundary segments cross in the projected

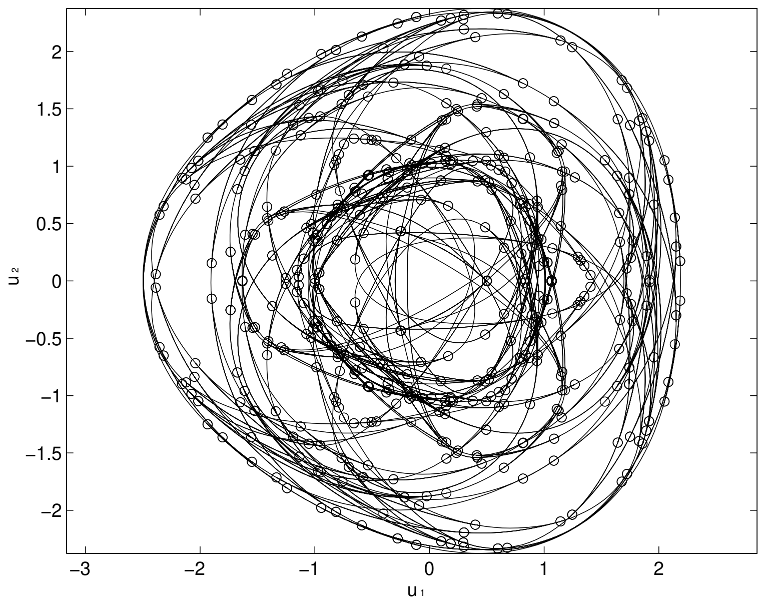

Figure 6.4: Singular Curves at $u_3 = 4.4$ Meters

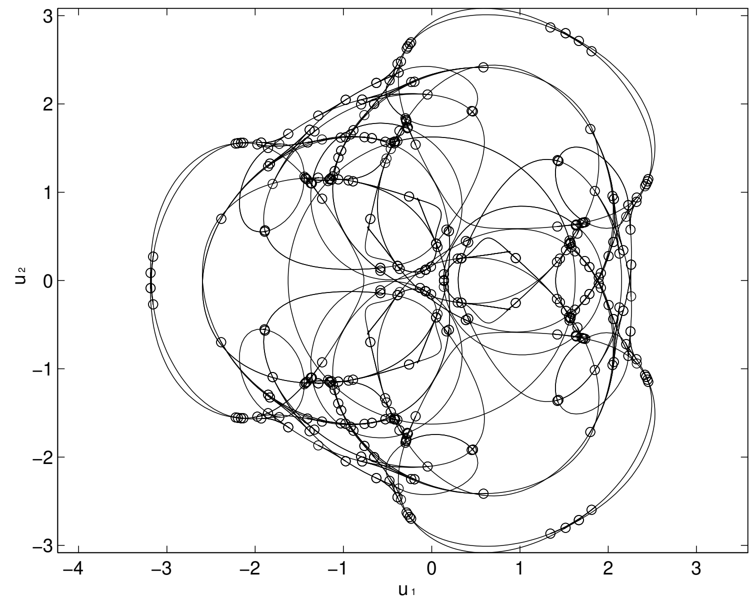

Figure 6.5: Singular Curves at $u_3 = 1.7$ Meters

output space at points that are not bifurcation points in the higher dimensional space. This causes difficulty in any attempt to only compute outer boundary curves. Thus, all segments satisfying Eq. 1.4 were computed.

Due to the complexity of curves in each slice, significant computer time is required. The computation process also requires many computations of first and second derivatives of the functions appearing in Eq. 1.4. This was initially implemented in an environment using the automatic differentiation package ADOL-C [27]. It was found, for the slice $u_3 = 4.7$, in Fig. 6.3 that about 69% of the compute time was used for calculating first derivatives of Eq. 1.4. Due to the substantial amount of time required to compute this simple slice, a faster method for computing these first derivatives was sought. Access to a preliminary version of ADIC [10], a new automatic differentiation software tool that produces compiled first derivatives, was obtained from Argonne National Laboratory. Excellent results were achieved using ADIC, combined in an experimental code that still uses ADOL-C for the necessary second order derivatives. The first derivatives provided by ADIC were about 295 times faster than those from ADOL-C. The percentage of time spent on first derivative calculation for the slice $u_3 = 4.7$ went from 69.13% to 0.76% of total compute time, making computations of the more complicated slices feasible. in an experimental code that still uses ADOL-C for the necessary second order derivatives. The first derivatives provided by ADIC were about 295 times faster than those from ADOL-C. The percentage of time spent on first derivative calculation for the slice $u_3 = 4.7$ went from 69.13% to 0.76% of total compute time, making computations of the more complicated slices feasible.

The outer boundary was extracted from each slice and collected to form the operational envelope boundary surface shown in Fig. 6.6.

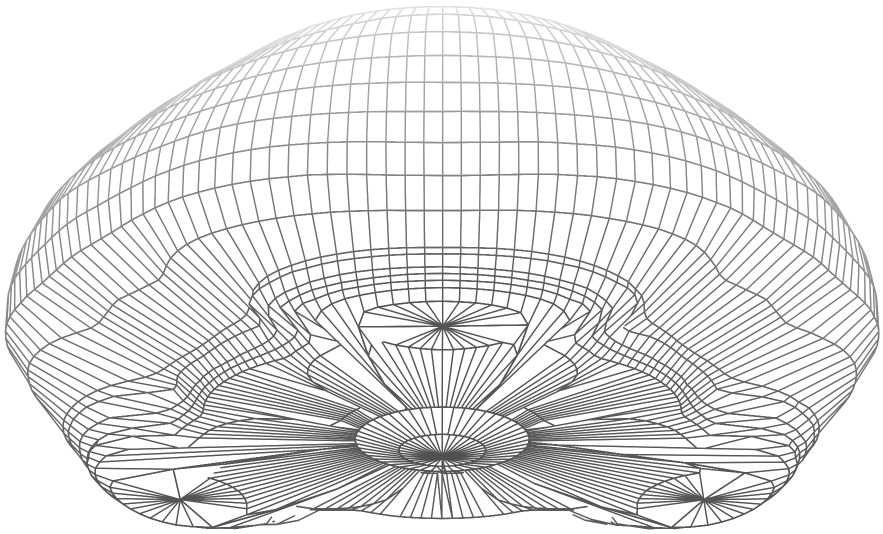

Figure 6.6: ∂OE for IDS Stewart Platform and Dome

6.3 Domains Of Interference

Consider an underlying mechanism that consists of a number of kinematically constrained moving bodies, with a vector of generalized coordinates that are subjected to kinematic constraint equations. Interference consideration is focused on a pair of working bodies of the underlying mechanism, denoted bodies 1 and 2. The origins of body-fixed x'-y'-z' reference frames that are used to characterize the position of each of the bodies are defined by vectors $\mathbf{r}_1(\mathbf{q})$ and $\mathbf{r}_2(\mathbf{q})$, which are functions of the generalized coordinates of the underlying mechanism. Orthogonal orientation transformation matrices $\mathbf{A}_1(\mathbf{q})$ and $\mathbf{A}_2(\mathbf{q})$, which are functions of the mechanism generalized coordinates, define the orientation of the body-fixed reference frames on bodies 1 and 2, respectively, relative to the global x-y-z frame. Points P in the domains of bodies 1 and 2 are characterized in terms of parameter vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, in their respective body-fixed reference frames, as $\mathbf{s}'^P_1 = \mathbf{f}_1(\boldsymbol{\mu}_1)$ and $\mathbf{s}'^P_2 = \mathbf{f}_2(\boldsymbol{\mu}_2)$. Any

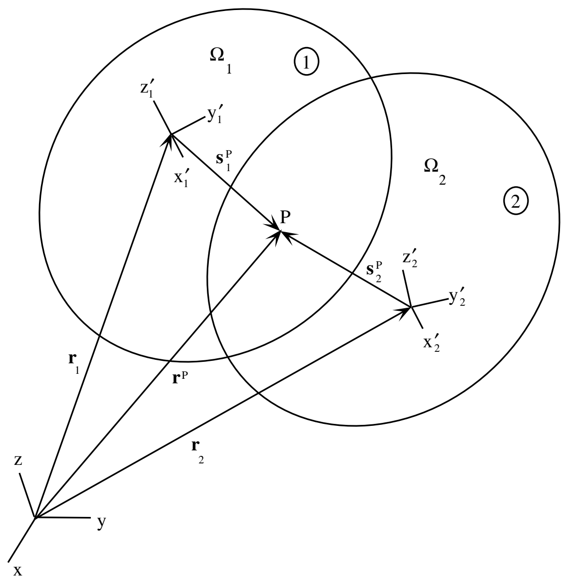

Figure 6.7: Interference of a Pair of Working Bodies

equations involving $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are combined with the kinematic constraint equations. Denoting the combined set of geometry parameterization vectors as $\boldsymbol{\mu} = \left[\boldsymbol{\mu}_1^T, \boldsymbol{\mu}_2^T\right]^T$, kinematic and geometric constraint equations are of the form $\boldsymbol{\Phi}(\mathbf{q}, \boldsymbol{\mu}) = \mathbf{0}$. The condition that some point P in space is occupied by a material point in each of bodies 1 and 2 is formulated as $\mathbf{r}_2(\mathbf{q}) + \mathbf{A}_2(\mathbf{q})\,\mathbf{f}_2(\boldsymbol{\mu}_2) - \mathbf{r}_1(\mathbf{q}) - \mathbf{A}_1(\mathbf{q})\,\mathbf{f}_1(\boldsymbol{\mu}_1) = \mathbf{0}$. This formulation is depicted in Fig. 6.7. The domain of points in body 1 that interfere with body 2 is denoted $I_1'D_2$. Let $\mathbf{u}_1'$ be a point on body 1, characterized in its body-fixed reference frame; i.e., $\mathbf{u}_1' = \mathbf{f}_1(\boldsymbol{\mu}_1)$. If $\mathbf{u}_1'$ is in $I_1'D_2$, then there exist $\mathbf{q}$ and $\boldsymbol{\mu}$ that satisfy

$$\boldsymbol{\Psi}(\mathbf{u}_1', \mathbf{z}) \equiv \begin{bmatrix} \mathbf{u}_1' - \mathbf{f}_1(\boldsymbol{\mu}_1) \\ \mathbf{r}_2(\mathbf{q}) + \mathbf{A}_2(\mathbf{q})\,\mathbf{f}_2(\boldsymbol{\mu}_2) - \mathbf{r}_1(\mathbf{q}) - \mathbf{A}_1(\mathbf{q})\,\mathbf{f}_1(\boldsymbol{\mu}_1) \\ \boldsymbol{\Phi}(\mathbf{q}, \boldsymbol{\mu}) \end{bmatrix} = \mathbf{0} \qquad (6.9)$$

Thus the domain of interference is $I_1'D_2 = \{\mathbf{u}_1' : \boldsymbol{\Psi}(\mathbf{u}_1', \mathbf{z}) = \mathbf{0},\ \textit{for some}\ \mathbf{z}\}$.

6.3.1 Planar Stewart Platform with Dome and Room

Consider the planar Stewart platform and dome of Section 6.2.1, operating in a room that is 3.0 units high and 5.4 units wide. The global reference frame is used as the local frame for the room and the center of the room is at $(0.7, 1.5)$. The room is selected so that the dome will interfere with the ceiling and the right wall, but not the floor and the left wall. The coordinates of points outside of the room are $\mathbf{s}'^P_2 = [\,\mu_{21}, \mu_{22}\,]^T$. Equations are thus needed to represent the right wall and the ceiling. Points to the right of the right wall can be given as $\mu_{21} = 3.4 + \mu_{23}^2$, and points above the ceiling can be given as $\mu_{22} = 3.0 + \mu_{24}^2$, where μ_{23} and μ_{24} are slack variables. The interference equations are $\mu_{21} - x - \mu_{11}\cos(\phi) + \mu_{12}\sin(\phi) = 0$ and $\mu_{22} - y - \mu_{11}\sin(\phi) - \mu_{12}\cos(\phi) = 0$. The governing equations for $I'_1 D_2$ relative to the right of the right wall are

$$\boldsymbol{\Psi}\left(\mathbf{u}, \mathbf{q}, \boldsymbol{\mu}\right) \equiv \left[\begin{array}{c} u_1 - \mu_{11} \\ u_2 - \mu_{12} \\ \mu_{21} - x - \mu_{11}\cos\phi + \mu_{12}\sin\phi \\ \mu_{22} - y - \mu_{11}\sin\phi - \mu_{12}\cos\phi \\ \ell_1^2 - (x - \cos\phi + 1)^2 - (y - \sin\phi)^2 \\ \ell_2^2 - (1 - x + \cos\phi)^2 - (y - \sin\phi)^2 \\ \ell_3^2 - (x + \cos\phi - 2)^2 - (y + \sin\phi)^2 \\ \mu_{11}^2 + (2 + 1.3(\mu_{12} - 0.5)^3)^2((\mu_{12} - 0.5)^2 - 1) + \mu_{13}^2 \\ \mu_{21} - 3.4 - \mu_{23}^2 \end{array}\right] = \mathbf{0} \qquad (6.10)$$

The governing equations for $I'_1 D_2$ above the ceiling are obtained by replacing the last of Eqs. 6.10 with $\mu_{22} - 3.0 - \mu_{24}^2 = 0$. Initial points on the boundaries of these sets were determined by geometry, and the method of Section 1.3 was used to map boundary segments. Results for $I'_1 D_2$ are presented in Fig. 6.8.

Points in region A of Fig. 6.8 result from the intersection of the dome with the ceiling and points in region B result from intersection with the right wall. The boundary segment from point 3 to point 5 in Fig. 6.8, is due to the edge of the ceiling

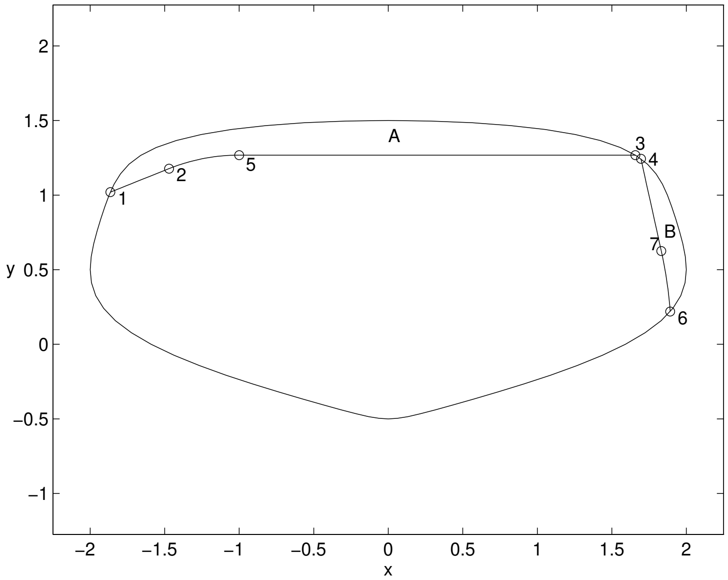

Figure 6.8: $I_1'D_2$ for Stewart Platform with Dome and Room

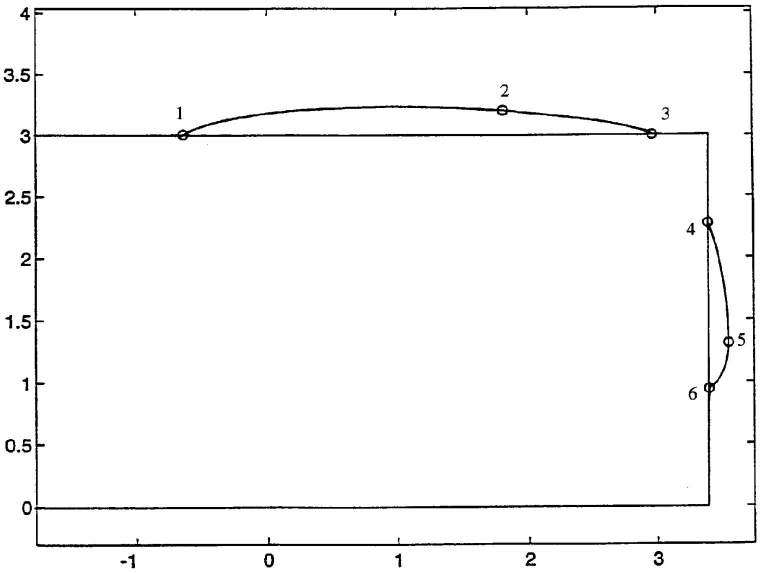

Figure 6.9: $I_2'D_1$ for Stewart Platform with Dome and Room

penetrating the dome, with all legs of the mechanism at full extension. The boundary segment from point 2 to point 5 is the result of the mechanism moving by reducing the length of the third actuator, with the other two legs held at their maximum lengths. The boundary segment from point 6 to point 7 is a result of varying ℓ_3 to its minimum length, with ℓ_1 at maximum and ℓ_2 at minimum length. The segment from point 7 to point 4 is the result of the edge of the ceiling being traced with ℓ_1 at maximum, and both ℓ_2 and ℓ_3 at minimum length. Similarly, results for $I_2'D_1$ are presented in Fig. 6.9. The boundary segment from point 1 to point 2 in Fig. 6.9 is due to the edge of the dome penetrating into the ceiling, with all legs of the mechanism at full extension. The boundary segment from point 2 to point 3 is the result of the mechanism moving by reducing the length of the second actuator, with the other two legs held at their maximum lengths. The boundary segment from point 4 to point 5 is a result of varying ℓ_3 to its minimum length, with ℓ_1 at maximum and ℓ_2 at minimum length. The segment from point 5 to point 6 is the result of the edge of the dome being traced with ℓ_1 at maximum and both ℓ_2 and ℓ_3 at minimum length.

6.3.2 Interference Between Two Planar Stewart Platform Domes

Consider the platform and dome of the previous example and add a second platform and dome, translated to the right 4.75 units in the global frame, as shown in Fig. 6.10. These two domed platform mechanisms are free to move independently and the separation is selected so that there are configurations in which the domes will interfere slightly with each other. The dome on the left is body 1 and the dome on the right is body 2. The equations that represent the second platform are the same as those given in Eq. 6.5, but with x replaced by $x^* - 4.75$ and right platform generalized coordinates $\mathbf{q}^* = [x^*, y^*, \phi^*, v_1^*, v_2^*, v_3^*]^T$. The equations that represent the right dome

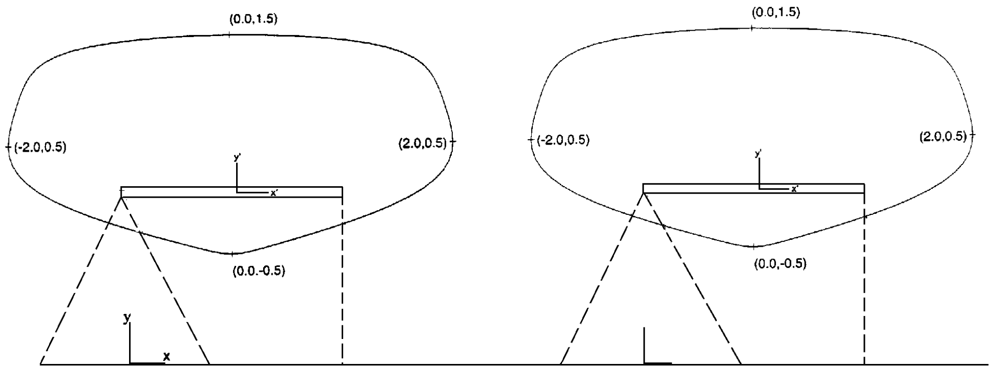

Figure 6.10: Two Planar Stewart Platforms With Domes

are the same as in Eq. 4.2, but with $\boldsymbol{\mu}_1$ replaced by $\boldsymbol{\mu}_2 = [\,\mu_{21},\mu_{22},\mu_{23}\,]^T$. Points in the domain of this dome are thus characterized by $\mathbf{s'}_2^P = [\,\mu_{21},\mu_{22}\,]^T$.

Interference equations for the domes are

$$\begin{bmatrix} x^* + \mu_{21}\cos(\phi^*) - \mu_{22}\sin(\phi^*) - x - \mu_{11}\cos(\phi) + \mu_{12}\sin(\phi) \\ y^* + \mu_{21}\sin(\phi^*) + \mu_{22}\cos(\phi^*) - y - \mu_{11}\sin(\phi) - \mu_{12}\cos(\phi) \end{bmatrix} = 0 \tag{6.11}$$

The governing equations for $I_1'D_2$ are then

$\Psi_1(\mathbf{u}, \mathbf{q}, \mathbf{q}^*, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2) \equiv$

$$\begin{bmatrix} u_1 - \mu_{11} \\ u_2 - \mu_{12} \\ x^* + \mu_{21}\cos(\phi^*) - \mu_{22}\sin(\phi^*) - x - \mu_{11}\cos(\phi) + \mu_{12}\sin(\phi) \\ y^* + \mu_{21}\sin(\phi^*) + \mu_{22}\cos(\phi^*) - y - \mu_{11}\sin(\phi) - \mu_{12}\cos(\phi) \\ \ell_1^2 - (x - \cos\phi + 1)^2 - (y - \sin\phi)^2 \\ \ell_2^2 - (1 - x + \cos\phi)^2 - (y - \sin\phi)^2 \\ \ell_3^2 - (x + \cos\phi - 2)^2 - (y + \sin\phi)^2 \\ \mu_{11}^2 + (2 + 1.3(\mu_{12} - 0.5)^3)^2((\mu_{12} - 0.5)^2 - 1) + \mu_{13}^2 \\ \ell^{*2}_1 - (x^* - \cos\phi^* - 3.75)^2 - (y^* - \sin\phi^*)^2 \\ \ell^{*2}_2 - (5.75 - x^* + \cos\phi^*)^2 - (y^* - \sin\phi^*)^2 \\ \ell^{*2}_3 - (x^* + \cos\phi^* - 6.75)^2 - (y^* + \sin\phi^*)^2 \\ \mu_{21}^2 + (2 + 1.3(\mu_{22} - 0.5)^3)^2((\mu_{22} - 0.5)^2 - 1) + \mu_{23}^2 \end{bmatrix} = 0 \tag{6.12}$$

Assembled configurations of the mechanisms with the domes overlapping were found

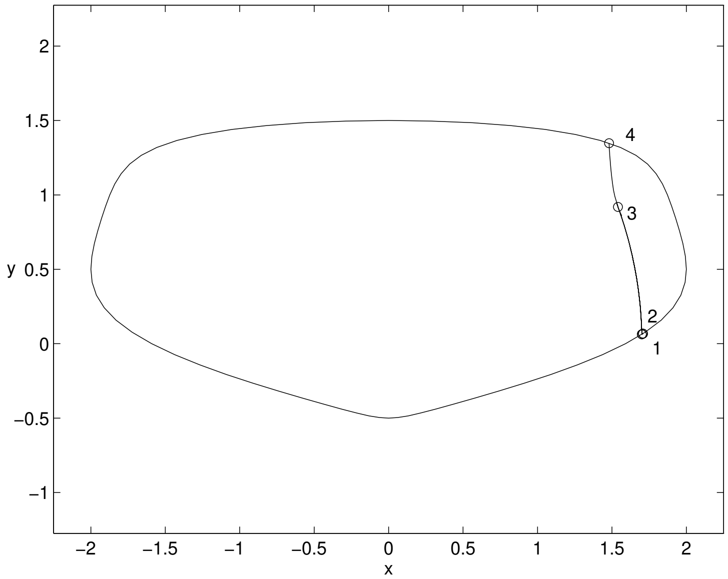

Figure 6.11: $I_1'D_2$ For Two Planar Stewart Platforms With Domes

geometrically and the method of Section 1.3.2 was used to find initial points on the boundaries. From these initial points, the method of Ref. 40 was used to map boundary segments. Results for $I_1'D_2$ are presented in Fig. 6.11, and $I_2'D_1$ is presented in Fig. 6.12.

In Fig. 6.11, a short segment is traced from point 1 to point 2 in which both body 1 and body 2 are moving. The curve from point 2 to point 3 is associated with the second platform in the fixed configuration $\mathbf{v}^* = \left[-\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}\right]^T$, with the first body moving such that its first leg is at its maximum length, the second at minimum length, and the third changing. Along the curve from point 3 to point 4, both platforms are fixed and the curve is thus the result of a portion of the boundary of the second dome interfering with the first dome. Points 1, 3, and 4 are higher order bifurcation points, where the row rank deficiency of $\mathbf{G_x}$ was greater than 1. In Fig. 6.12, points 1 and 4 are higher order bifurcation points. Along the curve from point 3 to point 4, both

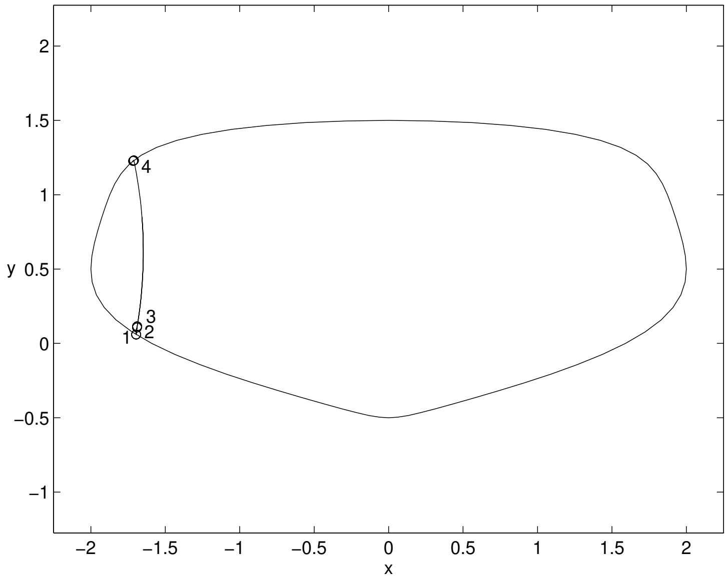

Figure 6.12: $I_2' D_1$ For Two Planar Stewart Platforms With Domes

domes are in a fixed configuration. Along the curves from point 1 to point 2 and from point 2 to point 3, body 1 is moving and body 2 is fixed.

CHAPTER 7
CONCLUSIONS AND RECOMMENDATIONS

7.1 Summary of New Results

New results are presented in Chapter 3 for computing solution curves at a certain class of bifurcation points. These results take advantage of mathematical models that include a technique for reducing the dimension of output spaces in mechanism workspace and related problems by intersecting with planes. Chapter 2 shows the equivalence of the two possible methods for this reduction and gives computational performance comparisons. This is an important result, since the technique for reducing output space dimension to one dimensional curves in two dimensional planes is used in Refs. 33, 34, 35, 36, and 40 and the result shows that using one of the two models provides a faster solution, with fewer equations and variables.

The class of bifurcation points treated in Chapter 3 is characterized by the condition that the Jacobian matrix $\mathbf{\Phi_z}$ is row rank deficient two. This chapter gives both new theory and numerical algorithms for faster computation than was previously possible of the branches and tangents for this common class of bifurcation points. The advances include new proofs of the invertibility of a certain matrix, which results in numerical efficiency because no Moore-Penrose inverses are required. The spatial serial manipulator example illustrates the need to consider cases in which repeated eigenvalues occur and demonstrates the procedure for applying the second order necessary condition using HOMPACK. This solution procedure, applying the second order necessary condition, is applicable to any type of multiple bifurcation point at which the Jacobian matrix $\mathbf{G_x}(\mathbf{x})$ has row rank deficiency greater than one. A recent paper by Coroian and Adkins [12] provides an extension to the specific multiple bifurcation points in which $\mathbf{\Phi_z}$ has row rank deficiency three.

The dense output procedure developed in Chapter 4 provides a fast method to obtain refined discretizations of boundary curves. Section 4.3.2 shows that using the tangent as the augmenting vector in Eq. 4.20 provides the best choice for correction iteration. The dense output routine enables good graphical representation for boundary curves of two dimensional output sets or solution curves within cutting planes. Challenges remain to incorporate information computed in one cutting plane with the next in a family of planes. This dense output routine could be the basis for triangulation methods on two dimensional boundary surfaces in $\Re^3$. These results, combined with the continuation point iteration procedure enhancements presented in Section 4.3, further the goal of providing a fast and reliable workspace analysis tool for use by engineers.

The numerical algorithm for barrier analysis presented in Chapter 5 is illustrated using planar manipulators with unilateral constraints to demonstrate the feasibility of a general purpose formulation and computer implementation of methods for analysis of barriers to control of manipulators. The method assists in identifying curves in the set S that contains the exterior boundary of the accessible output set. It shows that all curves in S are restrictions to control of nonzero normal output velocity, when the manipulator is in the associated singular configuration. The method also determines local mobility restrictions along selected curves in S that are interior to the accessible output set. While the experimental computer code used to implement the algorithm is not refined, it nevertheless has demonstrated the ability to automate the process of determining barriers to output control.

Chapter 6 presents formulations and results for two new areas of workspace analysis, domains of operation and domains of interference. These formulations demonstrate the broad applicability of the numerical methods to realistic engineering applications. The example in Section 6.2.2 gives both a realistic spatial example of the operational envelope of a complex manipulator and some insight into difficulties involved in the computation of its boundary. While the formulation of the operational envelope equations for the Iowa Driving Simulator is a direct application of methods presented in Ref. 39, with equations from Ref. 41 and numerical methods of Ref. 40, substantial computation is required to map the boundary surface of the operational envelope. Complex collections of curves that satisfy continuation equations and the presence of curves that intersect in the output space but not in the higher dimensional space to form the outer boundary, create complications for computation of boundary surface of the operational envelope. Results presented demonstrate, however, that this formulation and numerical method can be implemented for realistic and practical problems.

7.2 Suggestions for Future Research

Some challenges remain in providing workspace analysis as a general computational tool for working engineers. These challenges include implementing the numerical framework presented in this thesis and related literature in a widely available form, including easy to use interfaces and graphical output options.

In order to provide simple software interfaces, automatic formulation of the mechanical system equations and required boundary surface continuation equations needs to be available, freeing the user from coding these equations. Systematic modelling of the kinematics of mechanisms [31] is a way to provide the required equations.

The numerical workspace analysis methods require at least two derivatives. These can be provided by automatic differentiation or by using derivative information resulting from the structures obtained when systematic modelling is used [31].

Additional development of graphical output for computed boundary surfaces when the output dimension is three or greater needs to occur before easy to use workspace analysis software packages can be made commercially available. Standard plotting routines can be effectively used for two dimensional output space curves, but the three dimensional case needs to collect the computed family of cutting plane curves. These challenges include combining points across cutting planes and mapping very irregular surfaces. The dense output method developed in Chapter 4 will be a key tool in developing good graphical output routines.

The numerical methods for analysis of workspace barriers in Chapter 5 need to be developed into a general purpose software tool. Examples analyzed in this thesis used code specifically developed to post-process the accessible output set data. Barrier analysis could be done while computing the accessible output set, and should provide additional information about which singular curves are interior and which are associated with ∂A.

Automatic scaling of workspace problems is necessary for a general implementation. For some of the examples that have been developed in this thesis, the units of the variables had to be rescaled to get good results. Such improvements may aid in the solution of second order necessary conditions as indicated in Chapter 3, because the resulting systems are very sensitive to scaling. In cases where the Jacobian of the continuation equations has row rank deficiency and bifurcation points are detected, the iteration procedure for determining the bifurcation points should be improved to include a singular value decomposition which checks the magnitude of the smallest singular values and, if more than one singular value is very small, the iteration process should attempt to converge to a coordinate where the associated Jacobian has row rank deficiency the same as the number of small singular values.

In conclusion, this thesis furthers the development of of a general numerical formulation for determining implicitly defined boundary surfaces of workspaces. The next step is to add good graphical output routines and user interfaces for automatic formulation of the required equations. When this is done, workspace analysis will be made efficient and easy to use for the working engineer.

REFERENCES

[1] K. Abdul-Malek and H.-J. Yeh. Analytic output control barriers of serial manipulators. *International Journal of Robotics Research*, 1996. Submitted.

[2] S.K. Agrawal. Workspace boundaries of in-parallel manipulator systems. *International Journal of Robotics and Automation*, 7(2):94–99, 1992.

[3] S.K. Agrawal and R. Garimella. Workspace boundaries of free–floating open and closed chain planar manipulators. *Journal of Mechanical Design*, 116:105–110, March 1994.

[4] E.L. Allgower, K. Böhmer, and M. Golubitsky, editors. *Bifurcation and Symmetry: Cross Influence Between Mathematics and Applications*, volume 104 of *International Series of Numerical Mathematics*, Boston, 1992. Birkhäuser Verlag.

[5] E.L. Allgower and K. Georg, editors. *Computational Solution of Nonlinear Systems of Equations*, volume 26 of *Lectures in Applied Mathematics*, Providence, 1990. American Mathematical Society.

[6] E.L. Allgower and K. Georg. *Numerical Continuation Methods: An Introduction.* Springer-Verlag, Berlin, 1990.

[7] E.L Allgower and S. Gnutzmann. An algorithm for piecewise-linear approximation of implicitly defined two-dimensional surfaces. *SIAM Journal of Numerical Analysis*, 24(2):452–469, 1985.

[8] E.L Allgower and P.H. Schmidt. An algorithm for piecewise-linear approximation of an implicitly defined manifold. *SIAM Journal of Numerical Analysis*, 22(2):322–335, 1985.

[9] K.E. Atkinson. *An Introduction to Numerical Analysis.* John Wiley & Sons, New York, 2nd edition, 1989.

[10] C. Bischof, L. Roh, and A. Mauer. ADIC: A tool for the automatic differentiation of C program. Technical report, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439 U.S.A., 1996. In preparation.

[11] A. Castelo, S. De Freitas, and G. Tavares. PL approximation to manifolds and its application to implicit ODEs. In E.L. Allgower and K. Georg, editors, *Computational Solution of Nonlinear Systems of Equations*, volume 26 of *Lectures in*

Applied Mathematics, pages 99–111, Providence, 1990. American Mathematical Society.

[12] D.I. Coroian and F.A. Adkins. Calculation of continuation curves at multiple bifurcation points when mapping the boundaries of manipulator workspaces. Technical Report R-199, Center for Computer Aided Design, University of Iowa, Iowa City, Iowa 52242, U.S.A., 1996.

[13] D.W. Decker and H.B. Keller. Path following near bifurcation. In *Communications on Pure and Applied Mathematics*, volume 34, pages 149–175, New York, 1981. John Wiley & Sons.

[14] J. Denavit and R.S. Hartenberg. A kinematic notation for lower pair mechanisms based on matrices. *Journal of Applied Mechanics*, 77:215–221, 1955.

[15] J.P. Fink and W.C. Rheinboldt. Numerical methods for a class of finite dimensional bifurcation problems. *SIAM Journal of Numerical Analysis*, 15:1–15, 1978.

[16] J.P. Fink and W.C. Rheinboldt. Solution manifolds and submanifolds of parameterized equations and their discretization errors. *Numerical Mathematics*, 45:323–343, 1984.

[17] J.P. Fink and W.C. Rheinboldt. A geometric framework for the numerical study of singular points. *SIAM Journal of Numerical Analysis*, 24:618–633, 1987.

[18] E.F. Fitchter. A Stewart platform-based manipulator: General theory and practical construction. *International Journal of Robotics Research*, 5(2):157–182, 1986.

[19] F. Freudenstein and E.J. Primrose. On the analysis and synthesis of the workspace of a three-link, turning-pair connected robot arm. *Journal of Mechanisms, Transmissions, and Automation in Design*, 106:365–370, 1984.

[20] A. Frommer. A unified approach to methods for the simultaneous computation of all zeros of generalized polynomials. *Numerische Mathematik*, 54(1):105–116, 1988.

[21] C.B. Garcia and W.I. Zangwill. Global continuation methods for finding all solutions to polynomial systems of equations in N variables. Technical Report 7755, Center for Mathematical Studies in Business and Economics, University of Chicago, Chicago, IL 60637 U.S.A., 1977.

[22] K. Georg. An introduction to PL algorithms. In E.L. Allgower and K. Georg, editors, *Computational Solution of Nonlinear Systems of Equations*, volume 26 of *Lectures in Applied Mathematics*, pages 207–236, Providence, 1990. American Mathematical Society.

[23] C. Gosselin. Determination of the workspace of 6-DOF parallel manipulators. *Journal of Mechanical Design*, 112:331–336, 1990.

[24] A. Griewank. On solving nonlinear equations with simple singularities or nearly singular solutions. *SIAM Reviews*, 27:537–563, 1985.

[25] A. Griewank. On automatic differentiation. In M. Iri and K. Tanabe, editors, *Mathematical Programming: Recent Developments and Applications*, pages 83–108, Boston, 1989. Kluwer Academic Publishers.

[26] A. Griewank. Automatic evaluation of first and higher-derivative vectors. In R. Seydel, editor, *Bifurcation and Chaos: Analysis, Algorithms, Applications*, volume 97 of *International Series of Numerical Mathematics*, pages 135–148, Boston, 1991. Birkhäuser Verlag.

[27] A. Griewank, D. Juedes, and J. Srinivas. ADOL-C: A package for the automatic differentiation of algorithms written in C/C++. Preprint MCS-P180-1190, Argonne National Laboratory, Argonne, IL 60439 U.S.A., 1990.

[28] A. Griewank and G.W. Reddien. Characterization and computation of generalized turning points. *SIAM Journal of Numerical Analysis*, 21(1):176–185, February 1984.

[29] K.C. Gupta and B. Roth. Design considerations for manipulator workspace. *Journal of Mechanical Design*, 103:776–784, 1981.

[30] E. Hairer, S.P. Norsett, and G. Wanner. *Solving Ordinary Differential Equations: Volume I: Nonstiff Problems.* Springer-Verlag, New York, 2nd edition, 1993.

[31] E.J. Haug. *Computer Aided Kinematics and Dynamics of Mechanical Systems.* Allyn and Bacon, Boston, 1989.

[32] E.J. Haug, F.A. Adkins, and D.I. Coroian. Domains of mobility. *Advances in Design Automation*, ASME 24th Biennial Mechanisms Conference, Irving, CA, 1996. accepted.

[33] E.J. Haug, F.A. Adkins, and C.-M. Luh. Domains of interference between working bodies in mechanisms and manipulators. *Advances in Design Automation*, ASME DE 82(1):651–658, 1995.

[34] E.J. Haug, F.A. Adkins, and C.-M. Luh. Domains of operation and interference for bodies in mechanisms and manipulators. In J.-P. Merlet and B. Ravani, editors, *Proceedings of the Second Workshop on Computational Kinematics, Sophia Antipolis, France*, number 40 in Solid Mechanics Applications, pages 193–202, Boston, 1995. Kluwer Academic Publishers.

[35] E.J. Haug, F.A. Adkins, and C.-M. Luh. Operational envelopes for working bodies in mechanisms and manipulators. *Advances in Design Automation*, ASME DE 82(1):673–679, 1995.

[36] E.J. Haug, F.A. Adkins, C.C. Qiu, and J. Yeng. Analysis of barriers to control of manipulators. *Advances in Design Automation*, ASME DE 82(1):697–704,

1995.

[37] E.J. Haug and K.K. Choi. *Methods of Engineering Mathematics.* Prentice-Hall, Englewood Cliffs, New Jersey, 1994.

[38] E.J. Haug, K.K. Choi, and V. Komkov. *Design Sensitivity Analysis of Structural Systems.* Academic Press, Orlando, Florida, 1986.

[39] E.J. Haug, C.-M. Luh, and F.A. Adkins. Operational envelopes for working bodies in mechanisms and manipulators. *Journal of Mechanical Design*, to appear, 1996.

[40] E.J. Haug, C.-M. Luh, F.A. Adkins, and J.Y Wang. Numerical algorithms for mapping boundaries of manipulator workspaces. *Journal of Mechanical Design*, 118(2):228–234, 1996.

[41] E.J. Haug, J.Y. Wang, and J.K. Wu. Dextrous workspaces of manipulators, Part I: Analytical criteria. *Mechanics of Structures and Machines*, 20(3):321–361, 1992.

[42] M. Hazewinkel, R. Jurkovich, and J.H.P. Paelinck, editors. *Bifurcation Analysis : Principles, Applications, and Synthesis*, Boston, 1985. Kluwer Academic Publishers.

[43] C. Den Heijer and W.C. Rheinboldt. On steplength algorithms for a class of continuation methods. *SIAM Journal of Numerical Analysis*, 18:925–945, 1981.

[44] B. Hong. On the computation of multiple bifurcations with multiple parameters and symmetry. *SIAM Journal of Numerical Analysis*, 30(4):1134–1154, August 1993.

[45] A. Hoy. A new approach for solving singular nonlinear equations. In E.L. Allgower, K. Böhmer, and M. Golubitsky, editors, *Bifurcation and Symmetry: Cross Influence Between Mathematics and Applications*, volume 104 of *International Series of Numerical Mathematics*, pages 179–189, Boston, 1992. Birkhäuser Verlag.

[46] M.S. Hsu and D. Kohli. Boundary surfaces and accessibility regions for regional structures of manipulators. *International Journal of Mechanism and Machine Theory*, 1986.

[47] M.S. Hsu and D. Kohli. The Jacobian analysis of mechanical manipulators. *International Journal of Mechanism and Machine Theory*, 1986.

[48] M. Iri and K. Tanabe, editors. *Mathematical Programming: Recent Developments and Applications*, Boston, 1989. Kluwer Academic Publishers.

[49] D.Y. Jo and E.J. Haug. Workspace analysis of multibody mechanical systems using continuation methods. *Journal of Mechanisms, Transmissions, and Automation in Design*, 3:581–589, 1989.

[50] A. Jones, A. Gray, and R. Hutton. *Manifolds and Mechanics.* Cambridge University Press, New York, 1987.

[51] A. Kumar and K.J. Waldron. The workspace of a mechanical manipulator. *Journal of Mechanical Design*, 103:665–672, July 1981.

[52] P. Kunkel. Augmented systems for generalized turning points. In R. Seydel, editor, *Bifurcation and Chaos: Analysis, Algorithms, Applications*, volume 97 of *International Series of Numerical Mathematics*, pages 231–236, Boston, 1991. Birkhäuser Verlag.

[53] T. Küpper, R. Seydel, and H. Troger, editors. *Bifurcation: Analysis, Algorithms Applications*, volume 79 of *International Series of Numerical Mathematics*, Boston, 1987. Birkhäuser Verlag.

[54] Z.C. Lai and D.C.H. Yang. On the singularity anaylsis of simple six-link manipulators. ASME DET paper number 84, ASME, New York, 1984.

[55] Z.C. Lai and D.C.H. Yang. A new method for the singularity analysis of simple six-link manipulators. *International Journal of Robotics Research*, 5(2):66–74, 1986.

[56] T.W. Lee and D.C.H. Yang. On the evaluation of manipulator workspace. *Journal of Mechanisms, Transmissions, and Automation in Design*, 105:70–77, 1983.

[57] H. Lipkin and E. Pohl. Enumeration of singular configurations for robotic manipulators. *Journal of Mechanisms, Transmissions, and Automation in Design*, 113:272–279, 1991.

[58] F.L. Litvin. Application of theorem of implicit function system existence for analysis and synthesis of linkages. *Mechanism and Machine Theory*, 15:115–125, 1980.

[59] F.L. Litvin, P. Fanghella, J. Tan, and Y. Zhang. Singularities in motion and displacement functions of spatial linkages. *Journal of Mechanisms, Transmissions, and Automation in Design*, 108(4):516–523, 1986.

[60] F.L. Litvin, Z. Yi, V.P. Castelli, and C. Innocenti. Singularities, configurations, and displacement function for manipulators. *International Journal of Robotics Research*, 5(2):52–65, 1986.

[61] C.-M. Luh. *Numerical Methods for Workspace Analysis of Mechanisms and Manipulators.* Ph.D. dissertation, University of Iowa, 1995.

[62] C.-M. Luh, F.A. Adkins, E.J. Haug, and C.C. Qiu. Working capability analysis of Stewart platforms. *Journal of Mechanical Design*, 118(2):220–227, 1996.

[63] R. Menzel and G. Pönisch. A quadratically convergent method for computing simple singular roots and it application to determining simple bifurcation points. *Computing*, 32:127–138, 1984.

[64] J.-P. Merlet. Singular configurations of parallel manipulators and Grassman geometry. *International Journal of Robotics Research*, 8:45–56, 1989.

[65] J.-P. Merlet and B. Ravani, editors. *Proceedings of the Second Workshop on Computational Kinematics, Sophia Antipolis, France*, Boston, 1995. Kluwer Academic Publishers.

[66] H.D. Mittelmann and D. Roose, editors. *Continuations techniques in Bifurcation problems*, volume 92 of *International Series of Numerical Mathematics*, Boston, 1990. Birkhäuser Verlag.

[67] H.R. Daniali Mohammadi, P.J. Zsombor-Murray, and J. Angeles. Singularity analysis of planar parallel manipulators. *Mechanism and Machine Theory*, 30:665–678, 1995.

[68] G. Moore and A. Spence. The calculation of turning points of nonlinear equations. *SIAM Journal of Numerical Mathematics*, 17:567–76, 1980.

[69] A.P. Morgan. A method for computing all solutions to systems of polynomial equations. *ACM Transactions on Mathematical Software*, 9(1):1–17, March 1983.

[70] A.P. Morgan, A.J. Sommes, and C.W. Wampler. Polynomial continuation for mechanism design problems. In E.L. Allgower and K. Georg, editors, *Computational Solution of Nonlinear Systems of Equations*, volume 26 of *Lectures in Applied Mathematics*, pages 495–517, Providence, 1990. American Mathematical Society.

[71] A.P. Morgan, A.J. Sommese, and L.T. Watson. Finding all isolated solutions to polynomial systems using HOMPACK. *ACM Transactions on Mathematical Software*, 15(2):93–122, June 1989.

[72] E.D. Nering. *Linear Algebra and Matrix Theory.* John Wiley & Sons, New York, 2nd edition, 1970.

[73] L. Nielsen, C. Canudas deWit, and P. Hagander. Controllability issues of robots in singular configurations. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 2210–2215, Sacramento, California, 1991.

[74] D. Oblak and D. Kohli. Boundary surfaces, limit surfaces, crossable and noncrossable surfaces in workspace of mechanical manipulators. *Journal of Mechanisms, Transmissions, and Automation in Design*, 110:389–396, 1988.

[75] D.L. Piper and B. Roth. The kinematics of manipulators under computer control. In *Proceedings of the Second International Conference on the Theory of Machines and Mechanisms*, Warsaw, 1969.

[76] G. Pönisch. On computing coupled turning points on parameter dependent

nonlinear equations. In R. Seydel, editor, *Bifurcation and Chaos: Analysis, Algorithms, Applications*, volume 97 of *International Series of Numerical Mathematics*, pages 289–294, Boston, 1991. Birkhäuser Verlag.

[77] G. Pönisch and H. Schwetlick. Computing turning points of curves implicitly defined by nonlinear equations depending on a parameter. *Computing*, 26:107–121, 1981.

[78] C.C. Qiu, C.-M. Luh, and E.J. Haug. Dextrous workspaces of manipulators, Part III: Calculation of continuation curves at bifurcation points. *Mechanics of Structures and Machines*, 23(1):115–130, 1995.

[79] J. Rastegar. Workspace analysis of 4R manipulators with various degrees of dexterity. *Journal of Mechanisms, Transmissions, and Automation in Design*, 110:2–47, 1988.

[80] W.C. Rheinboldt. *Numerical Analysis of Parameterized Nonlinear Equations.* John Wiley & Sons, New York, 1986.

[81] W.C. Rheinboldt. On a moving frame algorithm and the triangulation of equilibrium manifolds. Technical Report ICMA-86-100, Institute of Computational Mathematics and Applications, Department of Mathematics and Statistics, University of Pittsburgh, Pittsburgh, PA 15260, 1986.

[82] W.C. Rheinboldt. On computation of multidimensional solution manifold of parameterized equations. Technical Report ICMA-86-102, Institute of Computational Mathematics and Applications, Department of Mathematics and Statistics, University of Pittsburgh, Pittsburgh, PA 15260, 1986.

[83] B. Roth. Performance evaluation of manipulators from a kinematic viewpoint. In *Performance Evaluation of Programmable Robots and Manipulators*, NBS Special Publication 459, pages 39–61, October 1976.

[84] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators.* McGraw Hill, New York, 1996.

[85] R. Seydel, editor. *Bifurcation and Chaos: Analysis, Algorithms, Applications*, volume 97 of *International Series of Numerical Mathematics*, Boston, 1991. Birkhäuser Verlag.

[86] B. Shimano and B. Roth. Ranges of motion of manipulators. In *Proceedings of the Second International CISM-IFToMM*, Warsaw, September 1976.

[87] R. Soylu and J. Duffy. Hypersurfaces of special configurations of serial manipulators and related concepts. Part II: Passive joints, configurations, component manifolds and some applications. *Journal of Robotic Systems*, 5:31–53, 1988.

[88] J. Spanos and D. Kohli. Workspace analysis of regional structures of manipulators. *Journal of Mechanisms, Transmissions, and Automation in Design*, 107:216–222, 1985.

[89] D. Stewart. A platform with six degrees of freedom. *Proceedings of the Institute for Mechanical Engineering*, 180:371–386, 1965.

[90] G. Strang. *Linear Algebra and Its Applications*. Academic Press, New York, 1980.

[91] A.E. Taylor and W.R. Mann. *Advanced Calculus*. John Wiley & Sons, New York, 1983.

[92] V.D. Tourassis and M.H. Ang. Identification and analysis of robot manipulator singularities. *International Journal of Robotics Research*, 11:248–259, 1992.

[93] M.J. Tsai and K.J. Waldron. Operating barriers within the workspaces of manipulators. In *Society of Manufacturing Engineers, Robots II*, 17th ISIR, pages 27–30, Chicago, IL, 1987.

[94] Y.C. Tsai and A.H. Soni. Accessible region and synthesis of robot arms. *Journal of Mechanical Design*, 103:803–811, 1981.

[95] Y.C. Tsai and A.H. Soni. An algorithm for the workspace of a general n-R robot. *Journal of Mechanisms, Transmissions, and Automation in Design*, 105:52–58, 1985.

[96] K.J. Waldron, S. Wang, and S.J. Bolin. A study of the Jacobian matrix of serial manipulators. *Journal of Mechanisms, Transmissions, and Automation in Design*, 107(2):230–238, 1985.

[97] J.Y. Wang and J.K. Wu. Dextrous workspaces of manipulators, Part II: Computational methods. *Mechanics of Structures and Machines*, 21(4):471–506, 1993.

[98] S.L. Wang and K.J. Waldron. Study of the singular configurations of serial manipulators. *Journal of Mechanisms, Transmissions, and Automation in Design*, 109(1):14–20, 1987.

[99] L.T. Watson, S.C. Billups, and A.P. Morgan. Algorithm 652 HOMPACK: A suite of codes for globally convergent homotopy algorithms. *ACM Transactions on Mathematical Software*, 13(3):281–310, September 1987.

[100] H. Weber and W. Werner. On the accurate determination of nonisolated solutions on nonlinear equations. *Computing*, 26:315–326, 1981.

[101] D.C.H. Yang and T.W. Lee. On the workspace of mechanical manipulators. *Journal of Mechanisms, Transmissions, and Automation in Design*, 105:62–69, 1983.

[102] F.-C. Yang and E.J. Haug. Numerical analysis of the kinematic dexterity of mechanisms. *Journal of Mechanical Design*, 116(1):119–126, March 1994.

[103] F.-C. Yang and E.J. Haug. Numerical analysis of the kinematic working capability of mechanisms. *Journal of Mechanical Design*, 116(1):111–118, March 1994.

[104] H.-J. Yeh and K. Abdel-Malek. Analytical boundary of the accessible output set for serial four degree-of-freedom manipulators. *Journal of Mechanical Design*, 1996. submitted.

[105] H.-J. Yeh and K. Abdul-Malek. Crossability analysis on singular surfaces of serial robot manipulators. *International Journal of Robotics Research*, 1996. To appear.

[106] H.-J. Yeh and K. Abdul-Malek. Crossable surfaces of robotic manipulators with joint limits. *Journal of Mechanical Design*, 1996. To appear.

[107] K.-Y. Young and C.H. Wu. Path feasibility and modification. *Journal of Robotics Systems*, 9(5):613–633, 1992.

[108] C.-D. Zhang and S.M. Song. Forward position analysis of nearly general Stewart platforms. *Journal of Mechanical Design*, 116:54–59, March 1994.