METHODS FOR THE DETERMINATION OF ACCESSIBLE WORKSPACES OF PLANAR STEWART PLATFORMS OF GENERAL DESIGN

by

Alexander Morrison Hay

Submitted in partial fulfillment of the requirements for the degree

Master of Engineering

in the
Faculty of Engineering,
University of Pretoria

December 1999

ABSTRACT

Methods for the determination of accessible workspaces of planar Stewart platforms of general design

by

Alexander Morrison Hay

Supervisor: Professor J. A. Snyman

Department of Mechanical and Aeronautical Engineering

Degree: Master of Engineering

Keywords: optimization, workspace determination, planar Stewart platform, parallel manipulator, non-convexity, redundancy.

In recent years Stewart platforms have been increasingly studied and developed. These parallel manipulators offer a number of advantages over traditional serial manipulators including high rigidity, good positioning accuracy and high load to weight ratio. The main disadvantage associated with parallel manipulators is that they have relatively limited workspaces. Numerous researchers have thus emphasized the need to develop refined methods for the determination of workspaces of such manipulators.

This study is primarily concerned with extensions to a novel optimization approach for the determination of manipulator accessible output sets. The optimization approach provides a *general* method for the determination of workspaces of both serial and parallel manipulators and has the considerable advantage that it may easily be automated. Furthermore, the approach allows for the easy and systematic implementation of various physical constraints acting on manipulators.

Established methods for workspace determination are reviewed and illustrated by application to a simple two degree of freedom example.

The original optimization approach is extended and generalized to enable the determination of *non-convex* workspaces. Simply stated, the approach consists of finding the points of intersection of the workspace boundary with a number of successive search elements. The points of intersection are determined by means of optimization techniques in which a dynamic constrained optimization algorithm is used.

Two new methodologies, the *modified ray method* and the *chord method*, are proposed. Differences between these methods are illustrated using a simple example. The optimization approach, embodied in the proposed methodologies, is applied to the determination of workspaces of planar Stewart platforms of varied designs. A formulation for all constraints acting on planar Stewart platforms is introduced and implemented in the optimization approach.

A special case of manipulator geometry, where the orientation of the platform is effectively redundant in determining the extreme reach of the manipulator, is identified and studied. A slight modification to the optimization methodologies is introduced to allow for the determination of workspaces of such *redundant manipulators*.

The modified ray and chord methods proposed in this study have proven capable of determining convex and non-convex manipulator workspaces. Of the two new methods, the chord approach is the most reliable in determining non-convex workspaces. Both optimization methodologies have been implemented in practical interactive computer systems, which allow for the easy determination of workspaces of planar Stewart platforms of arbitrary geometry.

SAMEVATTING

Metodes vir die bepaling van bereikbare werkruimtes van vlak Stewart platforms met algemene ontwerpe

deur

Alexander Morrison Hay

Studie-leier: **Professor J. A. Snyman**

Department van Meganiese en Lugvaartkundige Ingeneurswese

Graad: Magister in Ingenieurswese

Sleutelwoorde: optimering, werkruimte-bepaling, vlak Stewart platform, parallel-manipuleerder, nie-konveksiteit, oortolligheid

Gedurende die afgelope paar jaar is Stewart platforms toenemend bestudeer en ontwikkel. Ingesluit by die voordele wat parallel-manipuleerders bied in vergelyking met tradisionele serie-manipuleerders, is 'n hoë styfheid, goeie posisionele akuraatheid en 'n hoë las-tot-gewig verhouding. Die hoofnadeel geassosieer met parallel-manipuleerders is hulle relatief beperkte werkruimtes. Verskeie navorsers beklemtoon gevolglik die noodsaaklikheid om verfynde metodes te ontwikkel waarmee die werkruimtes van hierdie meganismes bepaal kan word.

Die klem van hierdie studie val of die uitbreiding van 'n nuwe optimeringsbenadering, waarmee manipuleerders se bereikbare uitset-reekse suksesvol bepaal is. Die optimeringsbenadering is *algemeentoepasbaar* op die werkruimte-bepaling van beide serie- en parallel-manipuleerders, en bied die beduidende voordeel dat dit maklik geoutomatiseer kan word. Bowendien is die benadering ideaal geskik om verskeie fisiese begrensings, waaraan die manipuleerders onderhewig is, eenvoudig en sistematies te implementeer.

Gevestigde werkruimte-bepaling-metodes word beoordeel en geïllustreer aan die hand van 'n eenvoudige meganisme met twee vryheidsgrade.

Die oorspronklike optimeringsmetode word uitgebrei en veralgemeen om die bepaling van *nie-konvekse* werkruimtes te inkorporeer. In eenvoudige terme gestel bestaan die uitgebreide metode uit die snypunt-bepalings van opeenvolgende soek-elemente met die rand van die werkruimte. Die snypunte word bepaal deur optimeringsmetodes waarin 'n dinamiesebegrensde optimeringsalgoritme gebruik word.

Twee verskillende soek-elemente word gebruik wat lei tot die kategorisering van die uitgebreide optimeringsmetode in die *gewysigde straal-metode*, en die *koord-metode*. 'n Illustratiewe voorbeeld lig die verskille tussen die twee metodes uit. Gevolglik word die werkruimtes van verskeie vlak Stewart platforms met verskillende ontwerpe bepaal met die gekategoriseerde optimeringsmetodes. Die voorgestelde formulering van al die fisiese begrensings, wat 'n invloed het op die vlak Stewart Platforms se werkruimtes, word gevolglik ook geïnkorporeer in die optimeringsmetodes.

'n Spesiale manipuleerder-geometrie, waar die beweegbare platform se orientasie inderdaad oortollig is in die bepaling van die manipuleerder se reikafstand, word ook bestudeer. Met behulp van 'n geringe modifikasie, word die voorgestelde optimeringsmetodes suksesvol gebruik om die werkruimtes van sulke *oortollige manipuleerders* mee te bepaal.

Die aanduiding is dat die gewysigde straal- en koord-metodes wat voorgestel word in hierdie studie, in staat is om konvekse en nie-konvekse manipuleerder-werkruimtes te bereken. Die koordmetode kan uitgesonder word as die betroubaarste van die twee waneer dit kom by die bepaling van nie-konvekse werkruimtes. Beide optimeringsmetodes is suksesvol geïmplimenteer in praktiese bruikbare interaktiewe rekenaarstelsels, waarmee die werkruimte van 'n vlak Stewart platform met 'n arbitrêre geometrie redelik maklik bepaal kan word.

ACKNOWLEDGEMENTS

I would like to thank Prof. Jan Snyman for his continued enthusiasm and interest during the course of this research project. His excellent mentorship has inspired and motivated me and will continue to do so in the future.

This study was partially funded by an NRF grant. I would like to acknowledge this financial assistance.

Finally I would like to thank my parents. Without their many years of encouragement and financial support I could not have pursued this study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: OVERVIEW OF STEWART PLATFORMS AND METHODS FOR DETERMINING THEIR WORKSPACES

1.1 Introduction

This study is concerned with an extension of a novel optimization approach to the determination of accessible output sets of planar manipulators. The outstanding feature of the original optimization approach proposed by Snyman et al. [1] and Du Plessis [2] is that it provides a technique for the determination of workspaces that may easily be automated. Furthermore the method allows for the easy and systematic implementation of constraints acting on the mechanism. In addition to these advantages, the optimization approach is generally applicable to parallel and serial manipulators.

The method is illustrated by its application to the planar Stewart platform. Stewart platform parallel manipulators have been increasingly studied and developed over the last few years (Merlet [3]), and offer a number of advantages over traditional serial manipulators (Fichter and McDowell [4]). One of the disadvantages of Stewart platforms is that they tend to have small workspaces in relation to their size. A number of researchers have thus emphasized the need to develop refined computer codes by means of which manipulator workspaces may be obtained.

In this chapter, a survey of Stewart platforms, their applications and existing methods for the determination of their workspaces are given. The different methods are illustrated by application to a simple example. Chapter 2 deals with an overview of, and extension to the original ray tracing optimization approach, which is further illustrated in chapter 3 by application to a generally constrained Stewart platform. A new approach, also based of the optimization principle, is described and illustrated in chapter 4. Finally a special case of redundant type behavior is treated in chapter 5.

1.2 Historical background and applications

In 1965 Stewart [5] proposed a six-degree of freedom platform, shown in Figure 1-1, as a design for a flight simulator. Parallel manipulators of similar design have subsequently often been referred to as 'Stewart platforms'.

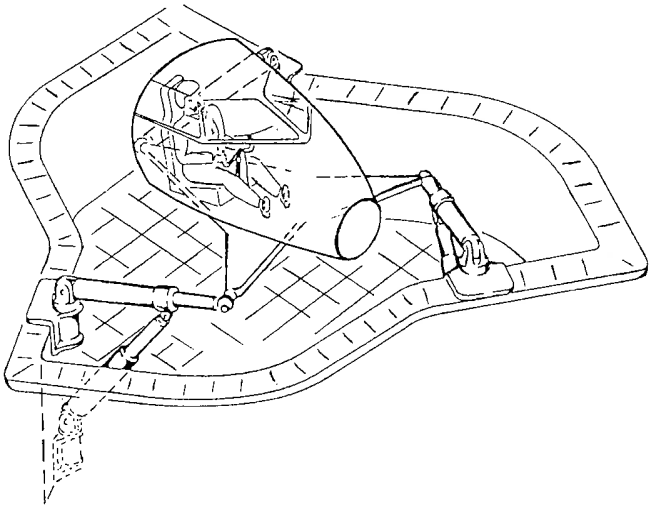

Figure 1-1: Stewart's flight simulator

Different parallel architectures have been proposed and studied by numerous authors. One of the most popular 6 degree of freedom spatial designs to emerge is the 6-3 configuration shown in Figure 1-2,which will be used here to illustrate the basic structure of Stewart platforms.

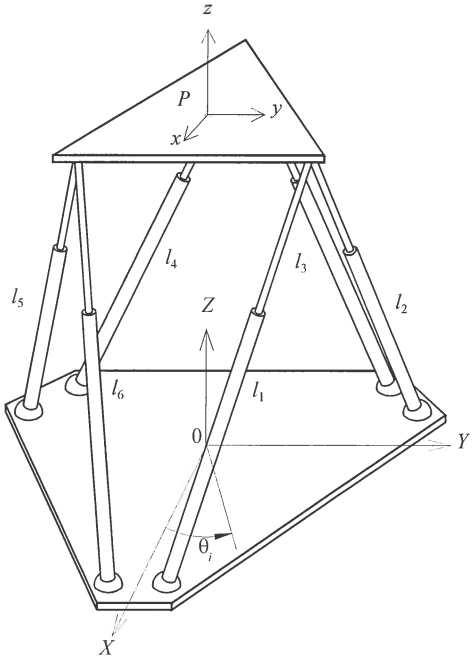

Figure 1-2: 6-3 spatial Stewart platform

The mechanism consists of a mobile platform connected to a fixed base by means of a number of linear actuators. The position and orientation of the platform can be controlled by varying

the lengths of the actuator legs. A point *P* fixed on the top platform, called the *working point*, is often chosen and used to describe the motion of the platform. This type of Stewart platform has also come to be known as a 'hexapod' due to the six legs used in this configuration. Many other types of parallel manipulators have been proposed; some using rotary actuators, for example the manipulator studied by Gosselin and Angeles [6], or specialized leg structures, such as the Turin manipulator studied by Ceccarelli [7].

The importance of Stewart platforms is that there are many more practical applications other than that originally envisaged by Stewart. Gough [5] had in 1949 already used a similar mechanism, shown in Figure 1-3, for the testing of tyres.

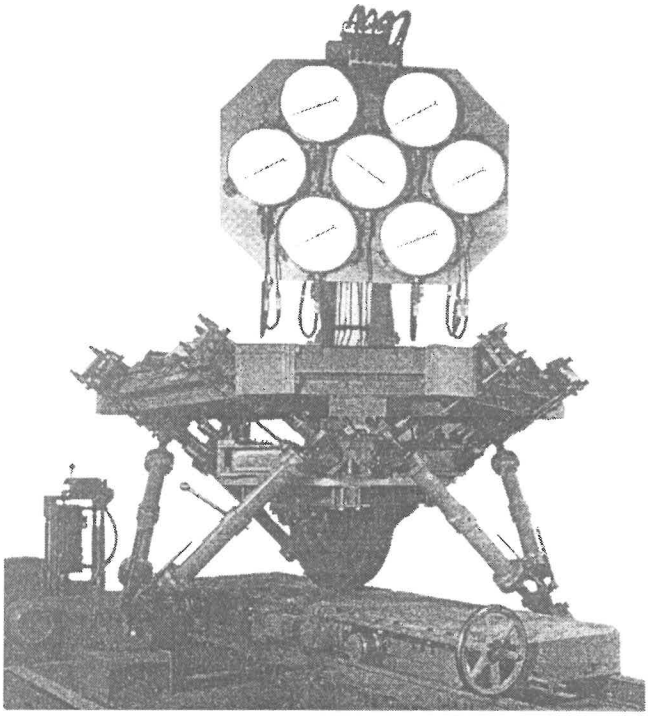

Figure 1-3: Gough's tyre test machine

A number of interesting prototypes based on the architecture of Stewart and Gough's platforms have been proposed and built. These parallel manipulators vary in size and application from a micromanipulator for ophthalmic surgery (Grace et al. [8]) to a manipulator for underground excavation (Arai et al. [9]). Other commercial uses include manipulators for assembly and pick-and-place applications (Sternheim [10]), and application of the parallel architecture to create six degree of freedom machining centers. Some examples of the latter are the Giddings and Lewis Variax Hexacenter [11] (shown in Figure 1-4), the Ingersoll Octahedral Hexapod [12] and the Hexel Tornado 2000 [13].

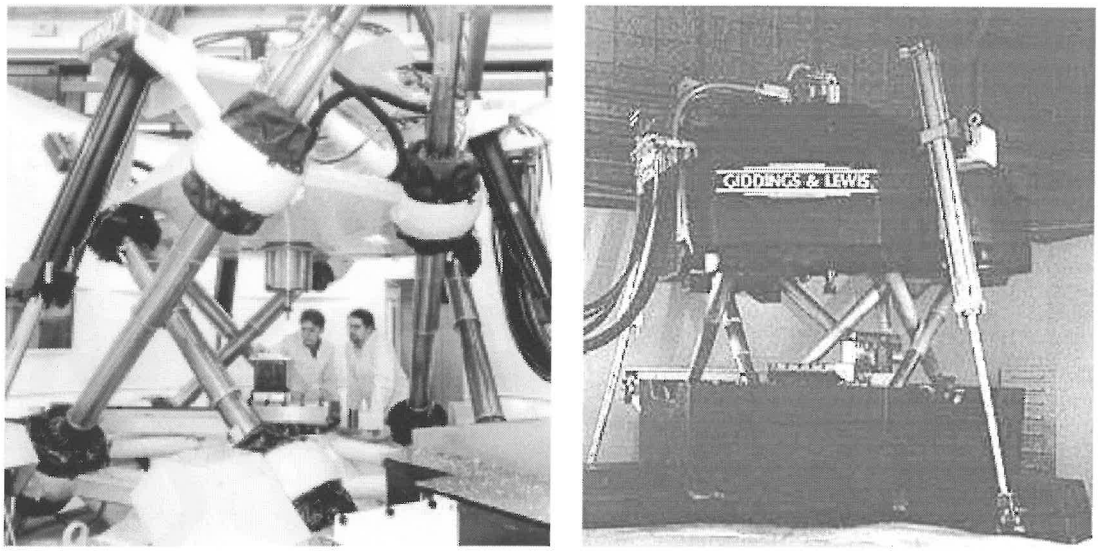

Figure 1-4: The Variax Hexacenter by Giddings & Lewis

The multitude of present applications are as a result of the advantages that parallel manipulators have over traditional serial manipulators. Since the load on the platform is approximately evenly distributed between the actuator legs, Stewart platforms have a high nominal load to weight ratio (Merlet [3]). The stresses in the legs are also mostly axial, which results in high platform rigidity. Parallel manipulators are often used in assembly tasks because of their high positioning accuracy. This accuracy is as a result of actuator errors being averaged at the working point, and not compounded as occurs with serial link manipulators. Furthermore the high stiffness of the manipulator contributes to the positioning accuracy because leg deformations are minimal. These properties are further discussed in the paper of Fichter and McDowell [4].

Although Stewart platforms possess a number of advantages, there are also disadvantages associated with them. Most notable is the tendency of parallel platforms to have small workspaces. The reason for this is that, in comparison to serial manipulators, the closed loops of the parallel manipulator impose additional kinematic constraints on the system. Gosselin [14] emphasizes that it is thus of primary importance to develop efficient tools for the determination of parallel manipulator workspaces. This is not an easy task since, for a parallel manipulator, the orientation and position of the working point are coupled in terms of the inputs to the manipulator. Existing methods for the determination of parallel manipulator workspaces are examined in the next subsection.

1.3 Existing methods for workspace determination

Depending on the orientation conditions imposed on the platform, different types of positional workspaces can be calculated. There is some inconsistency in the literature as to what the generally accepted terminology for naming different workspaces is. In the paper of Merlet et al. [15], various classes of manipulator workspaces for planar manipulators are defined and determined. Their terminology will be adopted in the work that follows. The workspaces described are:

1. *constant orientation workspace*: the set of points that can be reached by the working point of the platform with a specified constant orientation. Gosselin [14] also uses the term 'positioning workspace' to describe the constant orientation workspace.
2. *maximal workspace*: the set of points that can be reached by the working point of the platform with at least one orientation. The maximal workspace has also been referred to as the 'reachable workspace' (Kumar and Waldron [16], Wang and Hsieh [17]). The major part of the present study is devoted to the determination of maximal workspaces. In what follows, the term 'workspace' will therefore refer to 'maximal workspace' unless otherwise stated. In a similar way the term '*accessible output set*' has been used interchangeably with '*workspace*' by Haug et al. [18] and Snyman et al. [1] to refer to the maximal workspace.
3. *inclusive workspace*: the set of points that can be reached by the working point of the platform with at least one orientation in a prescribed range. This is a subset of the maximal workspace.
4. *dextrous workspace*: the set of points that can be reached by the working point of the platform for which all possible platform orientations can be attained. In other words the platform can assume any orientation for a prescribed position in the dextrous workspace.
5. *total orientation workspace*: the set of points that can be reached by the working point of the platform at which all orientations in a prescribed range can be attained. The term 'dextrous workspace' has also been used to describe the total orientation workspace (Haug et al. [18], Snyman et al. [1]).

Of the workspaces listed above the determination of the maximal workspace is probably the most challenging problem, since it represents the case with the fewest constraints present and thus the form of the workspace will generally be the most complex. Methods for workspace determination are usually primarily concerned with determining the boundary of the accessible output set of a manipulator. *Internal boundary curves* are also important features of

any workspace. These curves present possible barriers to the motion of the manipulator within the workspace (Haug et al. [19]).

Merlet [3,20] classifies the main methods for the determination of parallel manipulator workspaces. He identifies three main classes, namely *discretization methods*, *Jacobian matrix techniques* and the *geometrical approach*. In the discussion of these classes below, the simple planar two degree of freedom manipulator shown in Figure 1-5 will be used to illustrate each type of method.

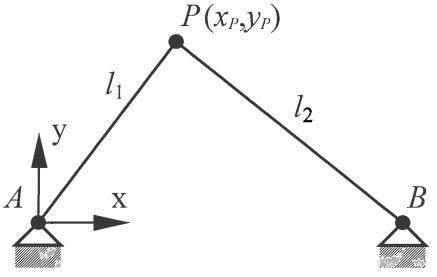

Figure 1-5: A two degree of freedom manipulator

The manipulator consists of two linear actuators, l_1 and l_2, connected to the ground by means of revolute joints A and B and to each other by a revolute joint P with global coordinates (x_P,y_P), which is the working point of the manipulator. Point A has coordinates (0,0) and point B coordinates $(x_B,0)$. With no limits on the actuator lengths, point P may be arbitrarily positioned in the x-y plane by controlling the lengths of legs 1 and 2. It is evident that this manipulator thus has two degrees of freedom. The motion of the manipulator is restricted when actuator legs have limits associated with them of the form

$$l_i^{\min} \le l_i \le l_i^{\max},\ i=1,2 \qquad \textbf{(1.1)}$$

These constraints, together with the geometry of the platform, determine the size and shape of the workspace. In this chapter, the values of the above constants will be taken as $x_B=4$, $l_1^{\min} = l_2^{\min} = 2.25$, $l_1^{\max} = 3.25$ and $l_2^{\max} = 3.75$ expressed in arbitrary units.

1.3.1 The discretization method

The discretization method in its most basic form (Yang and Lee [21], Chrisp and Gindy [22]) will be illustrated by its application to the two degree of freedom manipulator. As shown in Figure 1-6, a rectangular grid of points ($\{x_i,y_j\}$ $i=1,N$; $j=1,M$) is defined on the output plane at a given resolution. The working point P is then successively positioned at each of the grid node points and the leg length constraints checked to determine whether any of the constraints are violated. Each of these points can then be classified as either a feasible or an infeasible point. In such a way, an indication of the manipulator workspace can be obtained. Since this

approach is computationally intensive, various authors have used different methods to speed up the process.

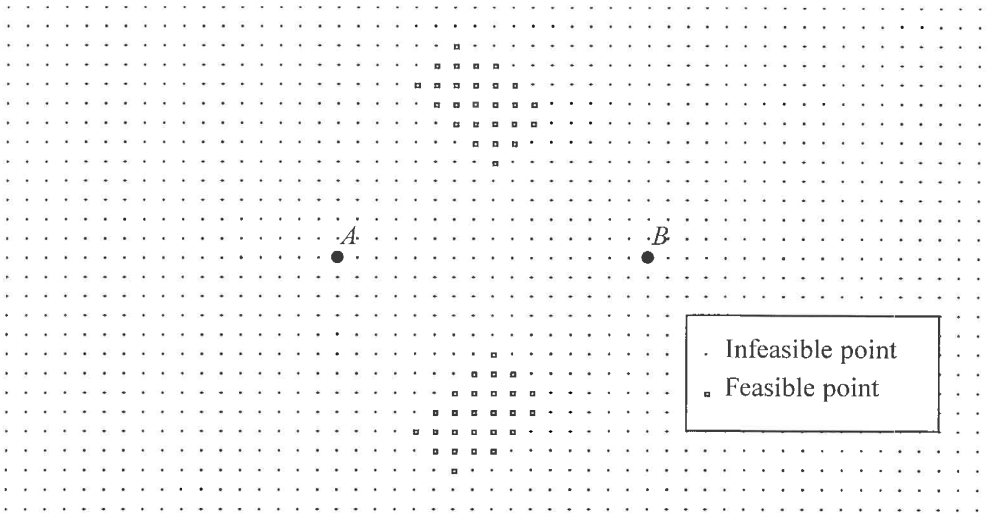

Figure 1-6: Workspace determination using the discretization method

Fichter [23] uses a more efficient contour-tracing method for determining the workspace of a six degree of freedom parallel manipulator. The method consists of fixing any four of the possible six spatial platform parameters (x, y, z, roll, pitch and yaw), thus defining a plane in which the workspace will be solved. The platform position is incremented with respect to the remaining two parameters and the inverse kinematics solved to give the leg lengths and platform configuration. Once more computed leg lengths are compared to the leg length limits of the manipulator to determine the feasibility of each incremented platform position. The contour-tracing algorithm is used to determine the next point to be sampled. A similar approach is used by Arai et al. [24] who also include the effects of limits on the passive joint angles and leg inferences in their analysis.

In a different approach Masory and Wang [25] and Chrisp and Gindy [22] have used relatively efficient searches in a cylindrical coordinate system to determine constant orientation workspaces of spatial Stewart platforms.

Merlet [20] states that discretization methods are usually not efficient and therefore time-consuming, especially when a high resolution mapping of the workspace is required. Furthermore it is difficult to vary more than two parameters simultaneously, which means that only constant orientation or constant position workspaces may be determined with ease.

Because of this, internal boundary curves cannot practicably be determined using the discretization method since these curves are associated with a range of platform orientations.

The discretization approach does however possess a number of advantages. Since the method consists of simply scanning and sampling the workspace, it is easy to implement as a code for determining manipulator workspaces. Additionally due to the fundamental nature of the algorithm the method is very stable.

1.3.2 The continuation method

Merlet [20] describes the second class of methods as *Jacobian matrix techniques*. Early application of this approach appears in the work of Jo and Haug [26,27] who refer to this class of methods as *continuation methods*. For any point on the boundary of a workspace the velocity of the manipulator along a normal to the boundary must be equal to zero. This implies that the Jacobian matrix of the manipulator kinematic constraints is row rank deficient on the boundary of the workspace.

For the example manipulator previously shown in Figure 1-5 the inverse kinematics relate the leg lengths to the working point position as follows:

$$\begin{aligned} l_1^2 &= x_P^2 + y_P^2 \\ l_2^2 &= (x_P - x_B)^2 + y_P^2 \end{aligned} \tag{1.2}$$

The leg length constraints (1.1) may be accounted for either by means of a slack variable approach or by introducing the transformation (Haug et al. [18])

$$l_i = l_i^0 + l_i^r \sin(l_i^*),\ i=1,2 \tag{1.3}$$

$$\text{where } l_i^0 = (l_i^{\min} + l_i^{\max})/2 \text{ and } l_i^r = (l_i^{\min} - l_i^{\max})/2$$

Defining $\boldsymbol{u}=[u_1,u_2]^T=[x_P,y_P]^T$ and $\boldsymbol{v}=[v_1,v_2]^T=[l_1^*,l_2^*]^T$ the generalized co-ordinates of the manipulator are given by $\boldsymbol{q}=[x_P,y_P,v_1,v_2]^T=[\boldsymbol{u}^T,\boldsymbol{v}^T]^T$. Using transformation (1.3), equations (1.2) can now be written as kinematic constraint equations

$$\boldsymbol{\Phi}(\boldsymbol{q}) = \begin{bmatrix} (l_1^0 + l_1^r \sin v_1)^2 - u_1^2 - u_2^2 \\ (l_2^0 + l_2^r \sin v_2)^2 - (u_1 - x_B)^2 - u_2^2 \end{bmatrix} = \boldsymbol{0} \tag{1.4}$$

The sub-Jacobian of $\boldsymbol{\Phi}(\boldsymbol{q})$ with respect to $\boldsymbol{v}$ is

$$\boldsymbol{\Phi}_v(\boldsymbol{q}) = \begin{bmatrix} 2(l_1^0 + l_1^r \sin v_1) l_1^r \cos v_1 & 0 \\ 0 & 2(l_2^0 + l_2^r \sin v_2) l_2^r \cos v_2 \end{bmatrix} \tag{1.5}$$

This matrix becomes rank deficient when $\cos v_1=0$ or when $\cos v_2=0$. When $\cos v_i=0$, $v_i = \pm\frac{(2j+1)\pi}{2}$, j=0,1,2,… Without loss of generality, consider only the cases where $v_i = \pm\frac{\pi}{2}$. Substituting these values for v_i into transformation **(1.3)** yields a total of four solution cases: $l_i = l_i^0 + l_i^r(1) = l_i^{\max}$ or $l_i = l_i^0 + l_i^r(-1) = l_i^{\min}$, i=1,2.

Consider the particular case where $l_1 = l_1^{\min}$ (i.e. $v_1 = -\frac{\pi}{2}$). Then, for a prescribed value of v_2 in the range [-π/2, π/2], the set of nonlinear equations contained in the kinematic constraints **(1.4)** have two analytical solutions to the pair (u_1,u_2):

$$u_1(v_2) = \frac{x_B + \left(l_1^{\min}\right)^2 - \left(l_2(v_2)\right)^2}{2x_B}$$
$$u_2(v_2) = \pm\sqrt{\left(l_1^{\min}\right)^2 - \left(u_1(v_2)\right)^2} \qquad \textbf{(1.6)}$$
$$\text{where } l_2(v_2) = l_2^0 + l_2^r \sin(v_2)$$

Thus letting v_2 vary between -π/2 and π/2 the loci of the two solution points (u_1, u_2) given by **(1.6)** correspond to curve 1 and curve 1' of Figure 1-7, depending on the sign chosen in **(1.6)**.

The solution curves for the cases where $l_1 = l_1^{\max}$, $l_2 = l_2^{\min}$ and $l_2 = l_2^{\max}$ may be determined in a similar manner. They are shown in Figure 1-7 as curves 2(2'), 3(3') and 4(4') respectively. There are no internal boundary curves.

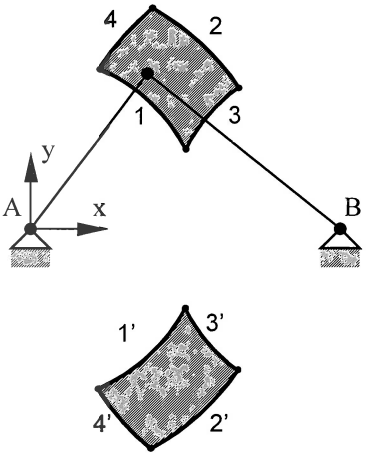

Figure 1-7: Workspace determination using the continuation method

The above analytical solution is only possible for simple problems. For more complex problems a *numerical procedure* is needed to trace the workspace boundaries. Jo and Haug

[26,27] use a *continuation method* in which the Jacobian rank deficiency condition is combined with the kinematic constraints to form a set of non-linear equations defining internal and external workspace boundaries. By successively varying only one input variable at a time the solution of the problem becomes much easier. It is then possible to map the solution curves of the nonlinear equation set, corresponding to different parts of the boundary, using a continuation method. Note that in the example given above, had a numerical procedure been used to solve equations (1.4), then either the top or bottom portions of the workspace would have been determined, depending on the chosen starting point of the procedure. The method is extensively described in the papers of Haug et al. [18, 28].

The main criticisms that Merlet [20] has of continuation methods is that they are difficult to implement and do not conveniently and readily allow for the introduction of other physical constraints of practical importance. Wang and Hsieh [17] state that since it is necessary to have explicit expressions for the constraint equations and Jacobian matrix of the mechanism, the method may become complicated for a generally constrained parallel manipulator.

The advantage that the continuation method possesses over other methods is that it is a broadly applicable numerical algorithm, which can be used for mapping the boundaries of both serial and parallel manipulator workspaces.

1.3.3 The geometrical method

The geometrical method applied to a planar platform was first proposed by Gosselin and Angeles [6] and has been further refined and extended by Merlet et al. [15]. For maximal workspace determination of the manipulators considered here, the method is based on the fact that the boundary of the workspace is attained whenever at least one of the actuators reaches an extreme length.

As an illustration of the method, the workspace of the example manipulator (shown in Figure 1-5) is calculated below using the geometrical algorithm. Annular regions limited by the maximum and minimum leg lengths of leg i are denoted ε_i (See Figure 1-8). For this two degree of freedom manipulator the workspace boundary will occur when at least one of the two actuators is at an extreme length. There are four possible cases: (1) $l_1 = l_1^{\min}$, (2) $l_1 = l_1^{\max}$, (3) $l_2 = l_2^{\min}$ and (4) $l_2 = l_2^{\max}$. Consider the case where $l_1 = l_1^{\min}$. The points of intersection of the annular region ε_2 and a circle, centered at A with radius $l_1^{\min}$, are determined. These points

of intersection are shown in Figure 1-8 as points a, b, c and d. Four arcs; ab, bc, cd and da are identified. The feasibility of the manipulator configuration corresponding to any arc is determined by examining the configuration of the manipulator at the mid-point of that arc. If the other leg length is within the prescribed limits then the configuration is acceptable. For this case, it is evident that arcs ab and cd are feasible and the others are not.

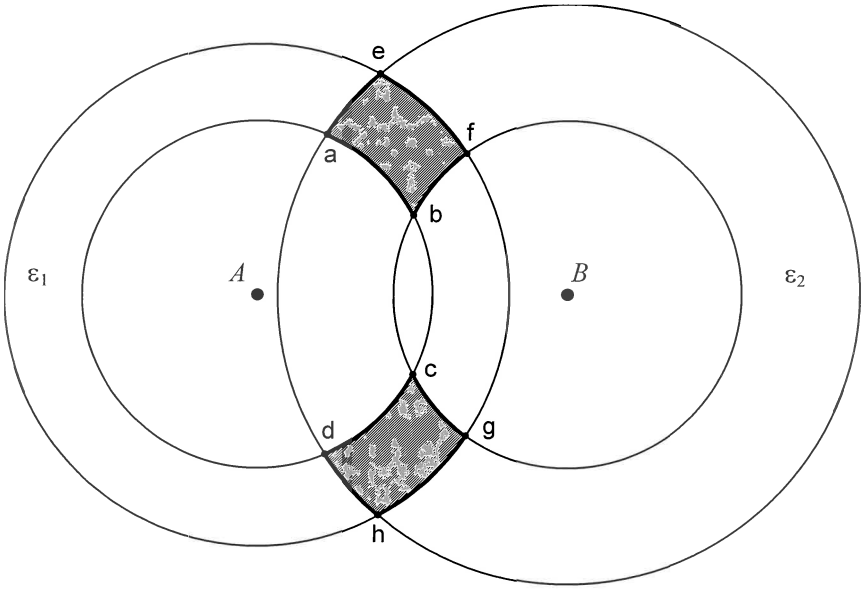

Figure 1-8: Workspace determination using the geometrical method

Similarly for case 2 feasible arcs ef and gh are identified, for case 3 the feasible arcs are gc and bf and for case 4 the arcs hd and ae. In this simple example, all of these arcs correspond to a part of the workspace boundary so the procedure is complete. The workspace of the manipulator is indicated by the hatched region in Figure 1-8.

For a general planar three degree of freedom manipulator such as that shown in Figure 1-9 an extreme platform extension can occur either when one, or two legs are at an extreme length.

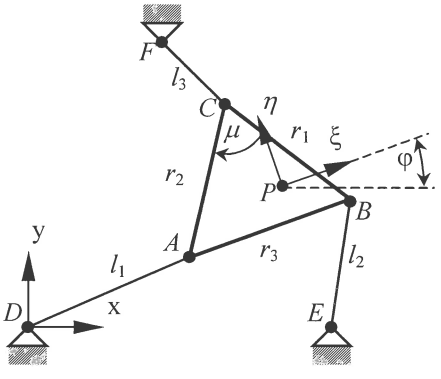

Figure 1-9: A three degree of freedom manipulator

As discussed below, the workspace boundary therefore consists of circular arcs and portions of four bar coupler curves.

Boundary points with one extreme leg length occur when the working point of the manipulator lies along the line of action of the actuator.

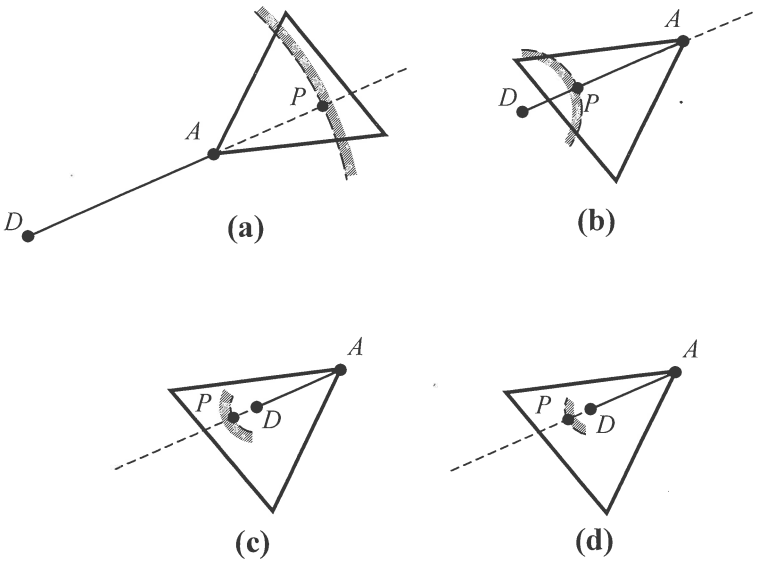

Figure 1-10: Platform-leg configurations for boundary points with one extreme leg length

For each leg, as shown for leg 1 in Figure 1-10, this condition leads to four possible combinations of leg extension and platform orientation, which may correspond to the workspace boundary. (a) $l=l^{max}$ and *DAP* aligned in this order. (b) $l=l^{min}$ and *DPA* aligned in this order. (c) $l=l^{min}$ and *PDA* aligned in this order. (d) $l=l^{max}$ and *PDA* aligned in this order. In each case, the hatched side of the section of workspace boundary shown is an infeasible region.

Considering one actuator leg at a time, and checking all possible configurations using a procedure similar to the one used in the two degree of freedom example above, feasible arcs, which correspond to internal and outer workspace boundaries, can be identified.

For boundaries with two extreme leg lengths fixed, the trajectory of the working point corresponds to the coupler curve of an equivalent four-bar mechanism. Since there are three legs, each with two possible extreme lengths, there are a total of twelve possible combinations of paired extreme leg lengths to be considered. The coupler curves corresponding to each these configurations are determined using the theory of four bar mechanisms. The next step is to calculate the points of intersection between the coupler curves, the annular regions ε_j and the circular arcs determined in the first step. The critical and limit points of the coupler curves are also determined. Once the coupler curves have been split into *arcs* of coupler curves between the calculated points of intersection, feasible arcs are identified as before.

The final step in the method is to identify which of the feasible arcs determined above are components of the workspace boundary. An arc is a component of the workspace boundary if motion normal to the arc in one direction leads to constraint violations, and motion in the opposing direction is feasible. To test for this condition the *Jacobian matrix* of the constraint equations ($l_i^2 = l_i^2(x, y, \varphi)$) as well as two normal vectors to the arc, pointing in opposing directions, are determined at a point on the arc. The leg velocities are then calculated for a platform velocity directed along each of these normal vectors. The sign of these joint velocities can be used in conjunction with the configuration of the manipulator to determine whether or not the arc is a component of the workspace boundary. For example, if for an arc corresponding to maximal leg lengths of legs 1 and 2, the leg velocities $\dot{l}_1, \dot{l}_2$ are both positive for motion directed in one normal direction, and negative for motion in the other, then the arc is part of the workspace boundary.

Gosselin [14] has extended the geometrical method to allow for the determination of six degree of freedom parallel manipulator workspaces, but only considers the extreme leg length in his approach. Merlet [29] includes the passive joint and leg interference constraints in his application of the geometrical method to a spatial Stewart platform.

The geometrical method is in essence an evaluation of possible manipulator extreme positions, the configurations of which follow from an analysis such as that of Jo and Haug [26,27], to determine which of these correspond to the workspace boundary. By a modification of the basic algorithm (Merlet et al. [15]) the geometric principle may also be applied to the calculation of constant orientation, inclusive maximal, dextrous and total orientation workspaces. Furthermore the geometrical method allows for the inclusion of the effects of limits on the passive joint angles as well as leg interference. The main drawback of the geometrical method is that it requires the calculation of the Jacobian matrix of the system to determine which arcs belong to the workspace boundary. For more complicated manipulators this may be a very difficult task.

1.3.4 Other methods

In addition to the mainstream methods classified by Merlet [20] other authors have proposed different special techniques for workspace determination.

1.3.4.1 The method of Bajpai and Roth

Bajpai and Roth [30] apply an essentially geometrical method to the determination of workspaces of a specific class of two degree of freedom parallel manipulator shown in Figure 1-11(a).

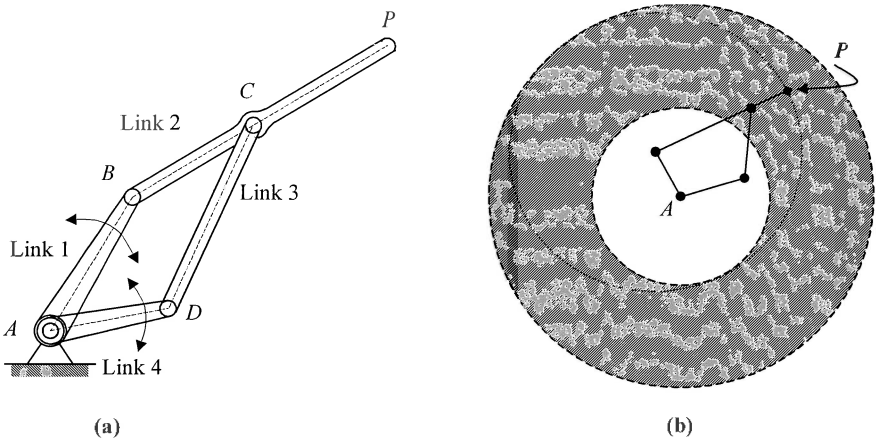

Figure 1-11: The method of Bajpai and Roth (a) a two degree of freedom manipulator and (b) the workspace of the manipulator

The method consists of fixing one input and examining the end effector motion caused by varying the other input. See for example Figure 1-11(b) where the dotted line depicts the path of working point *P* for a fixed orientation of link 1, and the workspace of the manipulator is indicated by the shaded region. By applying this and other conditions in an ad hoc manner the workspace of the manipulator for different link lengths can be determined.

1.3.4.2 The method of Kumar and Waldron

The method of Kumar and Waldron [16], later refined and applied to parallel manipulators by Kumar [31], is based on the fact that the application of a pure force on the working point drives a manipulator into an extreme position in the direction of the applied force. Depending on the configuration of the mechanism, these may correspond to internal or external boundaries of the workspace. The method thus requires mapping boundary curves corresponding to all possible special configurations. Figure 1-12 shows a two degree of freedom manipulator studied by Kumar. Special configurations of the manipulator and the boundary curves associated with each of the special configurations are given in Figure 1-13(a) and (b) respectively. The method is formulated using screw theory.

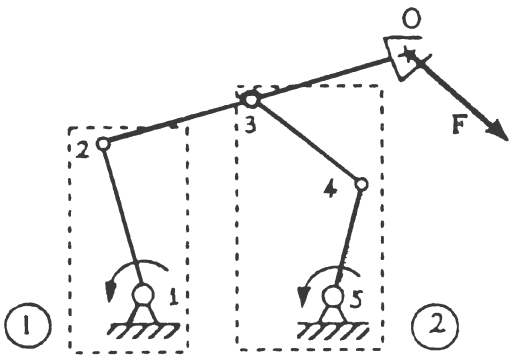

Figure 1-12: Two degree of freedom manipulator studied by Kumar

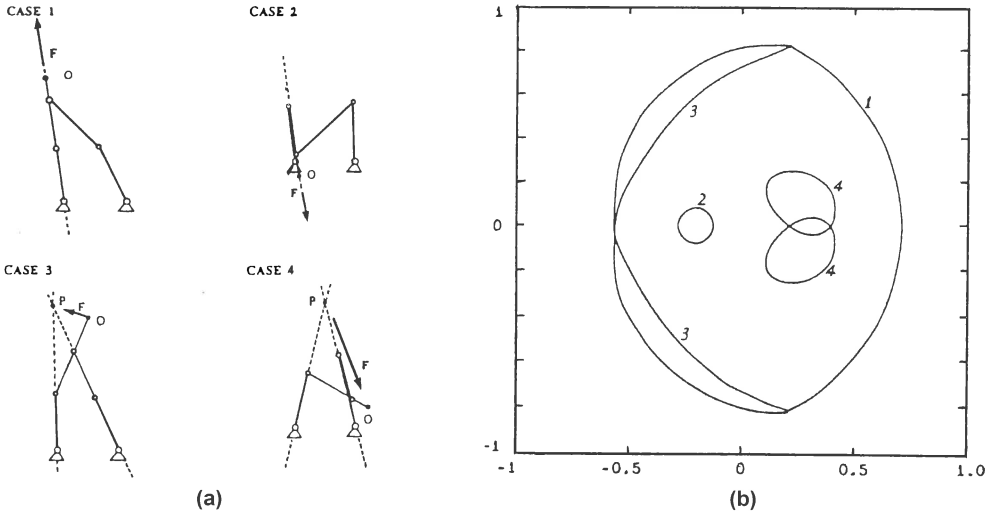

Figure 1-13: The method of Kumar and Waldron: (a) Special configurations and (b) the workspace of the manipulator

Kumar [31] also applies the method to the determination of the dextrous and total orientation workspaces of a three degree of freedom manipulator and a spatial PUMA robot.

1.3.4.3 The method of Wang and Hsieh

Wang and Hsieh [17] present a numerical method based on an optimization approach. They formulate an optimization problem in order to find the extreme reach of a six degree of freedom parallel manipulator with rotary actuated joints. Considering successive horizontal slices through the workspace, as shown in Figure 1-14, and successive search directions emanating from a fixed base point, the distance between the end effector and the fixed base point is maximized or minimized depending on the desired objective. A multi-start procedure is used for each search to ensure that the global minimum or maximum, corresponding to the actual workspace boundary, and not an internal boundary, is found.

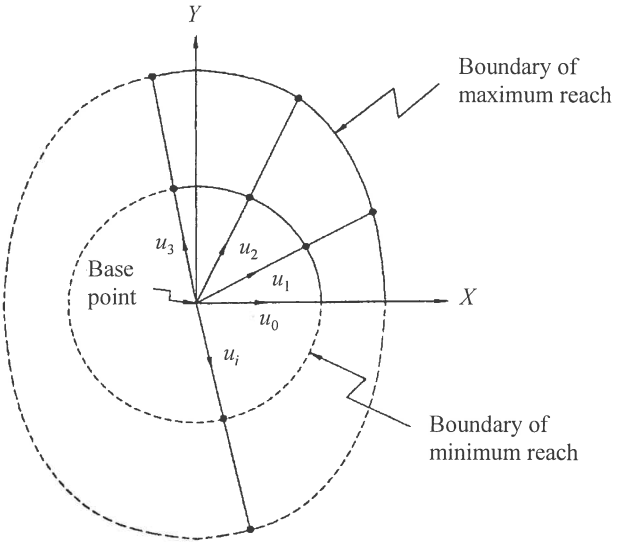

Figure 1-14: The method of Wang and Hseih: a horizontal slice through the workspace

1.4 Purpose of the present study

The current consensus amongst a number of workers in the field, for example Merlet [20], Gosselin [14] and Chrisp and Gindy [22], seems to be that the *geometrical method* represents the state-of-the-art technique for workspace determination. However, the geometrical method does not represent a truly general approach for the determination of workspaces of manipulators of varied design. The *continuation method* represents an approach that is more generally applicable to various manipulators, both serial and parallel. This method, however, does not readily allow for the introduction of all possible constraints. Furthermore, both of the above methods require the calculation of the Jacobian matrix, which may be a difficult task. In particular the continuation method appears to be a complicated method and not easily implementable. The challenge therefore still exists to find a method that is both general and allows for the easy application of constraints. Snyman et al. [1] and Wang and Hsieh [17] have independently proposed different numerical *optimization approaches* that handle constraints in a systematic way and do not require the computation of the Jacobian matrix. The main disadvantage of the approach of Snyman et al. is that, as it is currently formulated, it cannot in general be used to calculate workspaces that are *non-convex*.

The main purpose of this study is to extend the optimization approach of Snyman et al. [1] to additionally determine any non-convex workspaces that occur in the cases of planar manipulators of more general design. A second, and almost equally important objective, is to adapt the optimization approach to handle any additional physical constraints limiting the workspaces of planar parallel manipulators.

Other special problems that occur in the implementation will also be addressed. For example, when the working point of the manipulator is chosen at one of the leg attachment points, the manipulator exhibits an interesting redundant behavior along certain portions of the workspace boundary. This phenomenon will be investigated with the purpose of introducing a formulation that will allow for the determination of these so-called redundant workspaces. Finally the validity of the optimization method as a truly general approach for the determination of manipulator workspaces will be evaluated.

Chapter 2: THE RAY METHOD FOR THE DETERMINATION OF PLANAR PARALLEL MANIPULATOR WORKSPACES

2.1 Overview of the existing ray method for workspace determination

This chapter presents an extension to the novel optimization approach, proposed by Snyman et al. [1] and Du Plessis [2], for the determination of workspaces of planar manipulators. The work of these authors was motivated by the foundation paper of Haug et al. [18]. In their paper, Snyman et al. [1] successfully apply the optimization methodology to the determination of workspaces of manipulators studied by Haug et al. As an evaluation of the method's generality the workspaces of both a planar Stewart platform and a planar redundantly controlled serial manipulator are determined. It is also shown that the optimization approach is capable of determining interior boundaries of the workspaces. To facilitate the discussion of results a new and concise method for labelling bifurcation points and curves is proposed[1]. The workspace of the planar redundant serial manipulator determined by Snyman et al. [1] using the optimization approach is shown in Figure 2-1.

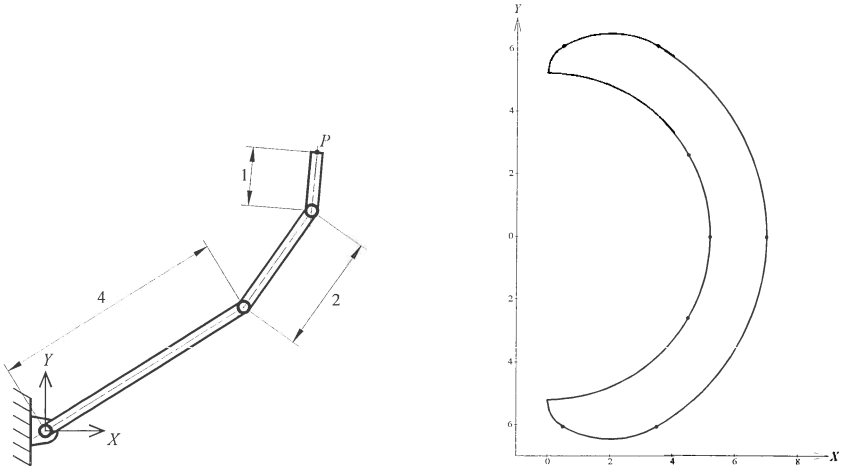

Figure 2-1: Planar serial manipulator and maximal workspace

Du Plessis [2] extends the optimization approach to determine the constant orientation and total orientation workspaces of the planar Stewart platform considered by Snyman et al. [1].

[1] This proposed method for bifurcation labelling is given in section 4.2.5.

Using the methodology for mapping spatial workspaces proposed by Snyman et al. [1] the constant orientation and total orientation workspaces of the 6-3 Stewart platform mentioned in section 1.2 and shown in Figure 1-2 have been determined. Some of the results obtained by Du Plessis [2] are given in Figure 2-2.

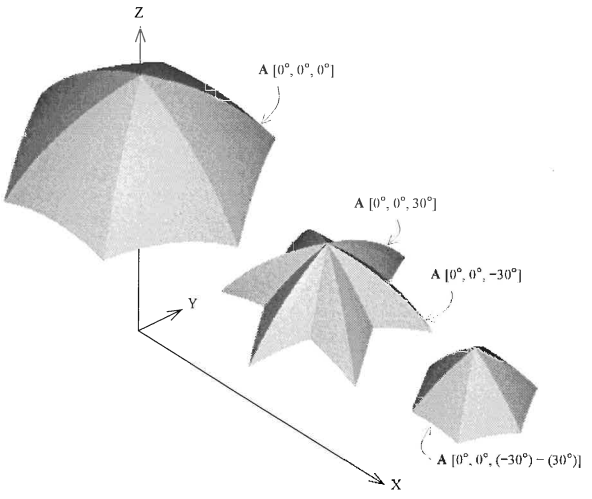

Figure 2-2: Constant orientation and total orientation workspaces of the 6-3 Stewart platform

In this chapter the work of Snyman and his co-authors is extended and generalized to enable the determination of non-convex planar workspaces. As a preliminary to the presentation of the general method, accessible output sets for manipulators are defined and criteria for determining their boundaries are stated. Based on the definition of the boundary of the accessible output set, a method for mapping the boundary of a planar non-convex workspace is developed. Simply stated, the method consists of finding a suitable initial radiating point, and then finding the points of intersection of a pencil of rays emanating from this point with the boundary of the accessible set. The points of intersection are determined by means of an optimization approach in which the proven robust dynamic constrained algorithm of Snyman [32,33,34] and Snyman et al. [35] is used[2]. If any segment of the workspace boundary cannot be determined due to non-convexity, then the missing segment is mapped using a suitably chosen new radiating point.

2.2 Description of the method

2.2.1 Coordinates

As described by Haug et al. [28], *generalized coordinates* $\boldsymbol{q} = [q_1, \ldots, q_{nq}]^T \in R^{nq}$ are defined that characterize the position and orientation of each body in the mechanism. In the vicinity of

[2] A brief description of Snyman's optimization algorithm is given in Appendix B.

an *assembled configuration* of the mechanism, these generalized coordinates satisfy m independent holonomic kinematic constraint equations of the form

$$\boldsymbol{\Phi}(\boldsymbol{q}) = \mathbf{0} \tag{2.1}$$

where $\boldsymbol{\Phi} : R^{nq} \rightarrow R^{m}$ is a smooth function.

Mechanisms are usually designed to produce a desired functionality. Specifying the values of a selected subset of the generalized coordinates, called the *input coordinates*, defines the motion of the mechanism. These input coordinate values are controlled by external influences with the intent of prescribing the motion of the mechanism. The vector of input coordinates is denoted by $\boldsymbol{v} = [v_1, \ldots, v_{nv}]^T$.

To characterize the functionality of the mechanism some measure of output, which is controlled by the mechanism inputs, must be monitored. *Output coordinates* are the subset of the mechanism's generalized coordinates that define the useful functionality of the mechanism. Output coordinates are distinct from input coordinates and are denoted by $\boldsymbol{u} = [u_1, \ldots, u_{nu}]^T$. A choice of input and output coordinates for a mechanism defines a mechanical system with an intended function. This mechanism is then called a manipulator.

Generalized coordinates of a mechanism that are neither input coordinates nor output coordinates are called intermediate coordinates, denoted by $\boldsymbol{w} = [w_1, \ldots, w_{nw}]^T$, where *nw=nq-nv-nu*.

2.2.2 Constraints and the accessible output set

Inequality constraints are often imposed on the input variables and may also apply to the intermediate variables. These respectively take the forms

$$\boldsymbol{v}^{\min} \leq \boldsymbol{v} \leq \boldsymbol{v}^{\max} \tag{2.2}$$

and

$$\boldsymbol{w}^{\min} \leq \boldsymbol{w} \leq \boldsymbol{w}^{\max} \tag{2.3}$$

There may also be additional inequality constraints acting on the system, representing relationships between the input, output and intermediate coordinates, that must be satisfied and which take on the general form

$$\boldsymbol{g}^{\min} \leq \boldsymbol{g}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) \leq \boldsymbol{g}^{\max} \tag{2.4}$$

The *accessible output set* of the manipulator is the collection of all possible output coordinates of the manipulator. To present this more precisely, the generalized coordinates are partitioned as follows:

$$\boldsymbol{q} = [\boldsymbol{u}^T, \boldsymbol{v}^T, \boldsymbol{w}^T]^T \tag{2.5}$$

The constraint equations (2.1) may be rewritten in terms of this partitioning of generalized coordinates:

$$\boldsymbol{\Phi}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = \boldsymbol{0} \tag{2.6}$$

The accessible output set $\boldsymbol{A}$ is defined as:

$$\begin{aligned}\boldsymbol{A} \equiv \{\, &\boldsymbol{u} \in R^{mu} : \boldsymbol{\Phi}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = \boldsymbol{0}\,;\ \boldsymbol{v} \text{ satisfying (2.2)};\ \boldsymbol{w} \text{ satisfying (2.3)};\\ &\boldsymbol{g}(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}) \text{ satisfying (2.4)}\}\end{aligned} \tag{2.7}$$

The boundary $\partial\boldsymbol{A}$ of the accessible output set may then be defined as:

$$\begin{aligned}\partial\boldsymbol{A} \equiv \{\, &\boldsymbol{u} \in R^{mu} : \boldsymbol{u} \in \boldsymbol{A} \text{ and } \exists \text{ an } \boldsymbol{s} \in R^{mu} \text{ such that for } \boldsymbol{u}' = \boldsymbol{u} + \lambda\boldsymbol{s},\\ &\lambda \in R \text{ arbitrarily small and either positive or negative, no } \boldsymbol{v} \text{ and } \boldsymbol{w} \text{ exist that}\\ &\text{satisfy } \boldsymbol{\Phi}(\boldsymbol{u}', \boldsymbol{v}, \boldsymbol{w}) = \boldsymbol{0} \text{ as well as inequalities (2.2), (2.3) and (2.4)}\}\end{aligned} \tag{2.8}$$

The method for determining the accessible workspace boundary of $\boldsymbol{A}$ is described in the next two sections, and is similar to that proposed by Snyman et al. [1]

2.2.3 Finding a point on $\partial\boldsymbol{A}$

With respect to the system of equations (2.1) and (2.6), a distinction can be made between two possibilities:

Case (i): where $m = nv$ and, given $\boldsymbol{u}$ and $\boldsymbol{w}$, system (2.6) may easily be solved to give $\boldsymbol{v}$ in terms of $\boldsymbol{u}$ and $\boldsymbol{w}$:

$$\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{u}, \boldsymbol{w}) \tag{2.9}$$

This is typically the situation with parallel manipulators, where the inverse kinematics can easily be solved.

Case (ii): where $m = nu$ and, given $\boldsymbol{v}$ and $\boldsymbol{w}$, system (2.6) may easily be solved to give $\boldsymbol{u}$ in terms of $\boldsymbol{v}$ and $\boldsymbol{w}$:

$$\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{v}, \boldsymbol{w}) \tag{2.10}$$

This is typically the situation with serial manipulators, where the forward kinematics is relatively easy to solve.

Although the application of the method in this study is restricted to parallel manipulators, i.e. Case (i) only, the general approach is nevertheless presented for both cases (i) and (ii).

Consider *Case(i)*. Assume that a radiating point $\boldsymbol{u}^0$ has been chosen and that it is interior to the accessible set, $\boldsymbol{A}$. Consistent with the definition of $\partial\boldsymbol{A}$ in **(2.8)**, a point $\boldsymbol{u}^b$ on the boundary in the direction $\boldsymbol{s} \in R^{nu}$ from $\boldsymbol{u}^0$ is determined by solving the following constrained optimization problem:

Problem (i):

$$\underset{\boldsymbol{u},\boldsymbol{w}}{\text{maximize}} \left\| \boldsymbol{u} - \boldsymbol{u}^0 \right\|$$

$$\text{such that } \boldsymbol{v}^{\min} \le \boldsymbol{v}(\boldsymbol{u},\boldsymbol{w}) \le \boldsymbol{v}^{\max}$$

$$\boldsymbol{w}^{\min} \le \boldsymbol{w} \le \boldsymbol{w}^{\max}$$

$$\text{and } \boldsymbol{g}^{\min} \le \boldsymbol{g}(\boldsymbol{u},\boldsymbol{v}(\boldsymbol{u},\boldsymbol{w}),\boldsymbol{w}) \le \boldsymbol{g}^{\max}$$

and subject to equality constraints

$$\boldsymbol{h}(\boldsymbol{u},\boldsymbol{s}) = \boldsymbol{0},\ \boldsymbol{h} \in R^{nu-1}$$

where $\|\cdot\|$ denotes the Euclidean norm. The equality constraints define a point on the parameterized straight line $\boldsymbol{u}(\lambda) = \boldsymbol{u}^0 + \lambda\boldsymbol{s}$, $\lambda \in R$. (For example, if $nu = 2$, $\boldsymbol{u} = (x, y)^T$, $\boldsymbol{u}^0 = (x^0, y^0)^T$ and $\boldsymbol{s} = (s_x, s_y)^T$, then $\boldsymbol{u} = \boldsymbol{u}^0 + \lambda\boldsymbol{s}$ has the components $x = x^0 + \lambda s_x$ and $y = y^0 + \lambda s_y$; it follows that $h(\boldsymbol{u},\boldsymbol{s}) = (x-x^0)/s_x + (y-y^0)/s_y = 0$. See Figure 2-3).

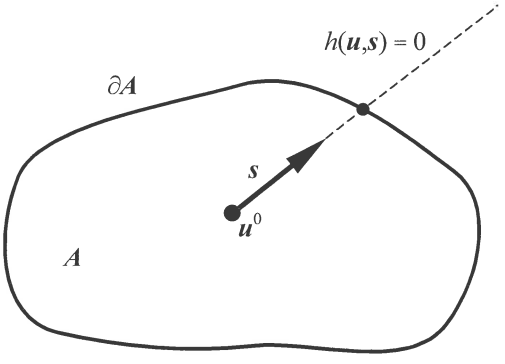

Figure 2-3: Ray in A to ∂A

For *Case (ii)* the associated constrained optimization problem is:

Problem (ii):

$$\underset{\boldsymbol{v},\boldsymbol{w}}{\text{maximize}} \left\| \boldsymbol{u}(\boldsymbol{v},\boldsymbol{w}) - \boldsymbol{u}^0 \right\|$$

$$\text{such that } \boldsymbol{v}^{\min} \le \boldsymbol{v} \le \boldsymbol{v}^{\max}$$

$$\boldsymbol{w}^{\min} \le \boldsymbol{w} \le \boldsymbol{w}^{\max}$$

$$\text{and } \boldsymbol{g}^{\min} \le \boldsymbol{g}(\boldsymbol{u}(\boldsymbol{v},\boldsymbol{w}),\boldsymbol{v},\boldsymbol{w}) \le \boldsymbol{g}^{\max}$$

and subject to equality constraints

$$\boldsymbol{h}(\boldsymbol{u}(\boldsymbol{v},\boldsymbol{w}),\boldsymbol{s}) = \boldsymbol{0},\ \boldsymbol{h} \in R^{nu-1}$$

where the equality constraints again define a point $\boldsymbol{u}$ on the straight line through $\boldsymbol{u}^0$ in the direction $\boldsymbol{s}$.

Note that should the radiating point $\boldsymbol{u}^0$ be chosen exterior to $\boldsymbol{A}$, then the above problems will become *minimization* problems.

2.2.4 Basic methodology for mapping the boundary of a planar accessible set

Assume a planar manipulator with a two-dimensional accessible set $\boldsymbol{A}$, and also for the moment that $\boldsymbol{A}$ is convex, which will not always be the case. Also assume that the radiating point $\boldsymbol{u}^0$ is an interior point as shown in Figure 2-4.

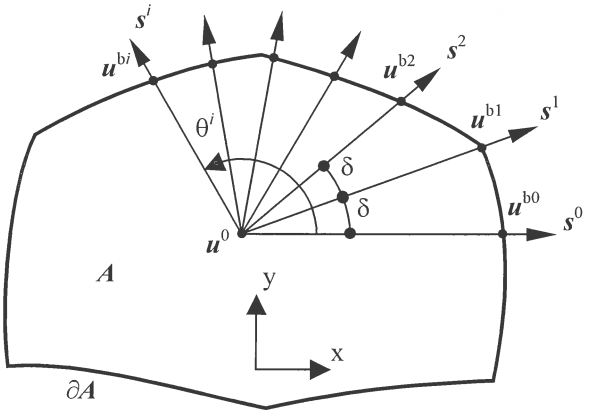

Figure 2-4: Numerical mapping of ∂A

The boundary $\partial \boldsymbol{A}$ may now be numerically mapped by successively solving *Problem (i)* or *(ii)* for successive rays, emanating at regular angular intervals δ from $\boldsymbol{u}^0 = (x^0, y^0)^T$, with respective directions $\boldsymbol{s}^i$, $i = 0,1,2,\ldots,N$. It is assumed here that the solution to the optimization problem for the first ray yields a point $\boldsymbol{u}^{b0}$ on the external workspace boundary (see Figure 2-4). Thereafter the starting point chosen for each successive optimization problem is the *solution to the previous problem*. This is done to ensure that solutions to consecutive optimization problems trace the same external workspace boundary. If, instead of this approach, the *radiating point* was used as the starting point for each optimization problem, the algorithm would sometimes find local interior solutions to the optimization problem, corresponding to internal workspace boundaries, instead of the global solution corresponding to the outer workspace boundary. The angle θ is measured in the right hand sense from the x-

axis. Each vector $\boldsymbol{s}^i$ thus lies at an angle θ^i to the x-axis. The mapping begins at an angle θ^0 corresponding to vector $\boldsymbol{s}^0$ and sweeps through an angle $N\delta$. The mapping process is depicted in Figure 2-4. For this case $\theta^0=0$.

Two inter-related questions arise in connection with the implementation of the methodology. The first relates to how the radiating point may be obtained. Depending on the particular geometry of each case, a suitable choice for $\boldsymbol{u}^0$ may be self-evident. If not, it is suggested that, for *Case (i)*, $\boldsymbol{u}^0$ may be obtained from **(2.9)** by solving for $\boldsymbol{u}$ in:

$$\overline{\boldsymbol{v}} = \boldsymbol{v}(\boldsymbol{u}, \overline{\boldsymbol{w}}) \tag{2.11}$$

where

$$\overline{\boldsymbol{v}} = (\boldsymbol{v}^{\min} + \boldsymbol{v}^{\max})/2$$

$$\overline{\boldsymbol{w}} = (\boldsymbol{w}^{\min} + \boldsymbol{w}^{\max})/2$$

In practice this can be done by solving the least squares optimization problem

$$\underset{\boldsymbol{u}}{\text{minimize}} \left\| \boldsymbol{v}(\boldsymbol{u}, \overline{\boldsymbol{w}}) - \overline{\boldsymbol{v}} \right\|^2 \tag{2.12}$$

For *Case (ii)*, if an obvious choice for $\boldsymbol{u}^0$ is not available, then an indication may be obtained from **(2.10)**:

$$\boldsymbol{u}^0 = \boldsymbol{u}(\overline{\boldsymbol{v}}, \overline{\boldsymbol{w}}) \tag{2.13}$$

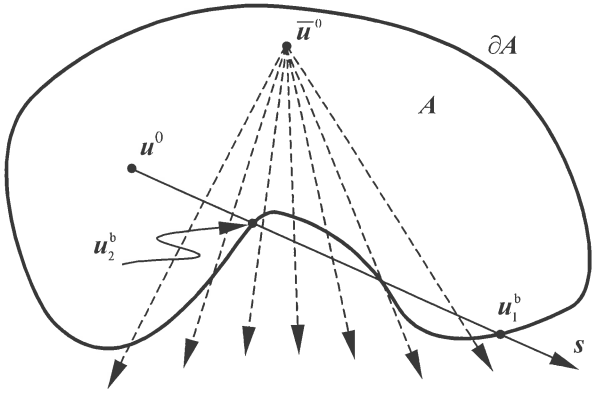

Figure 2-5: Complication if A is non-convex

The second question concerns the strategy to be adopted if non-convexity of $\boldsymbol{A}$ interferes with the mapping of the workspace boundary, as shown in Figure 2-5. It can be seen that, as a result of non-convexity and the particular positioning of $\boldsymbol{u}^0$, two boundary points $\boldsymbol{u}_1^b$ and $\boldsymbol{u}_2^b$ are possible.

A distinction can now be made between two types of non-convexity:

Type (i) Non-Convexity: A workspace exhibits type (i) non-convexity if there exists a radiating point $\overline{\boldsymbol{u}}^0 \in R$ internal to $\boldsymbol{A}$, such that there is a unique solution to *Problem (i)* or *(ii)* (as applicable) for *all* possible search direction $\boldsymbol{s}^i$ radiating from $\overline{\boldsymbol{u}}^0$. Figure 2-5 shows an example of type (i) non-convexity. Type (i) non-convexity can be dealt with by simply choosing a suitable radiating point and applying the methodology described above.

Type (ii) Non-Convexity: A workspace exhibits type (ii) non-convexity if there exists *no* radiating point $\overline{\boldsymbol{u}}^0 \in R$ internal to $\boldsymbol{A}$, such that there is a unique solution to *Problem (i)* or *(ii)* (as applicable) for *all* possible search directions $\boldsymbol{s}^i$ radiating from $\overline{\boldsymbol{u}}^0$. If this type of non-convexity occurs, then a modified ray approach must be adopted to determine the entire workspace boundary. This strategy is described in the next subsection.

2.2.5 Strategy for type (ii) non-convexity

The workspace shown in Figure 2-6(a) exhibits type (ii) non-convexity. For any arbitrarily chosen radiating point $\boldsymbol{u}_1^0$ within $\boldsymbol{A}$, only a segment $\partial \boldsymbol{A}^1$ of the workspace boundary can be mapped, as shown in Figure 2-6(b). It can be seen that for this particular positioning of $\boldsymbol{u}_1^0$, that segments of the boundary between points $\boldsymbol{u}^{bi}$ and $\boldsymbol{u}^{b(i+1)}$ and points $\boldsymbol{u}^{bj}$ and $\boldsymbol{u}^{b(j+1)}$ are not determined. It is clear that similar problems will arise for any other positioning of $\boldsymbol{u}_1^0$. Missed sections of the workspace boundary can be visually detected by examining the distribution of points along the workspace boundary. Large distances between adjacent points are indicative of such a missed section. Constraint violations, resulting in incorrectly determined boundary points, often occur in the vicinity of a missed section of the workspace boundary.

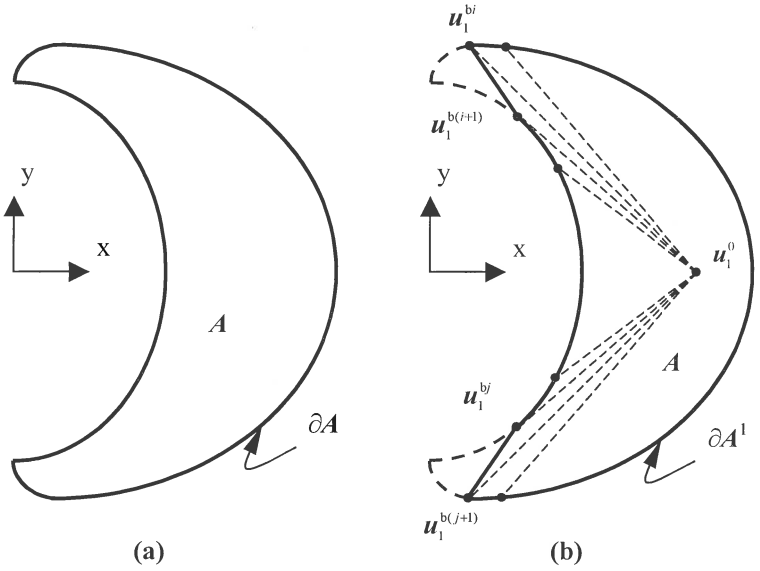

Figure 2-6: Partial determination of the workspace boundary

If any portion of the boundary cannot be mapped from an initial radiating point $\boldsymbol{u}_1^0$, then the missing section may be determined by choosing a new radiating point $\boldsymbol{u}_i^0$, i=2,3,…, and mapping the missing section ∂A^i as shown in Figure 2-7(b). Note that for the sub-section mapping to be successful, the new radiating point must be chosen so that there is a unique solution to *Problem (i)* or *(ii)* (as applicable) for the chosen search directions. A method which works well is to choose the new radiating point so that mapping takes place through an angle of π/2. As an example, consider Figure 2-7(a). Given two points p and q with associated coordinates $\boldsymbol{u}_1^{bp}$ and $\boldsymbol{u}_1^{bq}$ between which remapping must take place, it is easy to calculate β, the angle of the line joining points p and q and the x-axis. The coordinates of $\boldsymbol{u}_C$, the midpoint of the same line, are simply

$$\boldsymbol{u}_C = \frac{\boldsymbol{u}_1^{bp} + \boldsymbol{u}_1^{bq}}{2} \tag{2.14}$$

The coordinates of the new radiating point $\boldsymbol{u}_2^0$ are then taken as

$$\boldsymbol{u}_2^0 = \boldsymbol{u}_C + \frac{d}{2}\begin{bmatrix} \sin\beta \\ -\cos\beta \end{bmatrix} \tag{2.15}$$

where d is the distance between the two points. Once the missing section has been mapped, the newly determined points may be merged with the existing boundary points ∂A^{i-1}.

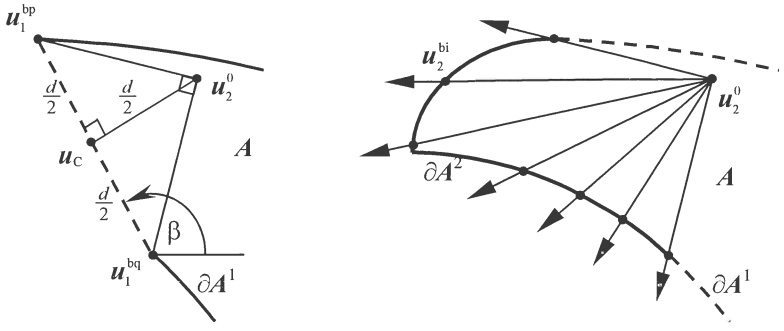

Figure 2-7: Mapping a missed section of the workspace boundary (a) Determining a new radiating point and (b) the newly mapped section ∂A^2

2.2.6 Precise mapping of the bifurcation points

Bifurcation points are important features on a workspace boundary since they often correspond to extreme positions of a manipulator, where a subset of constraints **(2.2)**, **(2.3)** and **(2.4)** are active. The number of degrees of freedom of a system will determine the dimension of this subset. Since a discrete method with fixed angular increments is used to map the boundary, it is unlikely that a bifurcation point will coincide exactly with a search ray. Figure

2-8 illustrates the importance of bifurcation point mapping. Figure 2-8(a), (b) and (c) respectively show the actual workspace boundary and the search directions, the discretized solution obtained without bifurcation mapping, and the solution obtained using precise bifurcation mapping.

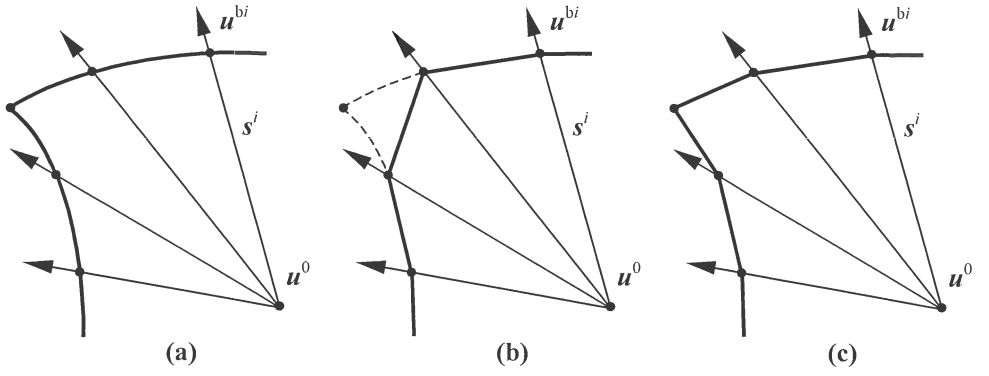

Figure 2-8: Bifurcation mapping, (a) the workspace boundary, (b) discretized solution and (c) solution with bifurcation mapping

In general, the positions of these bifurcation points can be determined by identifying the constraints active at the bifurcation point and then solving for the point of intersection of these constraints using an optimization approach. The exact implementation of the optimization approach to determine bifurcation points is manipulator dependent and will be discussed separately.

2.2.7 Voids in the workspace

It may occur that there are infeasible regions, or voids, within the boundaries of the maximal workspace. The boundary of such a void will be referred to as the *void boundary*. The dashed line in Figure 2-9 shows a void boundary.

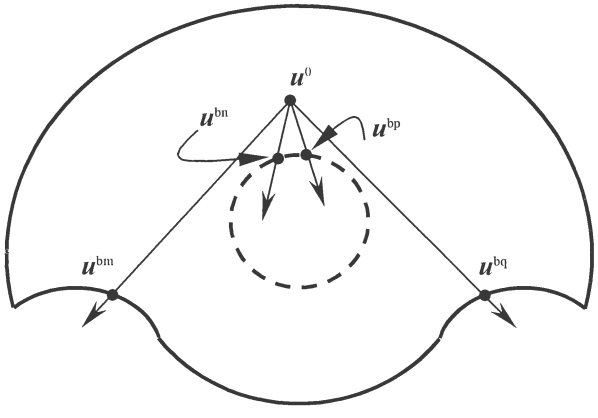

Figure 2-9: Void in the workspace

As the existence of such a void is not known *a priori*, the main challenge lies in detecting whether there are voids within the workspace or not. It is proposed that, once the external workspace boundary has been mapped, voids may be identified by an application of the ray

tracing principle as shown in Figure 2-9. *Problem (i)* of section 2.2.3 is applied to successive rays emanating from a suitably chosen *feasible* radiating point $\boldsymbol{u}^0$. The starting point for each of these optimization problems is chosen as the radiating point $\boldsymbol{u}^0$. If there is no void between the radiating point and the workspace boundary, then the workspace boundary will generally be found, as is the case with points $\boldsymbol{u}^{bm}$ and $\boldsymbol{u}^{bq}$. If, however, there is a void between the radiating point and the workspace boundary the optimizer will converge to a strong local minimum corresponding to the void boundary (points $\boldsymbol{u}^{bn}$ and $\boldsymbol{u}^{bp}$). By comparing the results of the ray search with the actual workspace boundary, it is possible to determine the approximate location of voids within the workspace.

Once a void has been identified, a radiating point inside the void can be chosen, and the boundaries of the void may then easily be determined by applying the ray methodology as before and *minimizing* the distance between the radiating point $\boldsymbol{u}^0$ and point $\boldsymbol{u}$:

$$\underset{\boldsymbol{u},\boldsymbol{w}}{\text{minimize}} \left\| \boldsymbol{u} - \boldsymbol{u}^0 \right\|$$

$$\text{such that } \boldsymbol{v}^{\min} \le \boldsymbol{v}(\boldsymbol{u},\boldsymbol{w}) \le \boldsymbol{v}^{\max}$$

$$\boldsymbol{w}^{\min} \le \boldsymbol{w} \le \boldsymbol{w}^{\max}$$

$$\text{and } \boldsymbol{g}^{\min} \le \boldsymbol{g}(\boldsymbol{u},\boldsymbol{v}(\boldsymbol{u},\boldsymbol{w}),\boldsymbol{w}) \le \boldsymbol{g}^{\max}$$

$$\text{and subject to equality constraints}$$

$$\boldsymbol{h}(\boldsymbol{u},\boldsymbol{s}) = \boldsymbol{0},\ \boldsymbol{h} \in R^{mu-1}$$

2.2.8 Basic modified ray algorithm for mapping the boundary of a general workspace

With reference to the above discussion the following procedure is suggested for the general determination of convex or non-convex workspace boundaries:

Step (i): Choose a suitable *initial radiating point* $\boldsymbol{u}_1^0$. Choose an angular increment δ. Set $i = 1$, $\theta_1^0 = 0$ and $N_1 = 2\pi/\delta$.

Step (ii): Successively solve *Problem (i)* or *(ii)* from θ_i^0 through a sweep angle of $N_i\delta$ to determine a section of the workspace boundary $\partial \boldsymbol{A}^i$. Monitor the active constraints and determine the position of individual bifurcation points as their existence arises along the boundary.

Step (iii): If $i \ge 2$ then merge $\partial \boldsymbol{A}^i$ with $\partial \boldsymbol{A}^{i-1}$.

Step (iv): Examine the workspace boundary $\partial \boldsymbol{A}^i$ to determine whether or not there is a boundary section missing. A missing section of workspace boundary may be present if large

displacements between successively determined boundary points occur. These large displacements occur as the algorithm jumps from one part of the workspace boundary to another more distant part. The criteria used is that if the distance between two successive boundary points is greater than twice the distance between the previous two boundary points, then there is a section of boundary missing.

If there is no section missing then the workspace has been fully determined (*end of procedure*). If there is a missing segment then set i=i+1 and proceed to step (v).

Step (v): Prompt for intersection numbers p and q corresponding to points $\boldsymbol{u}^{bp}$ and $\boldsymbol{u}^{bq}$ between which re-mapping will take place. Using the condition that the re-mapping angle be equal to $\pi/2$, calculate the new radiating point $\boldsymbol{u}_i^0$ as well as the starting angle θ_i^0 and the number of increments, N_i. Go to step (ii).

2.3 Application to the two degree of freedom manipulator

The example two degree of freedom manipulator considered in chapter 1 (see Figure 1-5), will be used here to illustrate the ray method. In agreement with the definitions given in section 2.2.1 the input coordinates are the leg lengths, $\boldsymbol{v} = (v_1, v_2)^T = (l_1, l_2)^T$, and the output coordinates are the coordinates of the working point, $\boldsymbol{u} = (u_1, u_2)^T = (x_P, y_P)^T$. There are no intermediate coordinates.

As with the continuation method, the inverse kinematics are performed to give the leg lengths in terms of the position of the working point:

$$\begin{aligned} l_1^2 &= x_P^2 + y_P^2 \\ l_2^2 &= (x_P - x_B)^2 + y_P^2 \end{aligned} \tag{2.16}$$

The kinematic constraints relating the input and output coordinates are obtained from **(2.16)** and written in terms of the generalized coordinates $\boldsymbol{q}$:

$$\boldsymbol{\Phi}(\boldsymbol{q}) = \begin{bmatrix} v_1^2 - u_1^2 - u_2^2 \\ v_2^2 - (u_1^2 - x_B) - u_2^2 \end{bmatrix} = \boldsymbol{0} \tag{2.17}$$

Explicit expressions for the leg lengths in terms of the output coordinates are obtained from **(2.17)**.

$$\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{u}) = \begin{bmatrix} \sqrt{u_1^2 + u_2^2} \\ \sqrt{(u_1 - x_B)^2 + u_2^2} \end{bmatrix} \tag{2.18}$$

The leg length constraints **(1.1)** imposed on the manipulator take the form

$$\boldsymbol{v}^{\min} \leq \boldsymbol{v}(\boldsymbol{u}) \leq \boldsymbol{v}^{\max} \tag{2.19}$$

where $\boldsymbol{v}^{\min} = [l_1^{\min}, l_2^{\min}]^T$ and $\boldsymbol{v}^{\max} = [l_1^{\max}, l_2^{\max}]^T$.

From a starting point $\boldsymbol{u}_0$, determined using **(2.12)**, and for a given search direction $\boldsymbol{s}$, *Problem (i)* for this example becomes

$$\underset{\boldsymbol{u}}{\text{maximize}} \left\| \boldsymbol{u} - \boldsymbol{u}^0 \right\|$$

$$\text{such that } \boldsymbol{v}^{\min} \leq \boldsymbol{v}(\boldsymbol{u}) \leq \boldsymbol{v}^{\max}$$

$$\text{and subject to equality constraint } h(\boldsymbol{u},\boldsymbol{s}) = (x\text{-}x^0)/s_x + (y\text{-}y^0)/s_y = 0$$

For this two degree of freedom manipulator bifurcation points will correspond to positions on the workspace boundary where two of the leg length constraints **(2.19)** are simultaneously active. The precise mapping of the bifurcation points is done by, having identified the subset of active constraints from **(2.19)** during the boundary mapping procedure, then minimizing

$$\left\| \boldsymbol{v}(\boldsymbol{u}) - \boldsymbol{v}^{\text{ext}} \right\|^2 \tag{2.20}$$

with respect to $\boldsymbol{u}$. The vector $\boldsymbol{v}^{\text{ext}}$ contains the extreme leg length values associated with the identified active constraints.

Consider the case where $x_B = 4$. Various length limits considered are given in Table 2-1.

Manipulator	$l_1^{\min}$	$l_1^{\max}$	$l_2^{\min}$	$l_2^{\max}$
L1	2.25	3.25	2.25	3.75
L2	2.00	3.25	2.00	3.75
L3	1.75	3.25	1.75	3.75

Table 2-1: Two degree of freedom manipulator leg length limits

Figure 2-10 shows one part of the workspace of manipulator L1 which was determined using the ray method. Each point determined is represented by a dot on the workspace boundary.

The distribution of points along the workspace boundary varies greatly. For the ray method, sections of the boundary closer to the radiating point, or which are close to perpendicular to the search rays are mapped at a finer resolution than the rest of the boundary. Due to the symmetry of the example it is known that the workspace is mirrored about the x-axis. The full workspace of manipulator L1 is shown in Figure 2-11(a).

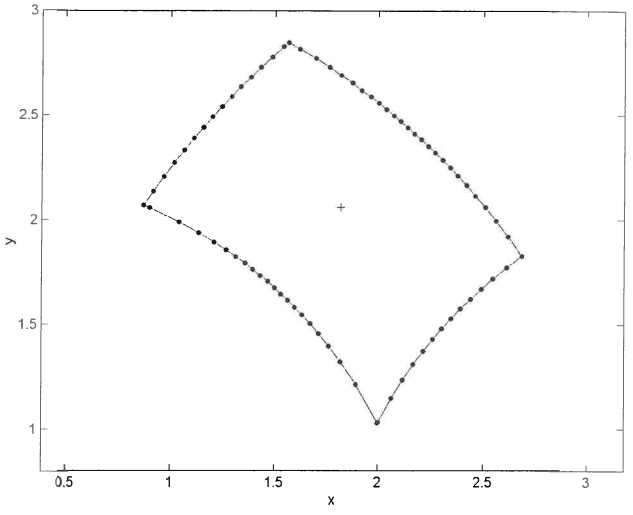

Figure 2-10: Part of the workspace of L1 determined using the ray method

Depending on the starting point used for optimization problem **(2.12)** to obtain the midpoint of the workspace one of two possible midpoints will be determined. The top or bottom parts of the workspace can be mapped using the relevant midpoint as the radiating point. Figure 2-11(b) and Figure 2-11(c) respectively show the workspaces of manipulators L2 and L3. In each of these cases the initial boundary determined using the original ray method is shown as a dotted line. The large vertical jumps that occur indicate the presence of non-convexity and the need for remapping. These workspaces were mapped in a counterclockwise manner.

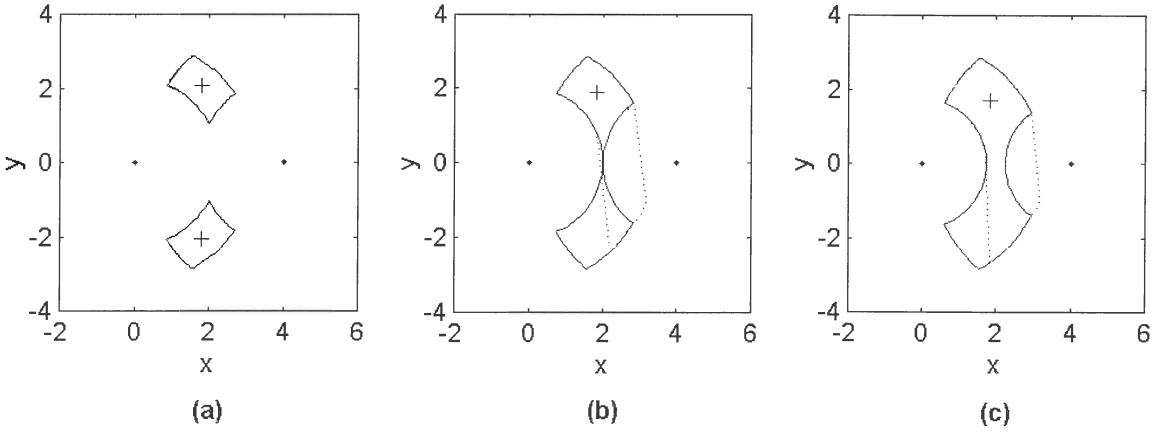

Figure 2-11: Workspaces of manipulators (a) L1, (b) L2 and (c) L3

2.4 Preliminary evaluation of the ray method

Snyman et al. [1] and Du Plessis [2] have already shown in their studies that the ray method in its original form can effectively be applied to the determination of convex workspaces and workspaces exhibiting type (i) non-convexity. Type (ii) non-convexity can usually be addressed by using the modified strategy suggested in this chapter. There are some cases,

however, in which even the modified strategy cannot be used to determine the workspace of a manipulator exhibiting type (ii) non-convexity. This can occur when there is a section of the boundary for which there exists no radiating point $\boldsymbol{u}^i \in R^{mu}$, such that, for the corresponding search directions, *Problem (i)* or *(ii)* (as applicable) has unique solutions.

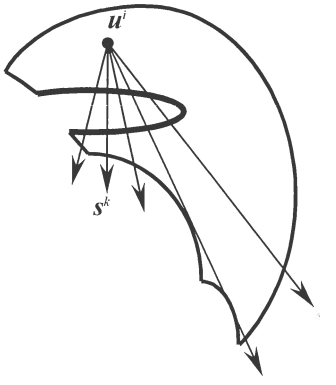

Figure 2-12: Non-unique boundary section

An example of this case is shown in Figure 2-12, where there exist no radiating points that can be used to uniquely map the bold section of the workspace boundary. In attempting to map this section of boundary, applying optimization *problem (i)* or *(ii)* along ray $\boldsymbol{s}^k$ results in two possible solutions. There is one local minimum corresponding to a point on the bold section of the workspace boundary, and one global minimum corresponding to another section of the workspace boundary. If these two solutions are in relatively close proximity, then the global minimum will almost always be found. It is therefore difficult to map the bold section of boundary. Type (ii) non-convexity can thus only sometimes be addressed using the modified ray method described in this chapter.

A further property of the modified ray method is that it requires user inputs during the determination of the workspace. This approach does have the advantage that, with sufficient interactions by the user, the workspace can usually be accurately found. This approach, however, tends to be tedious and time-consuming.

In the next chapter the modified ray method will be illustrated by its application to a generally constrained planar Stewart platform.

Chapter 3: APPLICATION OF THE RAY METHOD TO THE PLANAR STEWART PLATFORM CONSIDERING LEG INTERFERENCES, SINGULARITIES AND PASSIVE JOINT CONSTRAINTS

3.1 Geometry of the platform

In this chapter the modified ray approach will be applied to the planar Stewart platform shown in Figure 3-1. This manipulator is based on the platform studied by Haug et al. [18].

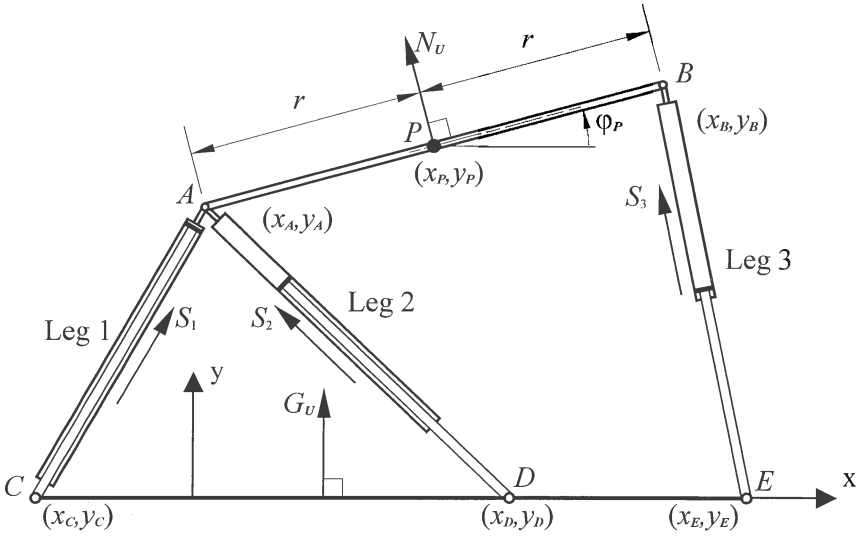

Figure 3-1: Planar Stewart platform

The mechanism consists of a platform of length $2r$ connected to a base by three linear actuators, which control the three output degrees of freedom of the platform. The actuators have leg lengths l_1, l_2 and l_3 and are joined to the base and platform by means of revolute joints identified by the letters $A - E$. For the purposes of this paper it will be assumed that $y_C = y_D = y_E = 0$. The coordinates of point P, the mid-point of the platform, are (x_P, y_P) and the orientation of the platform is φ_P. Unit vectors $\boldsymbol{S}_1$, $\boldsymbol{S}_2$, and $\boldsymbol{S}_3$ are defined along the actuators. $\boldsymbol{N}_U$ is a unit vector perpendicular to the platform. $\boldsymbol{G}_U$ is a unit vector perpendicular to the base.

With reference to the definitions given in sections 2.2.1 and 2.2.2, the actuator leg lengths are the input variables, i.e. $\boldsymbol{v} = [l_1, l_2, l_3]^T$. The global coordinates of the working point P form the output coordinates, i.e. $\boldsymbol{u} = [x_P, y_P]^T$. The rotation angle of the platform is the only

intermediate coordinate, i.e. $w = \varphi_P$. The generalized coordinates (2.5) for the Stewart platform are given by:

$$\begin{aligned} \boldsymbol{q} &= \left[\boldsymbol{u}^T, \boldsymbol{v}^T, \boldsymbol{w}^T\right]^T \\ &= \left[\boldsymbol{u}^T, \boldsymbol{v}^T, w\right]^T \\ &= \left[x_P, y_P, l_1, l_2, l_3, \varphi_P\right]^T \end{aligned} \tag{3.1}$$

3.2 Constraint equation formulation

3.2.1 Kinematic constraints

The Stewart platform has three degrees of freedom, since any three of the coordinates in (3.1) uniquely define the configuration of the system. This implies the existence of three kinematic constraint equations of the form (2.6), specifying the interrelationships between the coordinates.

For any given position and orientation of the platform, the coordinates of points A and B are given as follows:

$$\begin{aligned} x_A &= x_P - r\cos\varphi_P \\ y_A &= y_P - r\sin\varphi_P \\ x_B &= x_P + r\cos\varphi_P \\ y_B &= y_P + r\sin\varphi_P \end{aligned} \tag{3.2}$$

The inverse kinematics can then be performed to give the leg lengths in terms of x_P, y_P and φ_P:

$$\begin{aligned} l_1^2 &= (x_P - r\cos\varphi_P - x_C)^2 + (y_P - r\sin\varphi_P - y_C)^2 \\ l_2^2 &= (x_P - r\cos\varphi_P - x_D)^2 + (y_P - r\sin\varphi_P - y_D)^2 \\ l_3^2 &= (x_P + r\cos\varphi_P - x_E)^2 + (y_P + r\sin\varphi_P - y_E)^2 \end{aligned} \tag{3.3}$$

This can be rewritten in the standard form for the kinematic constraint equations (2.6) as:

$$\boldsymbol{\Phi}(\boldsymbol{u}, \boldsymbol{v}, w) = \begin{bmatrix} v_1^2 - (u_1 - r\cos w - x_C)^2 - (u_2 - r\sin w - y_C)^2 \\ v_2^2 - (u_1 - r\cos w - x_D)^2 - (u_2 - r\sin w - y_D)^2 \\ v_3^2 - (u_1 + r\cos w - x_E)^2 - (u_2 + r\sin w - y_E)^2 \end{bmatrix} = \boldsymbol{0} \tag{3.4}$$

There are a number of other constraints acting on the system, which limit the working region of point P.

3.2.2 Leg length constraints

Each of the three actuator legs has a maximum and minimum working length. The constraints on the leg lengths are formally expressed as:

$$0 < l_i^{\min} \le l_i \le l_i^{\max}, \; i = 1,2,3 \tag{3.5}$$

From (3.4) the explicit expressions for $\boldsymbol{v}$ follow:

$$\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{u}, w) = \begin{bmatrix} \sqrt{(u_1 - r\cos w - x_C)^2 + (u_2 - r\sin w - y_C)^2} \\ \sqrt{(u_1 - r\cos w - x_D)^2 + (u_2 - r\sin w - y_D)^2} \\ \sqrt{(u_1 + r\cos w - x_E)^2 + (u_2 + r\sin w - y_E)^2} \end{bmatrix} \tag{3.6}$$

The inequality constraints (3.5) may be written in the standard form:

$$\boldsymbol{v}^{\min} \le \boldsymbol{v}(\boldsymbol{u}, w) \le \boldsymbol{v}^{\max} \tag{3.7}$$

where $\boldsymbol{v}^{\min} = [l_1^{\min}, l_2^{\min}, l_3^{\min}]^T$, $\boldsymbol{v}^{\max} = [l_1^{\max}, l_2^{\max}, l_3^{\max}]^T$ and with $\boldsymbol{u}$ and w specified, $\boldsymbol{v}(\boldsymbol{u}, w)$ is given by (3.6).

3.2.3 Mechanical joint constraints

To calculate the angles between the actuators, platform and base the following unit vectors are determined:

$$\begin{aligned} \boldsymbol{S}_1 &= \frac{1}{l_1}[x_A - x_C, \quad y_A - y_C]^T \\ \boldsymbol{S}_2 &= \frac{1}{l_2}[x_A - x_D, \quad y_A - y_D]^T \\ \boldsymbol{S}_3 &= \frac{1}{l_3}[x_B - x_E, \quad y_B - y_E]^T \end{aligned} \tag{3.8}$$

$$\boldsymbol{N}_U = [-\sin\varphi_P, \quad \cos\varphi_P]^T \tag{3.9}$$

$$\boldsymbol{G}_U = [0, \quad 1]^T \tag{3.10}$$

Consider the dot product between $\boldsymbol{N}_U$ and any leg vector $\boldsymbol{S}_i$, $i=1,2,3$:

$$\boldsymbol{N}_U \cdot \boldsymbol{S}_i = \|\boldsymbol{N}_U\| \|\boldsymbol{S}_i\| \cos\xi_i$$

where ξ_i is the angle between the two vectors. Since $\boldsymbol{N}_U$ and $\boldsymbol{S}_i$ are unit vectors:

$$\boldsymbol{N}_U \cdot \boldsymbol{S}_i = \cos\xi_i$$

and thus

$$\xi_i = \cos^{-1}(\boldsymbol{N}_U \cdot \boldsymbol{S}_i), \quad i = 1,2,3 \tag{3.11}$$

The direction of rotation is found by examining the sign of the cross product between the two vectors:

$$\begin{matrix} \text{positive angle if } \alpha > 0 \\ \text{negative angle if } \alpha < 0 \end{matrix} \quad \text{where } \boldsymbol{N}_U \times \boldsymbol{S}_i = \alpha\hat{\mathrm{z}} \tag{3.12}$$

and $\hat{z}$ denotes the unit vector out of the plane in the z-direction in agreement with the right hand rule convention. By using **(3.11)** and **(3.12)** the angle between the platform normal vector and any actuator can be uniquely determined in the range $-\pi \leq \xi_i \leq \pi$.

Similarly for the angles between the base normal vector and the actuators:

$$\zeta_i = \cos^{-1}(\boldsymbol{G}_U \cdot \boldsymbol{S}_i), \quad i = 1,2,3 \tag{3.13}$$

The sign of the angle is once more defined as follows:

$$\begin{matrix} \text{positive angle if } \alpha > 0 \\ \text{negative angle if } \alpha < 0 \end{matrix} \quad \text{where } \boldsymbol{G}_U \times \boldsymbol{S}_i = \alpha\hat{z} \tag{3.14}$$

giving ζ_i uniquely in the range $[-\pi,\pi]$.

The actuators are physically connected to the platform at one end and the base at the other by means of revolute joints. The construction of these joints will impose certain limits on the maximum and minimum angles that the leg can attain relative to the platform or base. These constraints are formally expressed as follows:

$$\xi_i^{\min} \leq \xi_i \leq \xi_i^{\max}, \quad i = 1,2,3 \tag{3.15}$$

and

$$\zeta_i^{\min} \leq \zeta_i \leq \zeta_i^{\max}, \quad i = 1,2,3 \tag{3.16}$$

Since the angles ξ_i and ζ_i depend on the output and intermediate coordinates, **(3.15)** and **(3.16)** may be rewritten in the standard form as

$$\boldsymbol{g}^{\min} \leq \boldsymbol{g}(\boldsymbol{u}, w) \leq \boldsymbol{g}^{\max} \tag{3.17}$$

where $\boldsymbol{g} = [\xi_1, \xi_2, \xi_3, \zeta_1, \zeta_2, \zeta_3]^T$.

3.2.4 Leg interference constraints

If the mechanical joint constraints, presented in section 3.2.3, are not enforced or not sufficiently restrictive, then the platform may attain an inverted position as shown in Figure 3-2(a). Depending on the mechanical construction of the Stewart platform, such a configuration may not be physically attainable due to interference between the actuator legs 2 and 3, and 1 and 3 occurring at points I^I and I^{II} respectively. Figure 3-2(b) shows the limiting cases of leg interference when the platform is in positions AB^I and AB^{II}. By inspection of this figure it can be seen that possibility of leg inference at I^I and I^{II} for this geometry can be excluded by simply implementing the following constraints:

$$\xi_2 \leq \frac{\pi}{2} = \xi_2^{\max} \text{ and } \xi_3 \geq -\frac{\pi}{2} = \xi_3^{\min} \tag{3.18}$$

The limiting cases given in **(3.18)** clearly correspond to platform positions AB^{I} and AB^{II} with $\xi_2 = \pi/2$ and $\xi_3 = -\pi/2$ respectively.

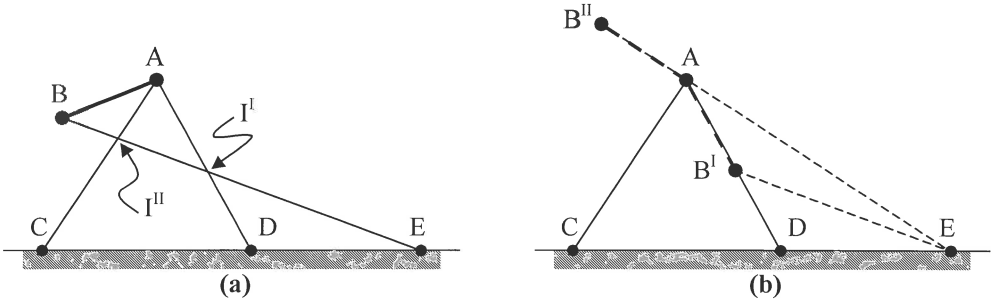

Figure 3-2: Leg interference

Leg interference may also occur between legs 1 and 2 when $\zeta_i = \pi/2$ or when $\zeta_i = -\pi/2$, i=1,2. The possibility of this interference may be excluded by ensuring that:

$$\zeta_i > -\frac{\pi}{2} = \zeta_i^{\min} \text{ and } \zeta_i < \frac{\pi}{2} = \zeta_i^{\max}, \quad i=1,2 \tag{3.19}$$

This could be alternatively be formulated as

$$y_P > r \sin \varphi_P \tag{3.20}$$

(See Appendix A)

3.2.5 Singularity constraints

Singular positions are often associated with real restriction on motion control of the platform. Vijaykumar et al. **[36]** note that, for serial manipulators, it is possible to almost completely eliminate singular positions by limiting the ranges of joint motion. In Appendix A it is shown that the planar Stewart platform is in a singular configuration when the platform is collinear with leg 3 or when legs 1 and 2 are collinear. It is thus possible to eliminate the occurrence of singular configurations within the workspace by the imposition of joint angle constraints. The imposition of the following constraints prevents the platform from assuming the first singular configuration of Appendix A:

$$\xi_3 > -\frac{\pi}{2} = \xi_3^{\min} \text{ and } \xi_3 < \frac{\pi}{2} = \xi_3^{\max} \tag{3.21}$$

Constraint expression **(3.19)** or **(3.20)** prevents the second singular case of Appendix A from occurring.

For the given geometry of the Stewart platform, the use of constraints **(3.21)** makes the use of the leg interference constraints **(3.18)** redundant.

3.2.6 Implementation of angular constraints

In general, when implementing constraints **(3.15)** to **(3.21)**, care should be taken to ensure that there are no redundant constraints present. If these redundant constraints run close to other constraints in x-y-φ space, they can cause the optimization algorithms to misidentify the actual active constraints. As an example of the elimination of redundant constraints, consider the case where the specification $\xi_i^{\min} = -1.5$ and $\xi_i^{\max} = 1.5$, i=1,2,3 is made in **(3.15)**. This means that constraints **(3.18)** and **(3.21)** are redundant, as ξ_i can never attain a value of $\pm\pi/2$, and therefore they do not need to be included in the optimization problem.

3.3 Calculation of bifurcation points

For a planar parallel manipulator with three degrees of freedom a *bifurcation point* will occur on the workspace boundary either when any three constraints are simultaneously active *or* when two constraints are active and one of the actuators and the working point are collinear.

For the case where three constraints are active at a bifurcation point, the precise mapping of the bifurcation point corners is done by, having identified the subset of active constraints from **(2.2)**, **(2.3)** and **(2.4)** during the boundary mapping procedure, then solving for

$$\min_{\boldsymbol{u},w} \left\| \boldsymbol{p}(\boldsymbol{u},w) - \boldsymbol{p}^{\text{ext}} \right\|^2 \tag{3.22}$$

where $\boldsymbol{p}(\boldsymbol{u},w)$ and $\boldsymbol{p}^{\text{ext}}$ respectively correspond to the constraint functions and the associated set of extreme values of the identified vector of active constraints.

For the case where two constraints are active at a bifurcation point, it is necessary to introduce an extra condition to ensure that the working point and relevant extreme actuator leg remain collinear. Assume that $\boldsymbol{a}^i$ is a vector pointing along leg i, which is to remain collinear with the working point P of the manipulator. Vector $\boldsymbol{b}^i$ is perpendicular to the line going from the platform attachment point of leg i to point P. Leg i will be collinear with the working point when vectors $\boldsymbol{a}^i$ and $\boldsymbol{b}^i$ are perpendicular, which is mathematically stated by the condition $\boldsymbol{a}^i \cdot \boldsymbol{b}^i = 0$. The position of the bifurcation point can thus be determined by, once having identified the relevant constraints and collinear leg, then solving

$$\min_{\boldsymbol{u},w}[\left(p_j(\boldsymbol{u},w) - p_j^{ext}\right)^2 + \left(p_k(\boldsymbol{u},w) - p_k^{ext}\right)^2 + \left(\boldsymbol{a}^i(\boldsymbol{u},w) \cdot \boldsymbol{b}^i(\boldsymbol{u},w)\right)^2] \tag{3.23}$$

where $p(\boldsymbol{u},w)$ and p^{ext} in the first two terms correspond to the constraint functions and associated extreme values of the identified active constraints and the last term contains the collinearity condition.

3.4 Implementation of the method

Equations **(3.7)**, **(3.17)**, **(3.4)** and **(3.6)** for the planar Stewart platform correspond to *Case (i)* of Section 2.2.3, specified in the general case by expressions **(2.2)**, **(2.4)**, **(2.6)** and **(2.9)**. The boundary ∂A of the accessible set of the planar Stewart platform may therefore be numerically determined by applying the methodology described in Section 2.2.8, in which optimization *Problem (i)* is successively solved. For the cases depicted here, boundary mapping was done at regular intervals of $\delta = 5°$. Note that there is no explicit restriction **(2.3)** on w.

The techniques described here have been implemented in an interactive computer code *WSPCON*. The user specifies the geometry of the platform as well as the limits on the joint angles and leg lengths. The code then automatically maps the workspace boundary including the bifurcation points. Once this is completed the boundary can be displayed and visually inspected for discontinuities that may indicate non-convexity. The user then specifies two points on the boundary between which re-mapping should take place, and the code calculates a new radiating point and start and end angles and writes a restart file containing this information. When the program is executed again, the new section of boundary is remapped and the merged with the existing boundary. The above procedure can be repeated indefinitely until the entire workspace boundary has been completely mapped.

If any constraint violations occur, the intersection points at which these occur are reported to the user who can use this information when choosing the section of boundary to be re-mapped.

The specific constrained optimization method used in solving the optimization problems is the dynamic trajectory method of Snyman **[32,33]** for unconstrained optimization applied to penalty function formulations (Snyman et al. **[35]**, Snyman **[34]**) of the constrained problems. The particular computer code used is *LFOPC* **[34]**.

3.5 Results for the planar Stewart platform

As an illustration of the proposed method for determining the workspace, an investigation of the effect of varying the bounds on the leg lengths was carried out on a normalized Stewart platform. The values of the constants defining the geometry of this normalized Stewart platform are given in Table 3-1 (also refer to Figure 3-1).

r	x_C	y_C	x_D	y_D	x_E	y_E
1.0	-1.0	0.0	1.0	0.0	2.0	0.0

Table 3-1: Planar Stewart platform constants

The values for the mechanical limits on the joint motion are chosen as $\xi_i^{\min} = \zeta_i^{\min} = -\pi/2$ and $\xi_i^{\max} = \zeta_i^{\max} = \pi/2$. The effects of **(3.15)**, **(3.18)** and **(3.21)** are then included in the following constraints, which are used in determining the workspace:

$$\xi_1 > -\pi/2 \text{ and } \xi_3 < \pi/2 \qquad \textbf{(3.24)}$$

For the platform configurations considered in the rest of this section, constraints **(3.16)** and **(3.19)** never become active.

In addition to the above, constraints **(3.5)** are imposed with leg length limits assuming various values different from that of the standard case defined in Table 3-2. This standard case, defined by Table 3-1 and Table 3-2, corresponds to the configuration and limits used by Haug et al. **[28]**.

i	$l_i^{\min}$	$l_i^{\max}$
1	$\sqrt{2}$	2
2	$\sqrt{2}$	2
3	1	$\sqrt{3}$

Table 3-2: Planar Stewart platform actuator length limits

The effects of respectively varying the maximum and minimum actuator lengths of legs 1 and 2 on the character and size of the workspace are shown in Figure 3-3 and Figure 3-5. In each of these figures, the central plot shows the workspace of the standard case. The plots to the right of this central case show the workspaces with the appropriate limit of Leg 1 increased by 25% and to the left decreased by 25%. The plots above the central plot show the workspaces with the limit of Leg 2 increased by 25% and below decreased by 25%. The mid-point (initial radiating point) of each workspace, as determined using equation **(2.11)**, is indicated by a '+'. Similarly, the effects of varying the limits of legs 1 and 3 are shown in Figure 3-4 and Figure 3-6.

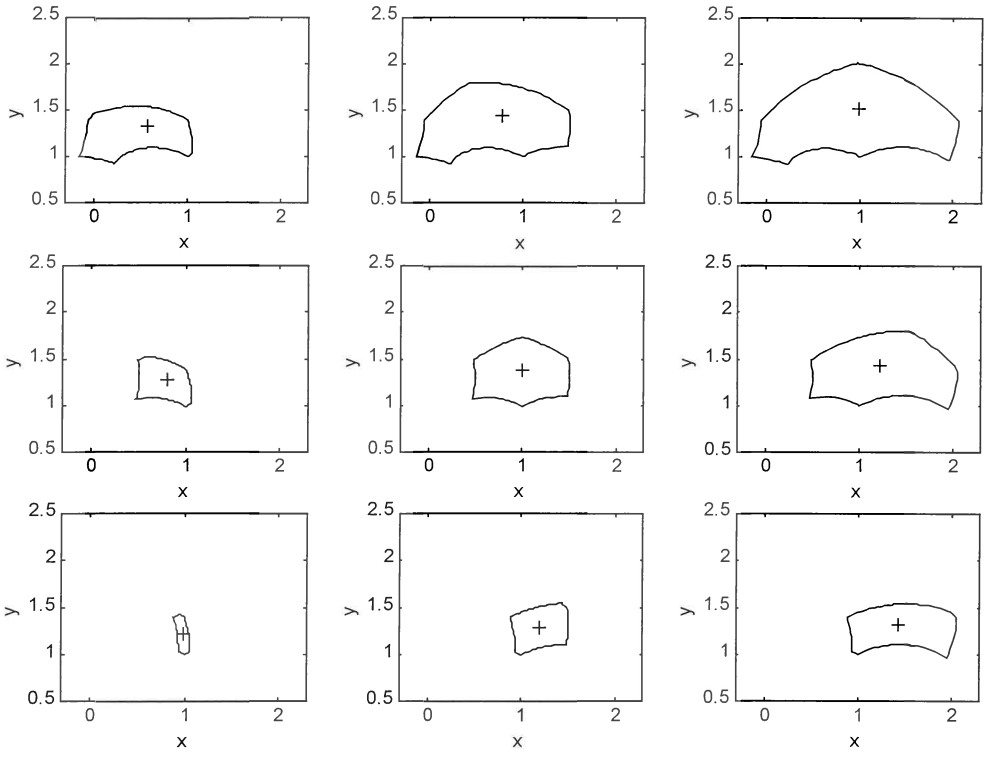

Figure 3-3: $l_i^{\max}$ variation, i=1,2

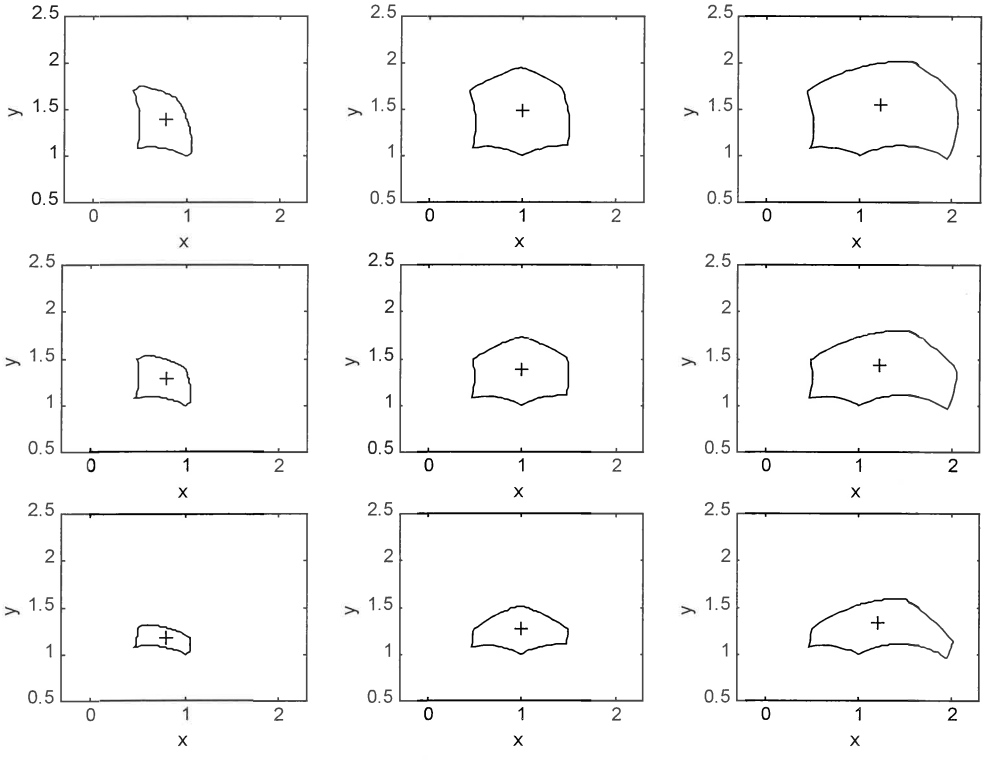

Figure 3-4: $l_i^{\max}$ variation, i=1,3

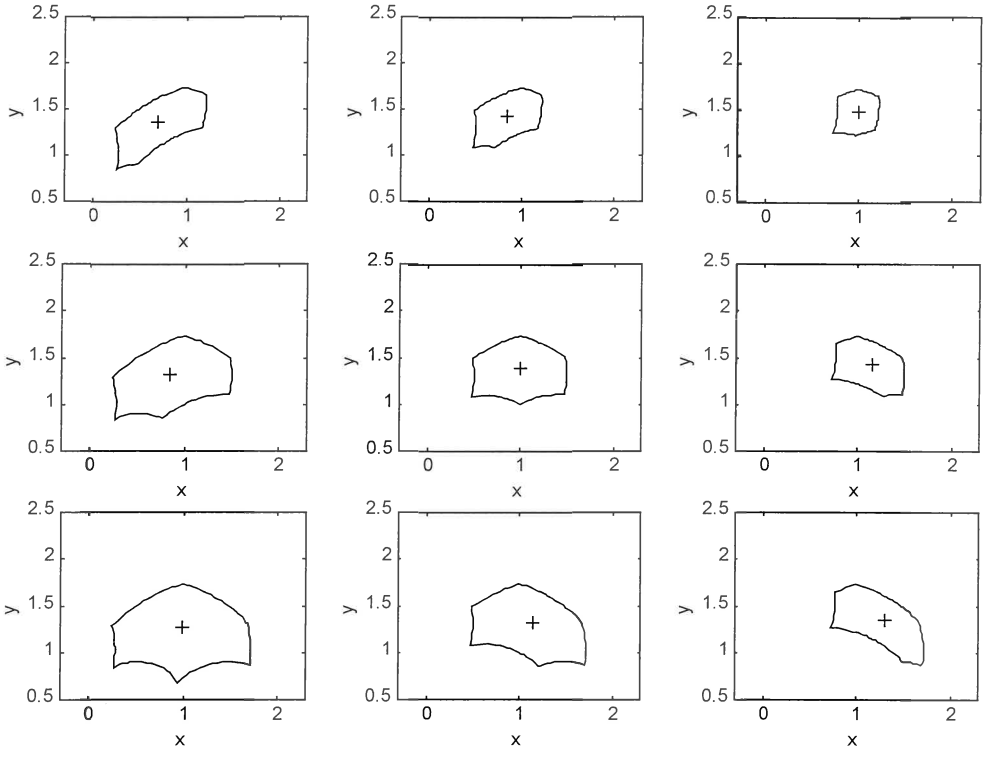

Figure 3-5: $l_i^{\min}$ variation, *i*=1,2

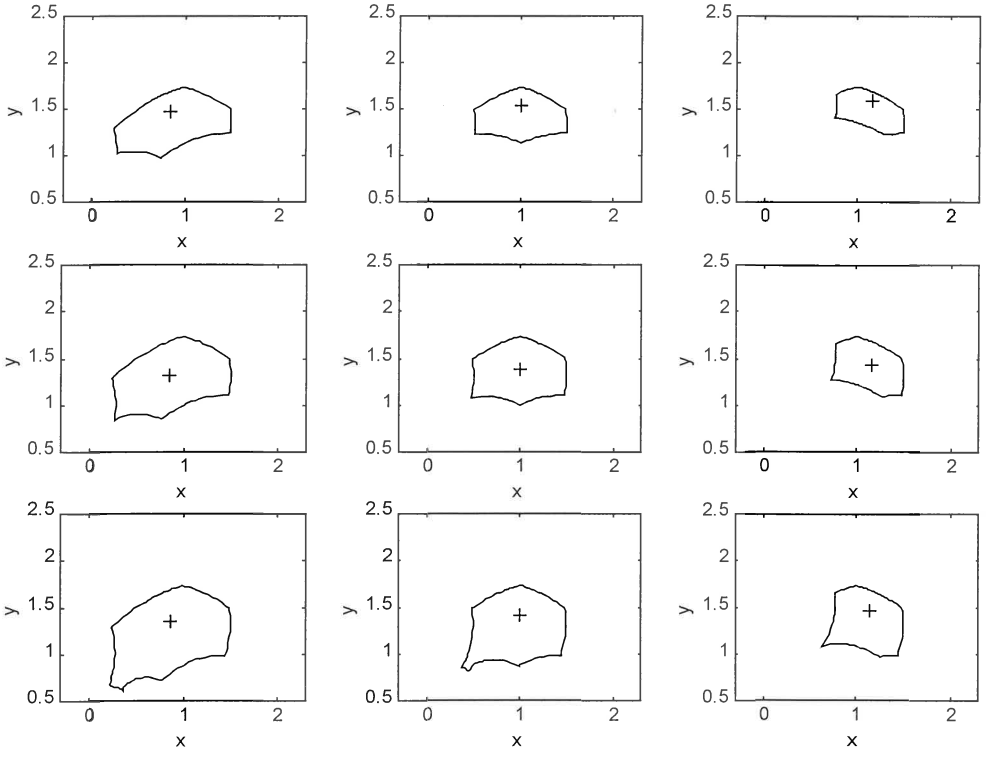

Figure 3-6: $l_i^{\min}$ variation, *i*=1,3

3.6 Conclusion

A comparison of the workspace boundary for the standard case, depicted as the central plot in Figures 3-3 to 3-6, with the results of Haug et al. [28], shows that equally accurate results are obtained using the optimization approach. Furthermore, the current method is capable of easily determining the workspaces of platforms of arbitrary geometry relative to the standard case. Of particular importance is the ease with which non-convex workspaces are determined An example of a non-convex workspace determined using the method is shown in Figure 3-7. This non-convex case differs from the standard case in having all of the maximum leg lengths increased by 25% and the minimum leg lengths of legs 1 and 2 decreased by 25%. As expected the analysis shows that the size of the workspaces of the Stewart platform increase in relation to the platform size with an increase in actuator stroke lengths.

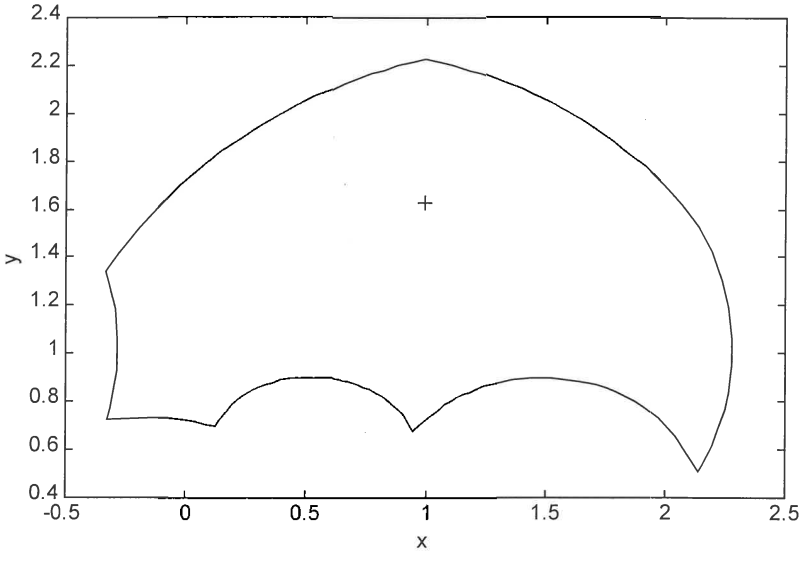

Figure 3-7: Non-convexity

The optimization approach, previously proposed by Snyman et al. [1], has successfully been extended in this chapter to allow for the determination of workspaces of planar Stewart platforms of arbitrary geometry. A formulation for accommodating additional planar constraints has been proposed. It has been shown that the optimization approach is capable of handling the additional constraints of leg interference, passive joint constraints and even singularity constraints with ease. The method, as embodied in a practical interactive computer system, allows for the easy determination of most non-convex workspaces.

Chapter 4: THE CHORD METHOD FOR THE DETERMINATION OF NON-CONVEX WORKSPACES

4.1 Introduction

It has been shown in the previous two chapters that the original ray method and the modified ray method are capable of determining most non-convex workspaces of planar Stewart platforms. However as pointed out in the chapter 2, there do arise certain cases in which the ray method cannot be used to determine the workspace. In this chapter, an alternative general approach, also based on optimization techniques, is proposed. This new approach retains the advantages of the original optimization approach, while being able to determine any non-convex workspace. A further advantage of the new approach is that it allows for a more automated determination of non-convex workspace boundaries.

The new chord method consists of choosing a suitable initial radiating point, and then finding the points of intersection of a ray emanating from this point with the boundary of the accessible set. Once this initial point has been determined, the workspace boundary is traced using constant chord length searches in which the points of intersection of constant radius circles, centered at each previous solution point, and the workspace boundary are determined. The points of intersection are once more determined by means of the dynamic constrained algorithm of Snyman [32,33,34] and Snyman et al. [35].

4.2 Description of the method

4.2.1 Coordinates

As described in chapter 2, generalized coordinates $\boldsymbol{q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_{nq}]^T \in R^{nq}$ are defined that characterize the position and orientation of each body in the mechanism and which, in the vicinity of an assembled configuration, satisfy m independent holonomic kinematic constraint equations of the form

$$\boldsymbol{\Phi}(\boldsymbol{q}) = \boldsymbol{0} \tag{4.1}$$

where $\boldsymbol{\Phi} : R^{nq} \rightarrow R^{m}$ is a smooth function.

The generalized coordinates are once more divided into the input coordinates, $\boldsymbol{v} = [v_1, \ldots, v_{nv}]^T$, output coordinates, $\boldsymbol{u} = [u_1, \ldots, u_{nu}]^T$ and intermediate coordinates $\boldsymbol{w} = [w_1, \ldots, w_{nw}]^T$. The constraint equations **(4.1)** are rewritten in terms of this partitioned generalized coordinates $\boldsymbol{q} = [\boldsymbol{u}^T, \boldsymbol{v}^T, \boldsymbol{w}^T]^T$ as follows:

$$\boldsymbol{\Phi}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = \boldsymbol{0} \tag{4.2}$$

4.2.2 Constraints and the accessible output set

Inequality constraints are often imposed on the input variables and intermediate variables. These take the form

$$\boldsymbol{v}^{\min} \leq \boldsymbol{v} \leq \boldsymbol{v}^{\max} \tag{4.3}$$

and

$$\boldsymbol{w}^{\min} \leq \boldsymbol{w} \leq \boldsymbol{w}^{\max} \tag{4.4}$$

There may also be additional inequality constraints acting on the system, representing relationships between the input, output and intermediate coordinates, that must be satisfied and which take the general form

$$\boldsymbol{g}^{\min} \leq \boldsymbol{g}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) \leq \boldsymbol{g}^{\max} \tag{4.5}$$

The accessible output set $\boldsymbol{A}$ and boundary $\partial\boldsymbol{A}$ of the manipulator are as defined by **(2.7)** and **(2.8)** of section 2.2.2.

The discussion below will be restricted to parallel manipulators, but the general methodology is equally applicable to serial manipulators as well.

4.2.3 Finding an initial point on ∂A

Assume a planar manipulator with a two-dimensional accessible output set $\boldsymbol{A}$. An initial radiating point $\boldsymbol{u}^0$ inside the accessible output set, $\boldsymbol{A}$, is chosen using the method outlined in section 2.2.4. As explained there, if an immediate choice for the radiating point is not evident, then $\boldsymbol{u}^0$ may be obtained from **(2.9)** by solving for $\boldsymbol{u}$ in:

$$\overline{\boldsymbol{v}} = \boldsymbol{v}(\boldsymbol{u}, \overline{\boldsymbol{w}}) \tag{4.6}$$

where

$$\overline{\boldsymbol{v}} = (\boldsymbol{v}^{\min} + \boldsymbol{v}^{\max})/2$$

$$\overline{\boldsymbol{w}} = (\boldsymbol{w}^{\min} + \boldsymbol{w}^{\max})/2$$

If $m = nv$, given $\boldsymbol{u}$ and $\boldsymbol{w}$, system **(2.6)** may easily be solved to give $\boldsymbol{v}$ in terms of $\boldsymbol{u}$ and $\boldsymbol{w}$:

$$\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{u}, \boldsymbol{w}) \tag{4.7}$$

Consistent with the definition of ∂A in **(2.8)**, an initial point $\boldsymbol{u}^{b0}=(x^{b0}, y^{b0})^T$ on the boundary in an arbitrarily chosen direction, designated by a unit vector $\boldsymbol{s}^1 \in R^{nu}$, from $\boldsymbol{u}^0$ is determined by solving the constrained optimization *Problem (i)*, which has been stated in Chapter 2 and is repeated here:

Problem (a):

$$\underset{\boldsymbol{u},\boldsymbol{w}}{\text{maximize}} \left\| \boldsymbol{u}-\boldsymbol{u}^0 \right\|$$

$$\text{such that } \boldsymbol{v}^{\min} \le \boldsymbol{v}(\boldsymbol{u},\boldsymbol{w}) \le \boldsymbol{v}^{\max}$$

$$\boldsymbol{w}^{\min} \le \boldsymbol{w} \le \boldsymbol{w}^{\max}$$

$$\text{and } \boldsymbol{g}^{\min} \le \boldsymbol{g}(\boldsymbol{u},\boldsymbol{v}(\boldsymbol{u},\boldsymbol{w}),\boldsymbol{w}) \le \boldsymbol{g}^{\max}$$

and subject to equality constraints

$$\boldsymbol{h}(\boldsymbol{u},\boldsymbol{s}^1)=\boldsymbol{0},\ \boldsymbol{h} \in R^{nu-1}$$

The solution of *Problem (a)* is illustrated in Figure 4-1

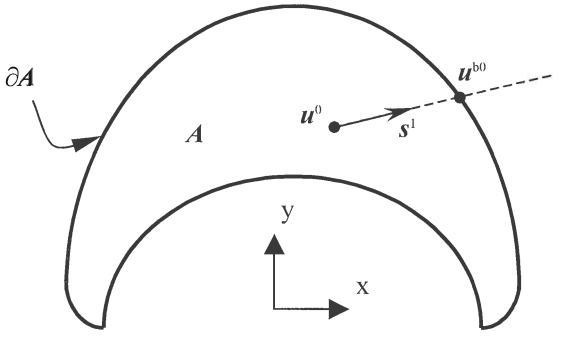

Figure 4-1: Finding an initial point on ∂A

Application of *problem (a)* will *usually* result in an initial point on the outer workspace boundary. It is possible, however, that a point on an *interior boundary curve* corresponding to a local minimum of *problem (a)* is determined instead. If only a portion of the apparent workspace boundary is mapped before breakdown occurs then it is possible that the initial boundary point had been incorrectly determined. The algorithm must then be restarted using a different search direction $\boldsymbol{s}^1$ so that a new initial point on the outer workspace boundary can be determined.

4.2.4 Basic methodology for mapping the boundary of the workspace

Starting at the initial boundary point $\boldsymbol{u}^{b0}$, the workspace boundary is traced by successive circular searches with constant chord length l as depicted in Figure 4-2.

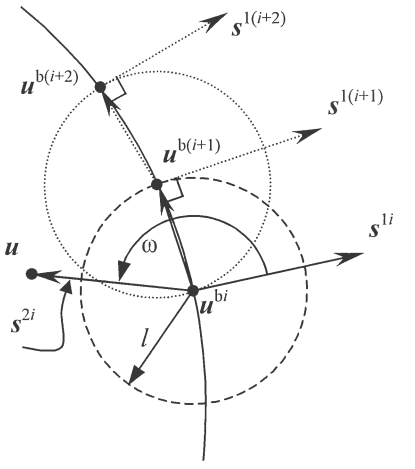

Figure 4-2: Mapping the workspace boundary

Consider any boundary point $\boldsymbol{u}^{bi}$ with an associated *unit* vector $\boldsymbol{s}^{1i}$ pointing *out* of the workspace. A vector $\boldsymbol{s}^{2i}$ from $\boldsymbol{u}^{bi}$ to an arbitrary output point $\boldsymbol{u}=(x, y)^T$, corresponding to the position of the working point P, is

$$\boldsymbol{s}^{2i} = \begin{bmatrix} x - x^{bi} \\ y - y^{bi} \end{bmatrix} \tag{4.8}$$

Dropping the superscript i, the angle ω between the unit vector $\boldsymbol{s}^1$ and the vector $\boldsymbol{s}^2$, defined in the right hand sense is

$$\omega = \begin{cases} \cos^{-1}\left(\dfrac{\boldsymbol{s}^1 \cdot \boldsymbol{s}^2}{\|\boldsymbol{s}^2\|}\right) \text{ if } \alpha \geq 0 \\ 2\pi - \cos^{-1}\left(\dfrac{\boldsymbol{s}^1 \cdot \boldsymbol{s}^2}{\|\boldsymbol{s}^2\|}\right) \text{ if } \alpha < 0 \end{cases} \tag{4.9}$$

where $\boldsymbol{s}^1 \times \boldsymbol{s}^2 = \alpha \hat{\mathrm{z}}$ and $\hat{\mathrm{z}}$ is the unit vector in the z-direction.

Clearly ω is a function of the output coordinates $\boldsymbol{u}$.

The next boundary point $\boldsymbol{u}^{b(i+1)}$ may then be found by solving the following optimization problem:

Problem (b):

$$\underset{\boldsymbol{u},\boldsymbol{w}}{\text{minimize}}\ \omega$$

$$\text{such that } \boldsymbol{v}^{\min} \leq \boldsymbol{v}(\boldsymbol{u},\boldsymbol{w}) \leq \boldsymbol{v}^{\max}$$

$$\boldsymbol{w}^{\min} \leq \boldsymbol{w} \leq \boldsymbol{w}^{\max}$$

$$\text{and } \boldsymbol{g}^{\min} \leq \boldsymbol{g}(\boldsymbol{u},\boldsymbol{v}(\boldsymbol{u},\boldsymbol{w}),\boldsymbol{w}) \leq \boldsymbol{g}^{\max}$$

and subject to the equality constraint

$$h(\boldsymbol{u},l) = (x - x^{bi})^2 + (y - y^{bi})^2 - l^2 = 0$$

Having solved *problem (b)*, $\boldsymbol{s}^{2i}$ with components s_x^{2i} and s_y^{2i} is precisely known and the new reference vector $\boldsymbol{s}^{1(i+1)}$ associated with the new boundary point $\boldsymbol{u}^{b(i+1)}$ can be determined as follows[3]:

$$\boldsymbol{s}^{1(i+1)} = \begin{bmatrix} \cos\left(\tan 2^{-1}\left(\frac{s_y^{2i}}{s_x^{2i}}\right) - \frac{\pi}{2} \right) \\ \sin\left(\tan 2^{-1}\left(\frac{s_y^{2i}}{s_x^{2i}}\right) - \frac{\pi}{2} \right) \end{bmatrix} \tag{4.10}$$

which defines a vector perpendicular to $\boldsymbol{s}^{2i}$ and pointing out of the workspace.

Since it has already been shown how a initial radiating point $\boldsymbol{u}^{b0}$ and reference vector $\boldsymbol{s}^{10}$ can be found, it follows that the boundary of the workspace ∂A may be mapped numerically by successively solving the above optimization *problem (b)* for $i = 0,1,2,\ldots$, and, each time using the solution to the previous problem as the starting point for the new optimization problem. Expression **(4.10)** is used to determine the associated reference vector for each new point.

It is evident that, if too large a chord length l is chosen in relation to the size of the workspace, entire sections of the workspace boundary may be missed. This must be borne in mind when selecting the chord length for a particular problem. As a starting choice, the chord length may be chosen as approximately one twentieth of the average actuator range. Should a greater workspace resolution be required, then a smaller chord length may be used, at the expense of computational time.

The algorithm is terminated when a specified maximum number of iterations is exceeded or when

$$\begin{aligned} \left\| \boldsymbol{u}^{bi} - \boldsymbol{u}^{b0} \right\| &\le l \text{ and} \\ \left\| \boldsymbol{u}^{bi} - \boldsymbol{u}^{b1} \right\| &\le l \end{aligned} \tag{4.11}$$

The area corresponding to the termination condition **(4.11)** is shown as the shaded region in Figure 4-3. Termination in this area is an indication of closure of the boundary, as is evident for the example depicted in the figure.

[3] The function tan2^{-1} has two input arguments and returns an angle in the range $[0,2\pi]$.

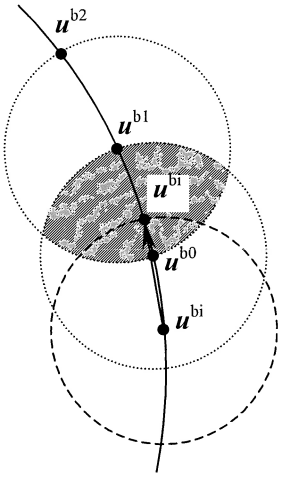

Figure 4-3: Termination of the algorithm

The algorithm formulation given above maps the workspace boundary in a counter-clockwise manner. In order to map in the clockwise direction it is necessary to modify the definition of ω **(4.9)** to

$$\omega = \begin{cases} \cos^{-1}\left(\dfrac{\boldsymbol{s}^1 \cdot \boldsymbol{s}^2}{\|\boldsymbol{s}^2\|}\right) \text{ if } \alpha \leq 0 \\ 2\pi - \cos^{-1}\left(\dfrac{\boldsymbol{s}^1 \cdot \boldsymbol{s}^2}{\|\boldsymbol{s}^2\|}\right) \text{ if } \alpha > 0 \end{cases} \qquad \textbf{(4.12)}$$

$$\text{where } \boldsymbol{s}^1 \times \boldsymbol{s}^2 = \alpha \hat{z}$$

and to change calculation **(4.10)** of the reference vector $\boldsymbol{s}^{1(i+1)}$ to

$$\boldsymbol{s}^{1(i+1)} = \begin{bmatrix} \cos\left(\tan 2^{-1}\left(\dfrac{s_y^{2i}}{s_x^{2i}}\right) + \dfrac{\pi}{2}\right) \\ \sin\left(\tan 2^{-1}\left(\dfrac{s_y^{2i}}{s_x^{2i}}\right) + \dfrac{\pi}{2}\right) \end{bmatrix} \qquad \textbf{(4.13)}$$

The same termination conditions **(4.11)** apply.

4.2.5 Bifurcation paths and bifurcation points

Whenever the manipulator moves along a trajectory such that motion is restricted in some direction, the manipulator is said to be moving along a *bifurcation path*. For the planar manipulators considered here this will occur either when two legs remain at extreme lengths while the third varies between extreme values or when one leg is at an extreme length and remains collinear with the working point while the others vary between extreme values. As is to be expected, the boundary of the maximal workspace consists of portions of bifurcation paths. The remaining portions of the bifurcation paths are internal bifurcation paths, which

correspond to positions within the workspace where the manipulator motion is restricted (Haug et al. [19]). Points of intersection of bifurcation paths in the output space usually[4] correspond to bifurcation points.

Bifurcation points are positions in the output space where the manipulator can branch or change directly from following one bifurcation path to another. For the accurate mapping of the workspace boundary, it is necessary to determine the exact position of bifurcation points along the workspace boundary. The precise determination of bifurcation points is done as described in section 2.2.6.

Snyman et al. [1] propose a new notation for labelling bifurcation points and curves which will be used here. The state of each leg i, i=1,2,3 is indicated by setting X_i=0 for a leg at its minimum length or X_i=1 for a leg at its maximum length. Setting X_i=- indicates a leg somewhere between its maximum and minimum lengths. The configuration of the platform at a specific bifurcation point is then indicated and labeled as a triplet enclosed in round brackets (X_1 X_2 X_3). For example, a bifurcation point (111) corresponds to a platform position where all of the legs are at their maximum length. Note that there may be more that one platform configuration for a given bifurcation labelling.

As mentioned above, it is also possible for a bifurcation point to occur when two legs, i and j, are at extreme lengths and either leg i or leg j is collinear with the working point of the platform. In this case the state of the remaining leg, k, is indicated by X_k=-. As an example, a bifurcation point labeled (10-) has leg 1 at a maximum length, leg 2 at a minimum and leg 3 at an intermediate length. Examination the bifurcation curves connecting with the bifurcation point will ascertain the identity of the leg collinear with the working point. One of these bifurcation curves will consist of only one leg at an extreme length and the other two at intermediate lengths. The leg at an extreme length is identified as the leg collinear with the working point.

The bifurcation curves are labeled in a similar way to the bifurcation points. Using the same conventions as outlined above for indicating the state of each leg, the bifurcation curve is labeled by a triplet enclosed in square brackets [X_1 X_2 X_3]. Thus, for example, the curve

[4] The exception are P-I points, discussed in section 4.2.6

connecting bifurcation point (011) to bifurcation point (111) is labeled [-11]. Along this curve legs 2 and 3 remain at their maximum lengths while leg 1 varies between its minimum and maximum lengths. Note that, in general, a leg need not vary between its maximum and minimum lengths for the platform to travel between two bifurcation points, but may also vary from an extreme length to some intermediate length, and then back to the same extreme length. This will occur when the bifurcation curve links two distinct bifurcation points, both corresponding to the same leg states.

Bifurcation curves corresponding to cases where one leg, i, remains at a constant extreme length, while one of the other two, j, varies between extreme values are labeled in a similar manner. In this case the third leg varies so that leg i and the working point of the platform remain collinear. As an example, bifurcation points (10-) and (11-) might be linked by the curve [1- -].

4.2.6 Projection-intersection points

At certain positions of the working point P within the workspace, and of particular interest, on the workspace boundary, it may occur that it is physically impossible for P to move directly to an adjacent workspace point. In such a case it is necessary for the working point to first move away from the initial point, so that a different orientation can be attained, before motion to the adjacent point can be achieved. Figure 4-4 shows an example of such an occurrence.

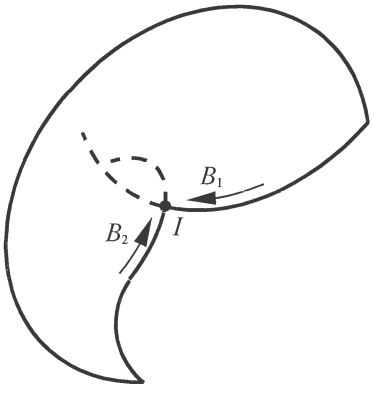

Figure 4-4: A projection-intersection point

For the example shown in this figure, point I can be approached moving along the workspace boundary in a continuous manner by bifurcation path B_1 from one side. From the other side point I can be approached along bifurcation path B_2. There exists, however, no normal bifurcation point at the *apparent* point of intersection of these two bifurcation curves, because the configuration reached from the one side differs in a discontinuous manner from the other

with respect to the orientation angle φ. If (x^I,y^I) denote the position of point I, then the two configurations at I are given by (x^I,y^I,φ^{B_1}) and (x^I,y^I,φ^{B_2}) where the respective orientation angles φ^{B_1} and φ^{B_2} may differ drastically. Point I does therefore not correspond to a normal bifurcation branch point where the paths B_1 and B_2 intersect in the x-y-φ space. Here intersection only *appears* to occur at I when the two paths are projected onto the xy-plane. An intersection point such as I will be named a *projection-intersection point* (P-I point).

Since in the optimization approach the previous solution point is used as the starting point for each optimization *problem (b)*, the paths traced correspond to the continuous physical bifurcation paths followed by the manipulator as it moves along the workspace boundary. Thus, if the manipulator follows a bifurcation path to the inside of the workspace, as depicted by the dashed lines in Figure 4-4, the algorithm will also trace this path to the inside of the workspace.

The occurrence of a P-I point on the boundary can be dealt with by mapping the workspace boundary near to the P-I point in both clockwise and anti-clockwise directions, as described in section 4.2.4. The coordinates of the P-I point can then be obtained by determining the projected point of intersection of these two boundary curves in x-y space.

If there are no P-I points or only one P-I point on the boundary then the workspace can be completely determined by mapping in clockwise and counterclockwise directions and then superposing the solutions to determine the intersection point.

The presence of more than one P-I points will be apparent if, after clockwise and counterclockwise mapping, there is no projected point of intersection of the two solutions. Some user interaction is then required in selecting different starting points so that all sections of the workspace boundary can be determined.

4.2.7 Voids in the workspace

Voids in the workspace can be detected exactly as described in section 2.2.7. The first boundary point and associated reference vector used by the chord method are directly obtained from the ray search.

4.2.8 Scaling the optimization problem

In any optimization problem, it is necessary to pay attention to the scaling of the problem. In the method presented here scaling is of particular importance as angles, measured in radians, and lengths, measured in arbitrary units, are combined in the formulation. Scaling should be carried out on the leg lengths to ensure that they are of approximately the same order as the angular measurement. If incorrectly scaled, the optimization algorithm may take a long time to converge, or may not even converge at all.

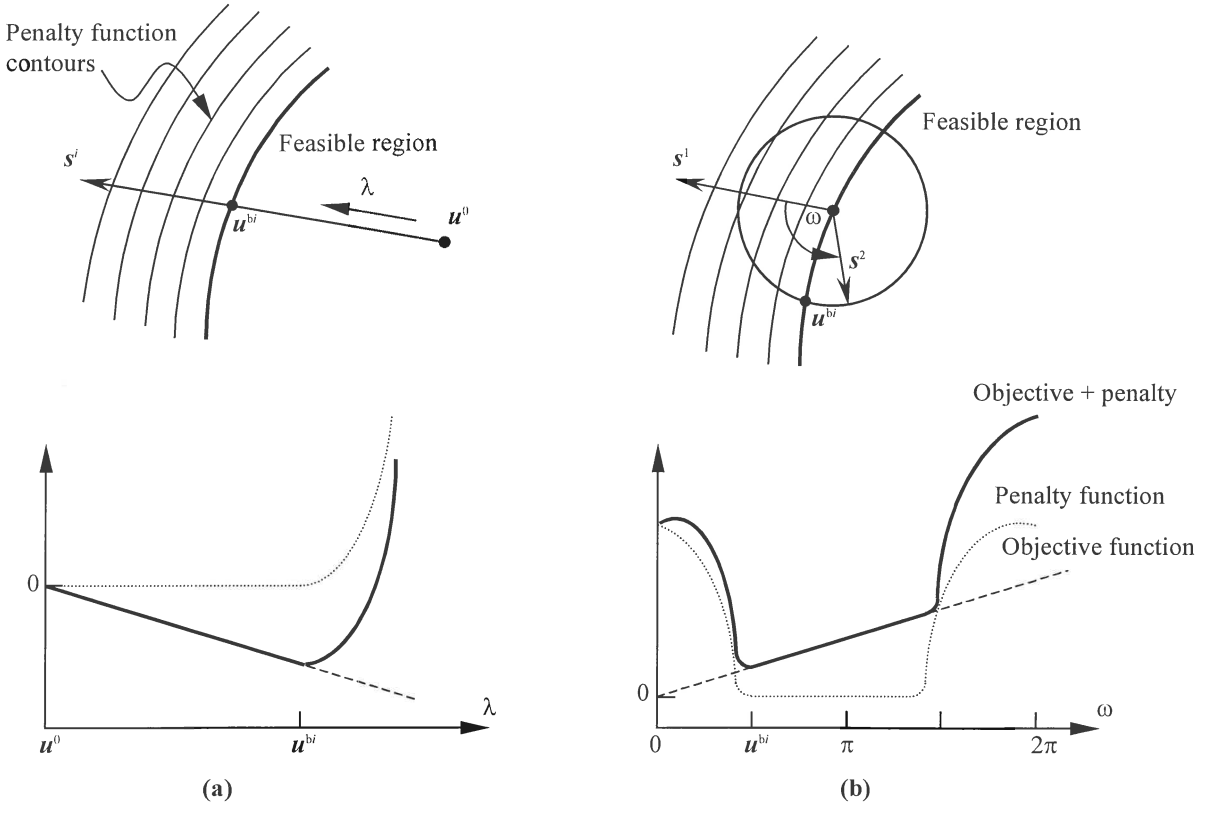

Figure 4-5: Penalty function approach for (a) the ray search and (b) the chord search

Another aspect of particular importance when using the chord method is the scaling of the objective function. This is due to the particular search geometry used in the chord approach. Figure 4-5 shows the differences between the variation in the penalty function along (a) ray search directions and (b) circular search directions. For the ray approach it is evident that the further the search point travels along the search ray into the infeasible region, the larger the contributions to the penalty function, due to the violations of the constraints, become.

For the chord approach the same is true along the circular constraint close to the workspace boundary. As the point travels further from the boundary along the circle, however, the penalty function levels off, and its gradient has less of an influence. In these cases it is

possible for the search point, during the optimization process, to travel past 0 and back to 2π. If this happens the search point may repeat its behavior and the algorithm may cycle without terminating. By correct scaling of the objective function this phenomenon can be eliminated. Another way of controlling this behavior is by limiting the maximum step size taken by the optimization algorithm.

4.2.9 The chord algorithm for mapping the boundary of a general workspace

To summarize, the algorithm consists of the following steps:

Step (i): Choose, or calculate using equation **(2.12)**, a suitable *initial radiating point* $\boldsymbol{u}_1^0$. Choose a chord length l. Choose K_{max}, the maximum number of iterations. Set i =1.

Step (ii): For an arbitrarily chosen unit search vector $\boldsymbol{s}^{10}$, solve *Problem (a)* to give an initial boundary point $\boldsymbol{u}^{b0}$. Note that $\boldsymbol{s}^{10}$ will always point out of the workspace and is thus a valid reference vector to be associated with the initial boundary point. Record the coordinates of $\boldsymbol{u}^{b0}$ as well as the constraints active at this point.

Step (iii): Solve *Problem (b)*, with ω defined by **(4.9)** or **(4.12)** to find the next boundary point $\boldsymbol{u}^{bi}$. Record the coordinates of $\boldsymbol{u}^{bi}$ as well as the constraints active at this point.

Step (iv): Determine the new reference vector $\boldsymbol{s}^{1i}$ using expression **(4.10)** or **(4.13)**.

Step (v): If condition **(4.11)** is violated or if $i<K_{max}$ then set $i=i+1$ and proceed to Step (iii).

Step (vi): Scan through the calculated boundary points and associated active constraint information to determine where bifurcation points occur, and calculate these bifurcation points. Insert the coordinates of these points at the correct place in the sequence of boundary points. *(End of procedure).*

4.3 Application to the two degree of freedom manipulator

The two degree of freedom manipulator, shown in Figure 1-5 of chapter 1 and considered in chapter 2, will be used to illustrate the proposed chord method. As before, the origin is at point A and the coordinates of point B are (4,0). The various length limits considered are as given in Table 2-1. One part of the workspace of manipulator L1, mapped using the chord method, is shown in Figure 4-6. Each dot in the figure represents a point determined along the workspace boundary. These are, excluding the bifurcation points, equally distributed along the workspace boundary in contrast to the distribution given by the modified ray method. *Constant chords* of the boundary have thus been determined.

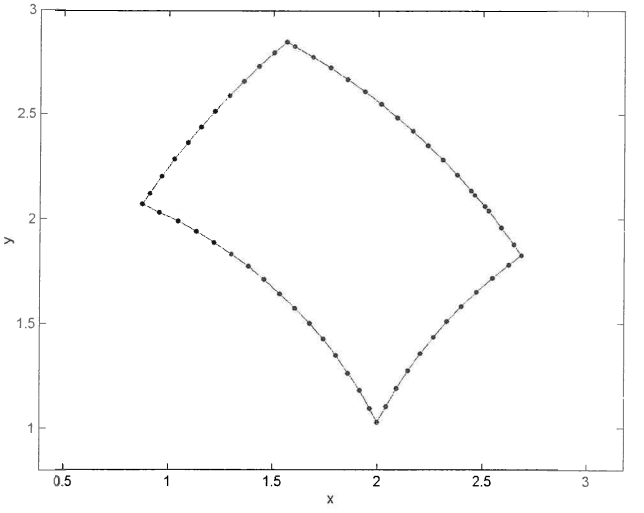

Figure 4-6: Part of the workspace of manipulator L1 determined using the chord method

Figure 4-7(a), (b) and (c) respectively show the workspaces of manipulators L1, L2 and L3. With the exception of the workspace L1, which requires two runs of the algorithm with different starting points to determine the top and bottom sections of the workspace, the workspaces were determined completely automatically. This demonstrates a definite advantage over the modified ray method (see section 2.3)

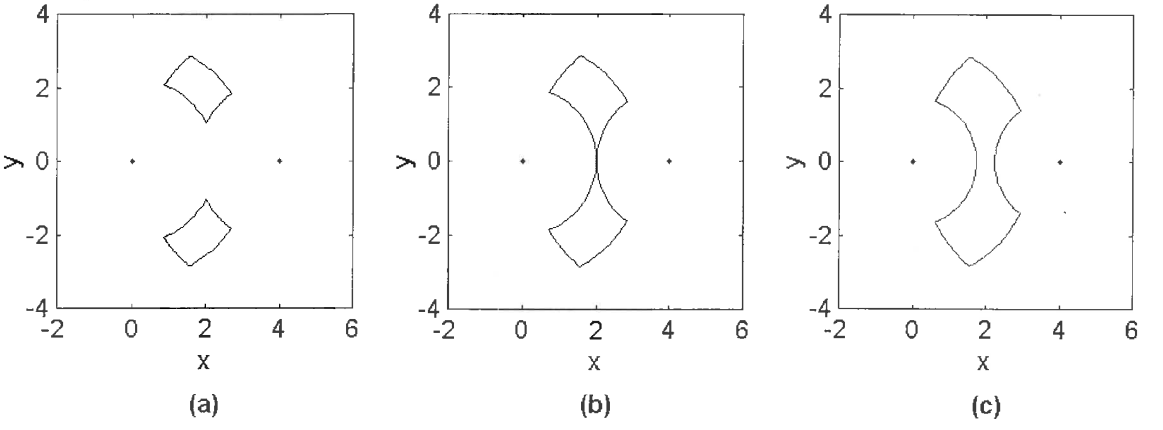

Figure 4-7: Workspaces of (a) L1, (b) L2 and (c) L3

4.4 Application to a general three degree of freedom planar manipulator

4.4.1 Geometry of the manipulator

A general three degree of freedom planar parallel manipulator is shown in Figure 4-8.

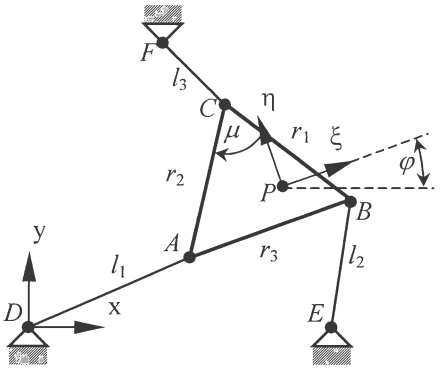

Figure 4-8: General planar Stewart platform

The manipulator consists of a mobile triangular platform with side lengths r_1, r_2 and r_3 and angle μ between sides r_1 and r_2. Three linear actuators l_1, l_2 and l_3 connect the vertices of the platform to the ground by means of revolute joints $A - F$. In what follows, it is assumed that the origin of the global coordinate system x-y is fixed at point D and that point E lies on the x-axis (x_D=y_D=y_E=0). This is only done for convenience and does not affect the general applicability of the algorithm in any way. The local coordinate system ξ-η is fixed to the platform at point P, the working point of the platform. The orientation of the platform φ is defined as the angle between the x-axis of the global coordinate system and line going from A to B.

With respect to the definitions given in section 4.2.1, the actuator leg lengths form the input coordinates $\boldsymbol{v} = [l_1, l_2, l_3]^T$. The global coordinates of the working point P form the output coordinates, i.e. $\boldsymbol{u} = [x_P, y_P]^T$. The rotation angle of the platform is the only intermediate coordinate, i.e. $w = \varphi$. The generalized coordinates for the platform are given by:

$$\begin{aligned} \boldsymbol{q} &= [\boldsymbol{u}^T, \boldsymbol{v}^T, w]^T \\ &= [x_P, y_P, l_1, l_2, l_3, \varphi]^T \end{aligned} \tag{4.14}$$

Note that if r_2=0 and y_P=0 then the above reduces to the planar Stewart platform which has already been analysed in Chapter 3.

4.4.2 Constraint equation formulation

In this analysis only the kinematic and leg length constraints will be considered, although there is no reason why other constraints, such as those used in the analysis of Chapter 3, cannot be used.

4.4.2.1 Kinematic Constraints

In the platform local coordinate system, the coordinates of the leg-platform attachment points A, B and C are (ξ_A,η_A), (ξ_B,η_B) and (ξ_C,η_C) respectively. The standard transformation from a local coordinate system with origin at (x_P,y_P) and orientation φ to the global coordinate system is

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_P \\ y_P \end{bmatrix} + \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} \tag{4.15}$$

Using transformation **(4.15)** the global coordinates of points A, B and C can easily be obtained. The inverse kinematics of the platform can then be performed to give the actuator leg lengths in terms of the position and orientation of the platform:

$$\begin{aligned} l_1^2 &= (x_A(x_P,y_P,\varphi)-x_D)^2 + (y_A(x_P,y_P,\varphi)-y_D)^2 \\ l_2^2 &= (x_B(x_P,y_P,\varphi)-x_E)^2 + (y_B(x_P,y_P,\varphi)-y_E)^2 \\ l_3^2 &= (x_C(x_P,y_P,\varphi)-x_F)^2 + (y_C(x_P,y_P,\varphi)-y_F)^2 \end{aligned} \tag{4.16}$$

This can be rewritten in the standard form for the kinematic constraint equations as:

$$\boldsymbol{\Phi}(\boldsymbol{u},\boldsymbol{v},w) = \begin{bmatrix} v_1^2 - (x_A(\boldsymbol{u},w)-x_D)^2 - (y_A(\boldsymbol{u},w)-y_D)^2 \\ v_2^2 - (x_B(\boldsymbol{u},w)-x_E)^2 - (y_B(\boldsymbol{u},w)-y_E)^2 \\ v_3^2 - (x_C(\boldsymbol{u},w)-x_F)^2 - (y_C(\boldsymbol{u},w)-y_F)^2 \end{bmatrix} = \mathbf{0} \tag{4.17}$$

4.4.2.2 Leg Length Constraints

The constraints on the leg lengths are:

$$0 < l_i^{\min} \le l_i \le l_i^{\max}, \; i = 1,2,3 \tag{4.18}$$

From **(4.16)** the explicit expressions for $\boldsymbol{v}$ are:

$$\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{u},w) = \begin{bmatrix} \sqrt{(x_A(\boldsymbol{u},w)-x_D)^2 + (y_A(\boldsymbol{u},w)-y_D)^2} \\ \sqrt{(x_B(\boldsymbol{u},w)-x_E)^2 + (y_B(\boldsymbol{u},w)-y_E)^2} \\ \sqrt{(x_C(\boldsymbol{u},w)-x_F)^2 + (y_C(\boldsymbol{u},w)-y_F)^2} \end{bmatrix} \tag{4.19}$$

These may be written in the standard form:

$$\boldsymbol{v}^{\min} \le \boldsymbol{v}(\boldsymbol{u},w) \le \boldsymbol{v}^{\max} \tag{4.20}$$

where $\boldsymbol{v}^{\min} = [l_1^{\min}, l_2^{\min}, l_3^{\min}]^T$, $\boldsymbol{v}^{\max} = [l_1^{\max}, l_2^{\max}, l_3^{\max}]^T$ and with $\boldsymbol{u}$ and w specified, $\boldsymbol{v}(\boldsymbol{u},w)$ is given by **(4.19)**.

4.4.3 Results for the general planar platform

Numerical values of the dimensions for the different manipulator geometries considered in this chapter are given in Table 4-1. These are purposely chosen to correspond to the manipulators considered by Merlet et al. [15].

Manipulator	r_1	r_2	r_3	x_E	x_F	y_F	μ
M1	25	25	25	20	0	10	60°
M2[5]	20.839	17.045	16.549	15.91	0	10	52.74°
M3	25	25	25	20	10	17.23	60°
M4	2	2	2	10	5	8.66	60°

Table 4-1: General planar Stewart platform manipulator dimensions

Workspaces for these manipulators determined using the chord method, and with the working point positioned at the centroid of the platform, are shown in Figure 4-9. In all cases the leg limits used are $2 \le l_1 \le 8$, $5 \le l_2 \le 25$ and $10 \le l_3 \le 25$. P-I points are indicated by an i. The workspace of M3, shown Figure 4-9(c), is of particular interest as it is both highly non-convex and is an example of a workspace boundary containing two P-I points, indicated by i1 and i2.

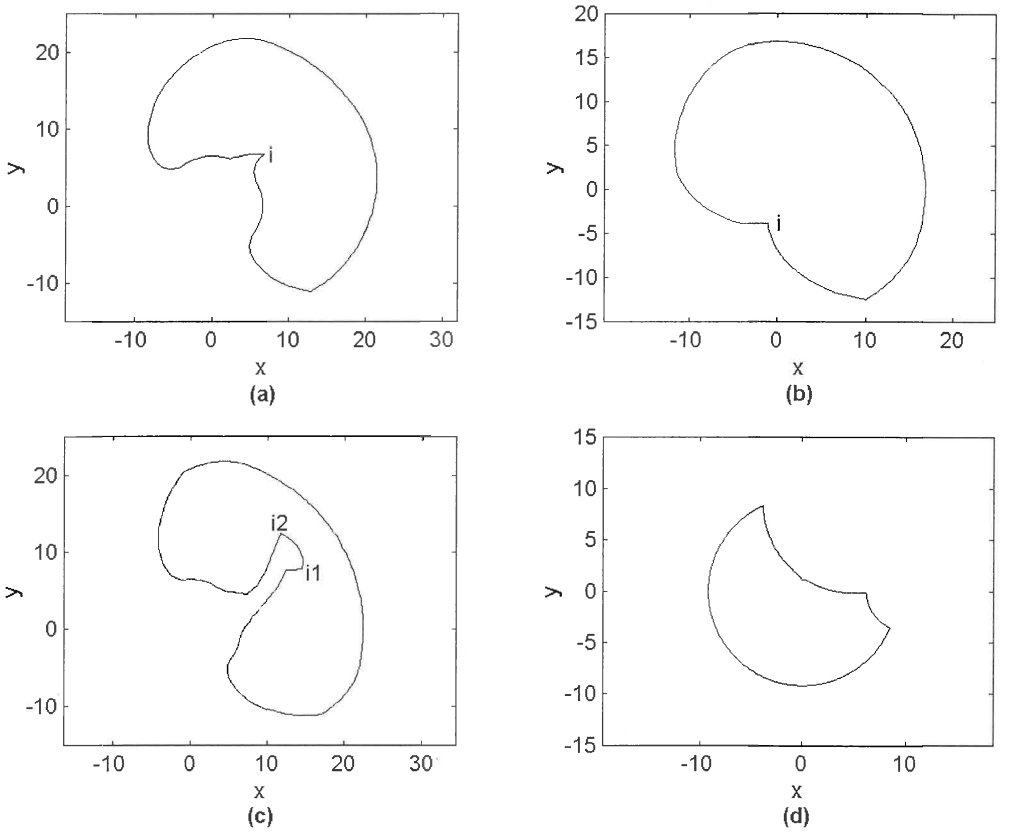

Figure 4-9: Maximal workspaces of (a) M1, (b) M2, (c) M3 and (d) M4

[5] In Merlet et al. [15] r_3=16.54.

4.5 Preliminary evaluation of the chord method

It has been shown in this chapter that the chord method is capable of determining extremely non-convex workspaces. The chord method has the advantage over the modified ray approach that less user interaction is required during the calculation of workspace boundary. In its present form, the chord algorithm successfully maps the workspace boundary from a starting point until a P-I point is reached. The workspace boundary can be automatically mapped if there are no P-I points or one P-I points present. If there are two or more P-I points then some user interaction is required in determining the workspace boundary.

Chapter 5: REDUNDANCY AND ITS TREATMENT BY BOTH THE RAY AND CHORD APPROACHES

5.1 Introduction

In the previous chapters the optimization approach has been applied to manipulators where the extreme reach of the parallel manipulator is dependent on the orientation of the platform. In this chapter a case will be studied where there are no unique orientations of the platform for certain portions of the workspace boundary. For these sections of the boundary the orientation is effectively redundant in determining the extreme reach of the manipulator. Such manipulators, *defined here as redundant manipulators*, are used by Merlet et al. [15] in illustrating the geometrical method. They make no mention of the special behavior associated with such manipulators.

A slight modification to the optimization approach, which is equally applicable to the chord and ray methods, is required in order to determine the workspaces of such redundant manipulators. In this chapter this modified approach is presented and the workspaces of the manipulators studied by Merlet et al. [15] are determined.

5.2 Geometry of the parallel manipulator exhibiting redundant behavior

The dimensions of the manipulator are defined in Figure 5-1.

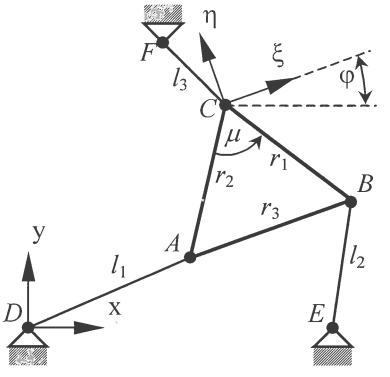

Figure 5-1: Redundant manipulator platform geometry

The geometry of the manipulator is the same as that considered in the previous chapter. The local coordinate system ξ-η is fixed at point *C*, which is also chosen as the working point of the platform. The orientation of the platform φ is defined as the angle between the x-axis of

the global coordinate system and line going from A to B[6]. It is once again assumed that the origin of the global coordinate system x-y is fixed at point D and that point E lies on the x-axis (x_D=y_D=y_E=0).

The same type of redundant behavior would be obtained if the working point were fixed at point A or point B.

The actuator leg lengths once more form the input coordinates $v=[v_1,v_2,v_3]^T=[l_1,l_2,l_3]^T$. The global coordinates of the working point, point C, now form the output coordinates, i.e. $\boldsymbol{u}=[u_1,u_2]^T=[x_C,y_C]^T$. The rotation angle of the platform is the only intermediate coordinate, i.e. $w=\varphi$. The generalized coordinates for the platform are given by:

$$\begin{aligned}\boldsymbol{q}&=[\boldsymbol{u}^T,\boldsymbol{v}^T,w]^T \\ &=[x_C,y_C,l_1,l_2,l_3,\varphi]^T\end{aligned} \tag{5.1}$$

Numerical values of the dimensions for the different manipulator geometries considered in this chapter, and once more taken from Merlet et al [15] are given in Table 5-1.

Manipulator	r_1	r_2	r_3	x_E	x_F	y_F	μ
M1	25	25	25	20	0	10	60°
M2[7]	20.839	17.045	16.549	15.91	0	10	52.74°
M3	25	25	25	20	10	17.23	60°
M4	2	2	2	10	5	8.66	60°

Table 5-1: Redundant Stewart platform manipulator dimensions

5.3 Constraint equation formulation

The kinematic and leg length constraints can be derived using the same procedure as that given in section 4.4.2.

5.3.1 Kinematic constraints

In the local coordinate system, the coordinates of the leg-platform attachment points A, B and C are (ξ_A,η_A), (ξ_B,η_B) and (ξ_C,η_C)=(0,0) respectively. The transformation from a local coordinate system located at (x_C,y_C) and orientation φ to the global coordinate system is

[6] Note that this definition differs from that chosen by Merlet who chooses the orientation angle ϕ_M as the angle between the x-axis and the line going from C to A. For manipulators with μ=60° and r_1=r_2=r_3 the transformation required is ϕ=ϕ_M-μ-180°=ϕ_M-240°.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_C \\ y_C \end{bmatrix} + \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} \tag{5.2}$$

Using transformation **(5.2)** the global coordinates of points A, B and C can easily be obtained. Note that since $(\xi_C, \eta_C)=(0,0)$ the global coordinates of C are not dependant on the orientation of the platform φ. The inverse kinematics of the platform can then be performed to give the actuator leg lengths in terms of the position and orientation of the platform:

$$\begin{aligned} l_1^2 &= (x_A(x_C, y_C, \varphi) - x_D)^2 + (y_A(x_C, y_C, \varphi) - y_D)^2 \\ l_2^2 &= (x_B(x_C, y_C, \varphi) - x_E)^2 + (y_B(x_C, y_C, \varphi) - y_E)^2 \\ l_3^2 &= (x_C - x_F)^2 + (y_C - y_F)^2 \end{aligned} \tag{5.3}$$

This can be rewritten in the standard form for the kinematic constraint equations as:

$$\boldsymbol{\Phi}(\boldsymbol{u}, \boldsymbol{v}, w) = \begin{bmatrix} v_1^2 - (x_A(\boldsymbol{u}, w) - x_D)^2 - (y_A(\boldsymbol{u}, w) - y_D)^2 \\ v_2^2 - (x_B(\boldsymbol{u}, w) - x_E)^2 - (y_B(\boldsymbol{u}, w) - y_E)^2 \\ v_3^2 - (u_1 - x_F)^2 - (u_2 - y_F)^2 \end{bmatrix} = \boldsymbol{0} \tag{5.4}$$

5.3.2 Leg length constraints

The constraints on the leg lengths are:

$$0 < l_i^{\min} \le l_i \le l_i^{\max}, i = 1,2,3 \tag{5.5}$$

From **(5.4)** the explicit expressions for $\boldsymbol{v}$ are:

$$\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{u}, w) = \begin{bmatrix} \sqrt{(x_A(\boldsymbol{u}, w) - x_D)^2 + (y_A(\boldsymbol{u}, w) - y_D)^2} \\ \sqrt{(x_B(\boldsymbol{u}, w) - x_E)^2 + (y_B(\boldsymbol{u}, w) - y_E)^2} \\ \sqrt{(u_1 - x_F)^2 + (u_2 - y_F)^2} \end{bmatrix} \tag{5.6}$$

These may be written in the standard form:

$$\boldsymbol{v}^{\min} \le \boldsymbol{v}(\boldsymbol{u}, w) \le \boldsymbol{v}^{\max} \tag{5.7}$$

where $\boldsymbol{v}^{\min} = [l_1^{\min}, l_2^{\min}, l_3^{\min}]^T$, $\boldsymbol{v}^{\max} = [l_1^{\max}, l_2^{\max}, l_3^{\max}]^T$ and with $\boldsymbol{u}$ and w specified, $\boldsymbol{v}(\boldsymbol{u}, w)$ is given by **(5.6)**.

5.4 Redundancy on the workspace boundary

For a redundant type manipulator, as described in section 5.2, there exist portions of the workspace boundary for which the formulations of *problem (i)* and *problems (a)* and *(b)* do not have unique solutions. Along these boundaries, a range of platform orientations are possible. The reason for this redundancy is that the working point of the platform coincides with a leg attachment point and thus the extreme objective function value may not implicitly

[7] In Merlet et al. [15] r_3=16.54

be affected by the orientation of the platform, as is otherwise the case. For the ray approach, a unique solution may be obtained by altering the objective function when non-uniqueness is detected. The objective function then becomes:

$$\underset{\boldsymbol{u},w}{\text{maximize}} \left\| \boldsymbol{u} - \boldsymbol{u}^0 \right\| \pm \varphi \tag{5.8}$$

For the chord approach the objective function is similarly modified:

$$\underset{\boldsymbol{u},w}{\text{minimize}}\, \omega \pm \varphi \tag{5.9}$$

In both cases, the modified objective function forces the platform to attain either a maximum clockwise or counter-clockwise rotation depending on the sign chosen, without affecting the reach. Note that it is possible for even the modified objective function not to have a unique solution in some cases. This will occur when the manipulator can rotate completely about a point without violating any constraints.

Non-unique boundary portions can be detected by examining the constraints active on the workspace boundary. When the only active inequality constraint is the constraint limiting the length of the leg *associated with* the working point of the platform, then non-uniqueness will be present. It is possible to have only one active inequality constraint and still have a unique solution, as is the case when the only active constraint is associated with one of the other legs (in this case leg 1 or 2).

Redundancy, or non-uniqueness, does not usually cause a severe problem in the determination of the workspace boundary, and using the standard objective function, the correct boundary may still be determined in x-y space. However since the uniqueness of the orientation angle of the platform affects the uniqueness of the bifurcation path which the platform follows in x-y-φ space, redundancy may affect the bifurcation path followed and thus the projection on the x-y-plane in giving the boundary of the workspace.

5.5 Bifurcation paths in x-y space

To illustrate the effects of redundancy on determining the workspace boundary, consider the workspace of manipulator M3 with leg limits $5 \le l_i \le 20$, i=1,2,3, shown in Figure 5-2, which has been determined using the ray method with $\boldsymbol{u}_1^0 = (10,\ 25)^T$ and the modified objective function **(5.8)**.

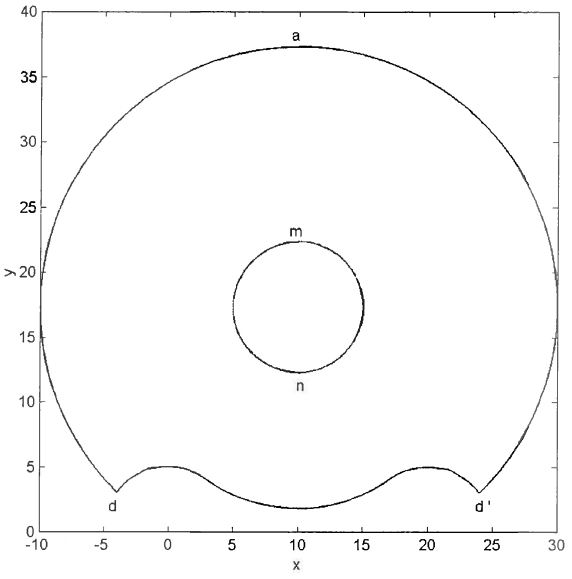

Figure 5-2: Maximal workspace of M3

The manipulator is a symmetrical about the line x=10. It is thus only necessary to map one side of the workspace boundary to determine the full maximal workspace. The symmetry has further implications on the character of the workspace. Gosselin [37] notes that the stiffness of such a manipulator becomes very low when the platform is positioned symmetrically near the center of the workspace. Portions of the workspace boundary where non-unique behavior occurs are indicated in Figure 5-2 as lines d-a-d' and m-n-m.

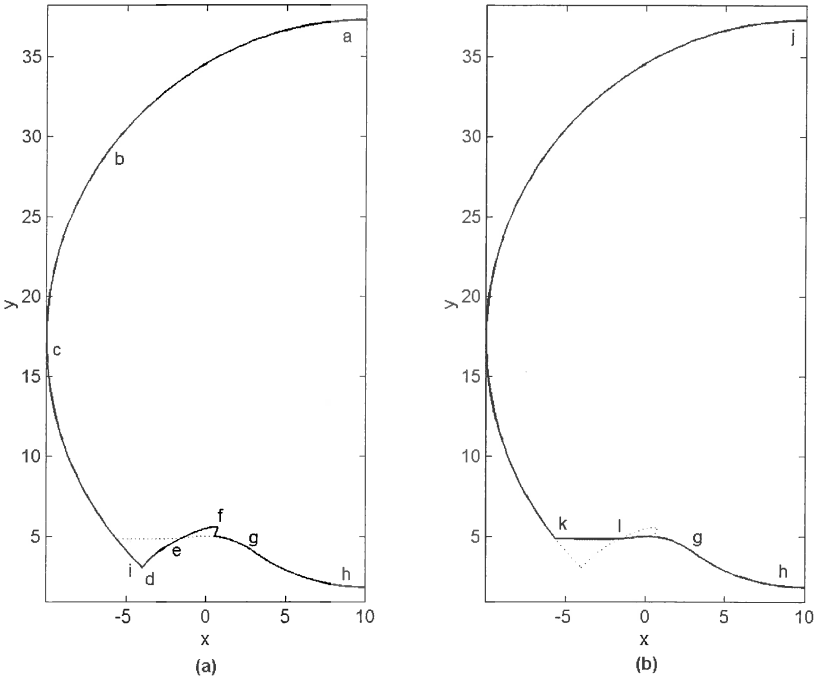

Figure 5-3: Bifurcation paths in x-y space (a)+φ modification and (b) -φ modification

Figure 5-3 shows the calculated projected bifurcation paths followed by the platform in x-y space as it moves along the workspace boundary from a to h and j to h. These were calculated using the modified ray method with the modified objective function. The bifurcation path followed by the platform and projected as the workspace boundary is uniquely decided by the orientation of the platform, as it moves from a (or j) and approaches the bifurcation point k. After point k two different boundary curves are mapped depending on the form of the modified objective function **(5.8)**. Figure 5-3(a) shows the path followed for a +φ modification and (b) the path followed for a -φ modification. If the method is applied without the objective function modification, path (b) will always be followed. Superpositioning of boundaries (a) and (b) gives the maximal workspace shown in Figure 5-2.

5.6 Detailed analysis of boundary bifurcation paths in x-y-φ space

To illustrate why the platform follows different bifurcation paths, consider Figure 5-4, which shows a plot of the platform orientation φ [rad] versus the search direction θ [rad] with $\boldsymbol{u}^0=(10, 25)^T$. The bifurcation points and curves are labeled using the notation proposed by Snyman et al. **[1]**. As the search direction θ varies from 0 to π the nature of optimization *Problem (i)* changes. Various regions corresponding to the different classes of solution may be identified.

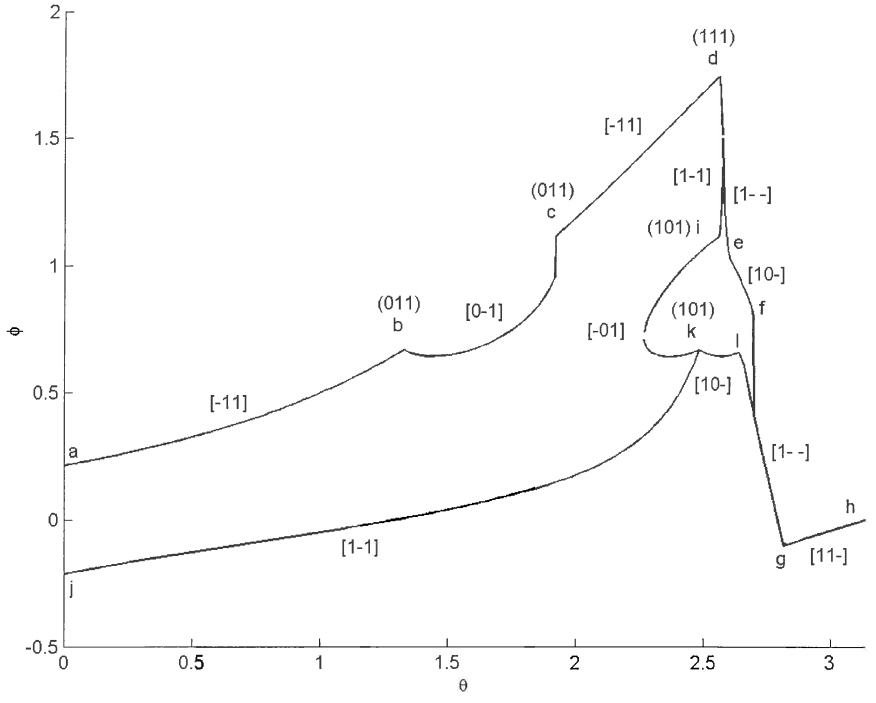

Figure 5-4: Bifurcation paths in θ-φ space

Figure 5-5 gives a two-dimensional representation of how the ray method objective function[8] $f = -\|\boldsymbol{u} - \boldsymbol{u}^0\|$ in the specified direction θ might look in the *f*-φ plane for a number of fixed search directions θ as one moves from left to right in Figure 5-4.

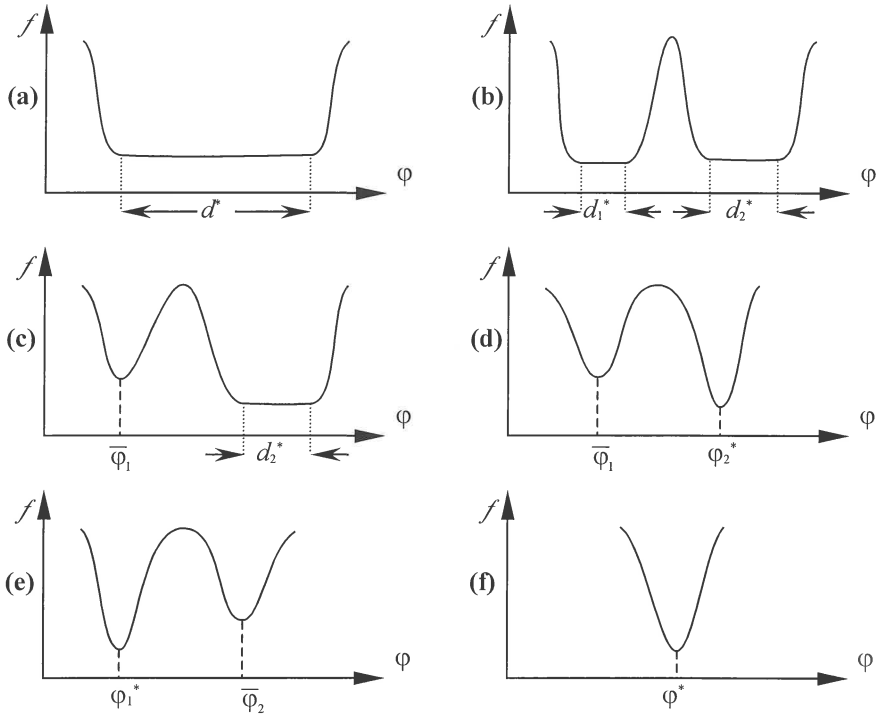

Figure 5-5: Objective function sections for different fixed θ values

With reference to Figure 5-4, Figure 5-5(a) corresponds to a search direction in the region of $\theta_a \leq \theta \leq 2.3$.[9] In this case the original objective function *f*, corresponding to $\|\boldsymbol{u} - \boldsymbol{u}^0\|$ in the specified direction, does not have a unique minimum. Instead there is flat solution plain of width d^*. Note that over this region of φ the x and y coordinates of the workspace boundary (x_C, y_C) remain constant.

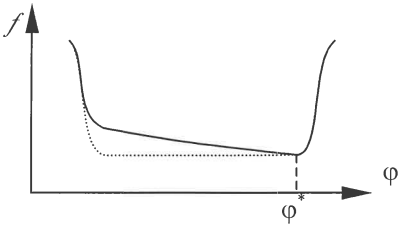

Figure 5-6: Objective function modification

The modified objective function **(5.8)** has the effect of sloping the floor of the valley and thus forcing the optimization problem to have a unique solution as shown in Figure 5-6.

[8] Negative sign here because optimization algorithms usually perform a minimization and consequently one is conditioned to visualize the minimization problem.
[9] The value of θ for any point n will be denoted θ_n.

Depending on whether a positive or negative modification is used, the most clockwise or anti-clockwise orientation of the manipulator may be determined.

From θ=2.3 to θ_k the solution plain splits into two separate plain solution regions of width d_1^* and d_2^* respectively as shown in Figure 5-5(b). Once again the x and y workspace boundary coordinates remain constant over both regions, while the platform orientation is non-unique. The four bounding values of the platform orientation can be determined by using the objective function modification and selecting a suitable starting point. Note that if the starting point is chosen to lie in one plain, then a solution will generally be found inside that plain.

Figure 5-5(c) gives a representation of the third case ($\theta_k \leq \theta \leq \theta_d$) in which one of the plains has become a valley with one strong local minimum $\overline{\varphi}_1$ and a non-unique global solution region of width d_2^* is the other remaining plain. Once again the bounds of the non-unique region may be determined using the objective function modification. There are distinct coordinates (x_C, y_C, φ) corresponding to $\overline{\varphi}_1$ and constant (x_C, y_C) coordinates but varying φ corresponding to the region of width d_2^*.

In the region $\theta_d \leq \theta \leq \theta_f$ the remaining plain also becomes a valley and there is either one strong local minimum $\overline{\varphi}_1$ and one global minimum φ_2^* as shown in Figure 5-5(d) or the inverse case occurs of a global minimum at φ_1^* and a strong local minimum at $\overline{\varphi}_2$ as shown in Figure 5-5(e). The final case ($\theta_f \leq \theta \leq \theta_h$) is shown in Figure 5-5(e) where one unique solution φ^* exists.

Since in the method used, the starting points for consecutive sub-problems are chosen as solutions to the previous sub-problems, and θ is always increasing, it is evident that only one complete valley can be mapped for every run of the program. The valley to be followed can be chosen by changing the sign of the objective function modification used in the non-unique region. This forces the platform into an extreme clockwise or counter-clockwise orientation corresponding to path j-k or a-b-c-d respectively. From k path k-l-g-h can be followed and from d path d-e-f-g-h. Since we are unsure from the outset as to which portions of each valley correspond to the global minimum, it becomes necessary to map both extremes and then

compare the results to find the maximal workspace boundary. Note the jump at f, which corresponds to the solution moving from a local minimum valley to the global minimum.

5.7 Maximal workspace determination

The maximal workspaces of the parallel manipulators listed in Table 5-1 can be determined by applying the method for determining non-convex workspaces, described in the previous chapters, to equations (5.4) with constraints (5.7). The maximal workspaces of M1 and M3 are shown in Figure 5-7. These were determined using the modified ray approach with the modified objective function and leg length limits $2 \le l_1 \le 8$, $5 \le l_2 \le 25$ and $10 \le l_3 \le 25$.

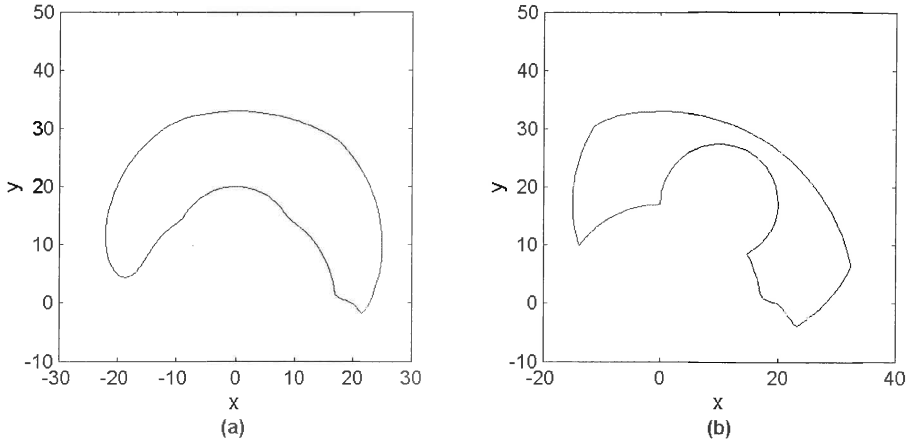

Figure 5-7: Maximal workspaces of (a) M1 and (b) M3

Figure 5-8 shows the maximal workspaces of M2 and M4, which were determined using the chord method with the objective function modification and leg length limits $2 \le l_1 \le 8$, $5 \le l_2 \le 25$ and $10 \le l_3 \le 25$.

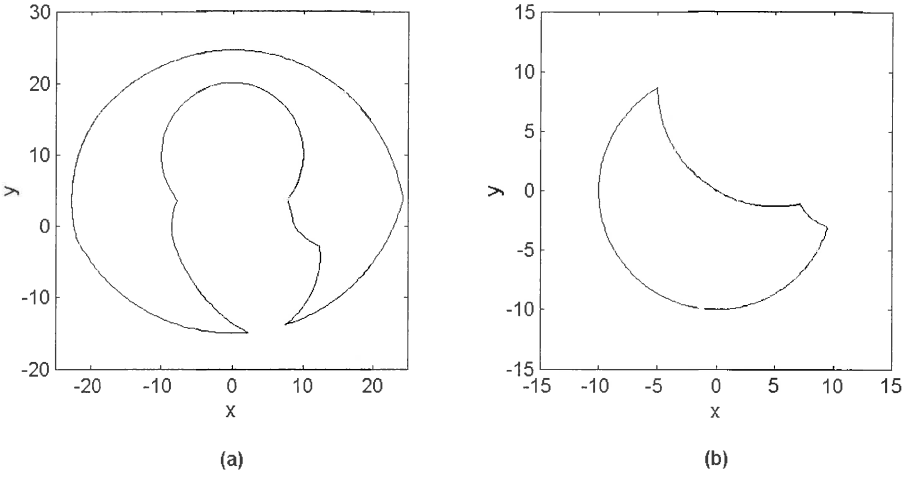

Figure 5-8: Maximal workspaces of (a) M2 and (b) M4

5.8 Conclusion

The workspaces determined are highly non-convex, proving the effectiveness of the described optimization methods. Furthermore, except for the workspace M2, they correspond exactly with the workspaces determined by Merlet et al. [15]. The reason for the discrepancy between the workspaces for M2 is that the dimensions of the manipulator are not accurately given in the paper of Merlet et al. [15]. The basic forms of the respective workspaces are however very similar.

Chapter 6: CONCLUSION

One of the main advantages of the optimization approach, previously proposed by Snyman et al. [1], is the ease with which constraints of all forms can be implemented and handled. In this respect the optimization approach is superior to the continuation method of Jo and Haug [26,27].

The advantage that the continuation method has over other existing methods, particularly the geometrical method (Gosselin and Angeles [6], Merlet et al [15]), is that it is a numerical method which is generally applicable to both serial and parallel manipulators. Snyman et al. [1] have shown that the optimization approach also possesses this property. Indeed the optimization method should be even more generally applicable to manipulators of hybrid design. In this study, the original ray approach of Snyman et al. [1] has successfully been modified to determine most non-convex workspaces. Another optimization approach, the chord approach, has been proposed and shown to be even more reliable in determining non-convex workspaces. Both methods, embodied in practical interactive computer systems, allow for the easy determination of convex and non-convex workspaces.

Work has so far primarily been concentrated on the determination of maximal workspaces. Other workspaces, particularly dextrous, total orientation and constant orientation workspaces are also of practical importance. Using the optimization approach, Du Plessis [2] has computed some of these workspaces for planar and spatial Stewart platforms and has thus shown that such workspaces can also be calculated using the optimization approach. Extension of the current methods for non-convexity to allow for the calculation of these workspaces will be of great practical use.

Although much success has been attained with the optimization approach, further study is needed before the method can be seen as truly automated. One particular aspect that needs to be addressed is the inability of the optimization algorithms to trace the extreme workspace boundary past a projection-intersection point. The main difficulty in dealing with such a situation is that there is no indication of the occurrence of such an P-I point until breakdown of the algorithm occurs.

Complete automation of the algorithm would lead to many possibilities for further work to be carried out on the geometrical optimization of manipulators. Some problems that could be addressed are maximization of the manipulator workspace and the determination of the geometry of a manipulator for a prescribed workspace.

Du Plessis [2] has already shown, for the specific case of a 6-3 Stewart platform, that the planar optimization approach can easily be extended to determine the spatial workspace. It is therefore believed that the current planar system can also be refined to determine the workspaces of spatial Stewart platforms of more varied designs. Future research will therefore be directed at developing such a general system for the characterization of spatial workspaces.

Appendix A: SINGULARITY ANALYSIS OF THE PLANAR STEWART PLATFORM

Consider the Stewart platform shown in Figure 3-1 and the vector of coordinates $\boldsymbol{q} = [x_P, y_P, \varphi_P, l_1, l_2, l_3]^T$. Writing expressions **(3.3)** in the standard form of the constraint equations, and substituting the specifications for the normalized design, given in Table 3-1 gives:

$$\boldsymbol{\Phi}(\boldsymbol{q}) = \begin{bmatrix} l_1^2 - (x_P - \cos\varphi_P + 1)^2 - (y_P - \sin\varphi_P)^2 \\ l_2^2 - (x_P - \cos\varphi_P - 1)^2 - (y_P - \sin\varphi_P)^2 \\ l_3^2 - (x_P + \cos\varphi_P - 2)^2 - (y_P + \sin\varphi_P)^2 \end{bmatrix} = \mathbf{0} \tag{A.1}$$

Differentiating **(A.1)** with respect to time yields

$$\boldsymbol{\Phi}_q \dot{\boldsymbol{q}} = \mathbf{0} \tag{A.2}$$

where

$$\boldsymbol{\Phi}_q = \left[\begin{array}{ccc|ccc} -(x_P - \cos\varphi_P + 1) & -(y_P - \sin\varphi_P) & -\sin\varphi_P(x_P - \cos\varphi_P + 1) + \cos\varphi_P(y_P - \sin\varphi_P) & l_1 & 0 & 0 \\ -(x_P - \cos\varphi_P - 1) & -(y_P - \sin\varphi_P) & -\sin\varphi_P(x_P - \cos\varphi_P - 1) + \cos\varphi_P(y_P - \sin\varphi_P) & 0 & l_2 & 0 \\ -(x_P + \cos\varphi_P - 2) & -(y_P + \sin\varphi_P) & \sin\varphi_P(x_P + \cos\varphi_P - 2) - \cos\varphi_P(y_P + \sin\varphi_P) & 0 & 0 & l_3 \end{array}\right] = [\boldsymbol{A} \,\vdots\, \boldsymbol{B}] \tag{A.3}$$

Using the above partitioning **(A.2)** can now be separated as follows:

$$\boldsymbol{A}\begin{bmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{\varphi}_P \end{bmatrix} = -\boldsymbol{B}\begin{bmatrix} \dot{l}_1 \\ \dot{l}_2 \\ \dot{l}_3 \end{bmatrix} \tag{A.4}$$

In accordance with Gosselin and Angeles **[38]**, a distinction can be made between three different types of singularities:

Type I singularities occur when det($\boldsymbol{B}$)=0. Mathematically, this condition leads to l_1=0 or l_2=0 or l_3=0. However, since the actuators have a finite range of motion, this type of singularity will occur when one of the actuator legs reaches its minimum or maximum length (Sefrioui and Gosselin **[39]**):

$$l_i = l_i^{\min} \text{ or } l_i = l_i^{\max}, \quad i = 1,2,3 \tag{A.5}$$

The corresponding configurations occur on the boundary of the manipulator workspace or on internal boundaries between regions of the workspace. On these boundaries, the platform can be controlled, but cannot move in all possible directions.

Type II singularities occur when det($\boldsymbol{A}$)=0. From expression **(A.3)** the determinant of $\boldsymbol{A}$ is

$$\begin{aligned}\det(\boldsymbol{A}) &= 2\sin^2\varphi_P - y_P^2\cos\varphi_P - x_P\sin^2\varphi_P + x_P y_P\sin\varphi_P + y_P\sin\varphi_P\cos\varphi_P - 2y_P\sin\varphi_P \\ &= (\sin\varphi_P - y_P)(2\sin\varphi_P - x_P\sin\varphi_P + y_P\cos\varphi_P)\end{aligned} \tag{A.6}$$

Setting expression **(A.6)** equal to 0 results in two possible solutions:

1. $(2\sin\varphi_P - x_P\sin\varphi_P + y_P\cos\varphi_P) = 0 \Rightarrow \tan\varphi_P = \dfrac{y_P}{x_P - 2}$ **(A.7)**

   This corresponds to the configuration shown in Figure A-1(a) where the platform and the third leg are collinear.

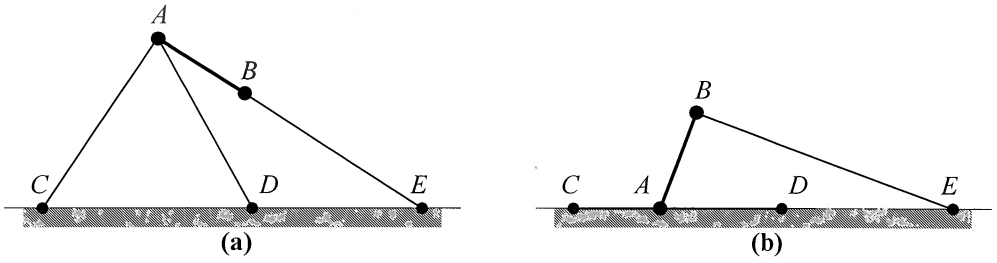

Figure A-1: Singular configurations

2. $(\sin\varphi_P - y_P) = 0 \Rightarrow y_A = 0$ **(A.8)**

   This configuration is shown in Figure A-1(b) where the legs 1 and 2 are collinear.

In these configurations the platform is locally moveable, even if the actuator leg lengths are fixed. In such a position, the platform cannot be controlled.

Type III singularities, also known as architecture singularities, occur when $\boldsymbol{A}$ and $\boldsymbol{B}$ are simultaneously singular. When this occurs, finite output motion is possible even when the actuators are fixed. For the Stewart platform considered one example of this would occur if $x_C = x_D$ and $l_1 = l_2$. A platform in a type III singular configuration cannot be controlled. It is assumed that these types of singularities are avoided by the proper choice of geometric parameters.

Haug et al. **[18]** have determined that the same planar Stewart platform is in a singular position when the platform and any one of the actuator legs are collinear, and the same leg is at a minimum or maximum length. The apparent additional singularities can be accounted for by the fact that Haug et al. introduce an input coordinate transformation to ensure that the leg length constraints are automatically satisfied. Their singularities therefore correspond to type I singularities (Haug et al. **[19]**). Incidentally, in the optimization approach used in this study, the use of leg length constraints during workspace determination ensures that leg length constraints are never violated.

Although the preceding analysis was performed on a specific normalized geometry, it is evident that the results should hold true for any planar Stewart platform that is configured as shown in Figure 3-1 with $y_C = y_D = y_E = 0$.

Appendix B: SNYMAN'S DYNAMIC TRAJECTORY OPTIMIZATION METHOD

B.1 Background

The dynamic trajectory method (also called the "leap-frog" method) for the unconstrained minimization of a scalar function $f(\boldsymbol{x})$ of n real variables represented by the vector $\boldsymbol{x}=(x_1,x_2,\ldots,x_n)^T$ was originally proposed by Snyman [32,33]. The original algorithm has recently been modified to handle constraints by means of a penalty function formulation. (Snyman et al [34,35]). The method possesses the following characteristics:

- Uses only function *gradient* information $\nabla f(\boldsymbol{x})$,
- *No* explicit *line searches* are performed,
- Extremely *robust*: handles steep valleys, and discontinuities in functions and gradients,
- Algorithm seeks *low local minimum* – can be used as a basic component in a methodology for global optimization, and
- Not as efficient as classical methods on smooth and near-quadratic functions.

B.2 Basic dynamic model

The algorithm is modeled on the motion of a particle of unit mass in a n-dimensional conservative force field with potential energy at $\boldsymbol{x}$ given by $f(\boldsymbol{x})$. At $\boldsymbol{x}$, the force on the particle is given by

$$\boldsymbol{a} = \ddot{\boldsymbol{x}} = -\nabla f(\boldsymbol{x}) \tag{B.1}$$

from which it follows that for the time interval [0,t]

$$\begin{aligned} \tfrac{1}{2}\|\dot{\boldsymbol{x}}(t)\|^2 - \tfrac{1}{2}\|\dot{\boldsymbol{x}}(0)\|^2 &= f(\boldsymbol{x}(0)) - f(\boldsymbol{x}(t)) \\ T(t) - T(0) &= f(0) - f(t) \end{aligned} \tag{B.2}$$

or

$$f(t) + T(t) = \text{constant } \{\text{conservation of energy}\}$$

Note that since $\Delta f = -\Delta T$ as long as T increases f decreases. This forms the basis of the dynamic algorithm.

B.3 LFOP: Basic algorithm for unconstrained problems

Given $f(\boldsymbol{x})$ and a starting point $\boldsymbol{x}(0)=\boldsymbol{x}^0$

- Compute the dynamic trajectory by solving the initial value problem (IVP)

$$\ddot{\boldsymbol{x}}(t) = -\nabla f(\boldsymbol{x}(t)) \qquad \text{(B.3)}$$
$$\dot{\boldsymbol{x}}(0) = \boldsymbol{0}, \quad \boldsymbol{x}(0) = \boldsymbol{x}^0$$

- Monitor $\dot{\boldsymbol{x}}(t) = v(t)$. Clearly as long as $T = \frac{1}{2}\|\boldsymbol{v}(t)\|^2$ increases f($\boldsymbol{x}$(t)) decreases – OK!
- When $\|\boldsymbol{v}(t)\|$ decreases apply some interfering strategy to extract energy and thereby increase the likelihood of descent.
- In practice a numerical integration "leap-frog" scheme is used to integrate the IVP **(B.3)** Compute for k=0,1,2,… and time step Δt

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \boldsymbol{v}^k \Delta t \qquad \text{(B.4)}$$
$$\boldsymbol{v}^{k+1} = \boldsymbol{v}^k + \boldsymbol{a}^{k+1} \Delta t$$

where $\boldsymbol{a}^k = -\nabla f(\boldsymbol{x}^k), \quad \boldsymbol{v}^0 = \frac{1}{2}\boldsymbol{a}^0 \Delta t$

- A typical interfering strategy is

  If $\|\boldsymbol{v}^{k+1}\| \geq \|\boldsymbol{v}^k\|$ continue

  else

$$\text{set } \boldsymbol{v}^k = \frac{\boldsymbol{v}^{k+1} + \boldsymbol{v}^k}{4}, \quad \boldsymbol{x}^k = \frac{\boldsymbol{x}^{k+1} + \boldsymbol{x}^k}{2} \qquad \text{(B.5)}$$

  compute new $\boldsymbol{v}^{k+1}$ and continue.
- Further heuristics are used to determine an initial Δt, to allow for magnification and reduction of Δt, and to control the step size.

B.4 LFOPC: Modification for constrained problems

Constrained optimization problems are solved by the application, in three phases, of LFOP to a penalty function formulation of the problem **[34,35]**. Given a function $f(\boldsymbol{x})$, with equality constraints h_i=0 (i=1,2,…,m) and inequality constraints $g_j \leq 0$ (j=1,2,…,m) and penalty parameter $\mu >> 0$, the penalty function problem is to minimize

$$P(\boldsymbol{x}, \mu) = f(\boldsymbol{x}) + \sum_{i=1}^{m} \mu h_i^2(\boldsymbol{x}) + \sum_{j=1}^{m} \beta_j g_j^2(\boldsymbol{x}) \qquad \text{(B.6)}$$

$$\text{where } \beta_j = \begin{cases} 0 \text{ if } g_j(\boldsymbol{x}) \leq 0 \\ \mu \text{ if } g_j(\boldsymbol{x}) > 0 \end{cases}$$

Phase 0: Given some $\boldsymbol{x}^0$, then with the overall penalty parameter $\mu = \mu_0 (= 10^2)$ apply LFOP to $P(\boldsymbol{x},\mu_0)$ to give $\boldsymbol{x}^*(\mu_0)$

Phase 1: With $\boldsymbol{x}^0 = \boldsymbol{x}^*(\mu_0)$, $\mu = \mu_1 (= 10^4)$ apply LFOP to $P(\boldsymbol{x},\mu_1)$ to give $\boldsymbol{x}^*(\mu_1)$ and identify active constraints i_a=1,2,…,n_a; $g_{i_a}(\boldsymbol{x}^*(\mu_1)) > 0$

Phase 2: With $\boldsymbol{x}^0 = \boldsymbol{x}^*(\mu_1)$, use LFOP to minimize

$$P_a(\boldsymbol{x},\mu_1) = \sum_{i=1}^{r} \mu_1 h_i^2(\boldsymbol{x}) + \sum_{i_a=1}^{n_a} \mu_1 g_i^2(\boldsymbol{x}) \tag{B.7}$$

to give $\boldsymbol{x}^*$.

B.5 The use of LFOPC in the optimization approach

The dynamic trajectory algorithm is essential to the successful implementation of the optimization approach. This method is a proven robust method. The trajectory nature of the algorithm ensures controlled and stable convergence to the optimum. In particular, when solving the successive optimization problems inherent in the optimization approach the trajectory method is more reliable in tracking the local "optimization valley", which corresponds to the workspace boundary, than other more established classical optimization techniques. This local convergence property can be controlled in the optimization algorithm by means of a maximum step size parameter (DELT). Furthermore LFOPC provides a convenient method for determining the bifurcation points by means of the minimization of an error function.

REFERENCES

1. J.A. Snyman, L.J. du Plessis and J. Duffy *'An optimization approach to the determination of the boundaries of manipulator workspaces'*, Technical Report, Department of Mechanical and Aeronautical Engineering, University of Pretoria (1998). Submitted for publication.

2. L.J. du Plessis, *'An optimization approach to the determination of manipulator workspaces'*, Master of Engineering Thesis, Department of Mechanical and Aeronautical Engineering, University of Pretoria (1999).

3. J.-P. Merlet, *'Parallel manipulators: state of the art and perspectives'*, Advanced Robotics. **8**(6) 589-596 (1994).

4. E.F. Fichter and E.D. McDowell, *'A novel design for a robot arm'*, Proceedings of the ASME International Computer Technology Conference, San Francisco, 250-256 (1980).

5. D. Stewart, *'A platform with six degrees of freedom'*, Proceedings of the Institution of Mechanical Engineers. **8**(Part1, 6) 589-596 (1965-1966).

6. C. M. Gosselin and J. Angeles, *'The optimum kinematic design of a planar three-degree-of-freedom parallel manipulator'*, Journal of Mechanisms, Transmissions, and Automation in Design. **110** 35-41 (1988).

7. M. Ceccarelli, *'Displacement analysis of a Turin Platform parallel manipulator'*, Advanced Robotics. **11**(1) 17-31 (1997).

8. K.W. Grace, J.E. Colgate, M.R. Glucksberg and J.H. Chun, *'A six degree of freedom micromanipulator for ophthalmic surgery'*, IEEE Conference on Robotics and Automation, Atlanta. 630-635 (1993).

9. T. Arai and K.Cleary, *'Development of a parallel link manipulator for underground excavation task'*, Proceedings of the International Symposium on Advanced Robot Technology. 541-548 (1991).

10. F. Sternheim, *'Tridimensional computer simulation of a parallel robot. Results for the 'DELTA4' machine'*, Proceedings of the 18th International Symposium on Industrial Robotics, Lausanne. 333-340 (1988).

11. University of Nottingham Website; Faculty of Engineering; School of Mechanical, Materials, Manufacturing Engineering and Management; Advanced Manufacturing Technology Research Group; Rapid Response Aerospace Manufacturing Project. (http://www.nottingham.ac.uk/school4m/research/amt/index/htm)

12. NIST Manufacturing Engineering Laboratory Website; National Advanced Manufacturing Testbed; Characterization, Remote Access and Simulation of Hexapod Machines. (http://www.mel.nist.gov/namt/projects/hexapod/hex1.htm)

13. Hexel Corporation Website. (http://www.hexel.com)

14. C.M. Gosselin, *'Determination of the workspace of 6-DOF parallel manipulators'*, Journal of Mechanical Design. **112** 331-336 (1990).

15. J.-P. Merlet, C.M. Gosselin and N. Mouly, *'Workspaces of planar parallel manipulators'*, Mechanisms and Machine Theory. **33**(1) 7-20 (1998).

16. A. Kumar and K.J. Waldron, *'The workspaces of a mechanical manipulator'*, Journal of Mechanical Design **103** 665-672 (1981).

17. L-C.T. Wang and J-H. Hsieh, *'Extreme reaches and reachable workspace analysis of general parallel robotic manipulators'*, Journal of Robotic Systems **15**(3) 145-159 (1998).

18. 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.

19. E.J. Haug, F.A. Adkins, C. Qiu and J. Yen, *'Analysis of barriers to control of manipulators within accessible output sets'*, Technical Report R-174: Center for Computer-Aided Design, University of Iowa (1994).

20. J.-P. Merlet, *'Determination of the orientation workspace of parallel manipulators'*, Journal of Intelligent and Robotic Systems. **13** 143-160 (1995).

21. D.C.H. Yang and T.W. Lee, *'Feasibility study of a platform type of robotic manipulators from a kinematic viewpoint'*, Journal of Mechanisms, Transmissions and Automation in Design. **106** 191-198 (1984).

22. A.G Chrisp and N.N.Z. Gindy, *'Parallel link machine tools: simulation, workspace analysis and component positioning'*, Research Report, RRAM, Dept. of Mechanical Engineering and Operations Management, The University of Nottingham.

23. E.F. Fichter, *'A Stewart platform-based manipulator: general theory and practical construction'*, International Journal of Robotics Research. **5**(2) 157-182 (1986).

24. T. Arai, K. Cleary, T. Nakamura, H. Adachi and K. Homma, *'Design, analysis and construction of a prototype parallel link manipulator'*, IEEE International Workshop on Intelligent Robots and Systems, Ibaraki, Japan **1** 205-212(1990).

25. O. Masory and J. Wang, *'Workspace evaluation of Stewart platforms'*, Robotics, Spatial Systems and Mechanical Systems ASME DE-Vol.**45** 337-346 (1992).

26. D.-Y. Jo and E.J. Haug, *'Workspace analysis of multibody mechanical systems using continuation methods'*, Journal of Mechanisms, Transmissions, and Automation in Design. **111** 581-589 (1989).

27. D.-Y. Jo and E.J. Haug, *'Workspace analysis of closed-loop mechanisms with unilateral constraints'*, Advances in Design Automation ASME DE **19**(3) 53-60 (1988).

28. E.J. Haug, C.M. Luh, F.A. Adkins and J.Y. Wang, *'Numerical algorithms for mapping boundaries of manipulator workspaces'*, IUTAM Fifth Summer School on Mechanics, Aalborg, Denmark: Concurrent Engineering Tools for Dynamic Analysis and Optimization (1994).

29. J.-P. Merlet, *'Designing a parallel manipulator for a specific workspace'*, Research Report 2527, INRIA, Sophia-Antipolis, France. (1995).

30. A. Bajpai and B. Roth, *'Workspace and mobility of a closed-loop manipulator'*, International Journal of Robotics Research **5**(2) 131-142 (1986).

31. V. Kumar, *'Characterization of workspaces of parallel manipulators'*, Journal of Mechanical Design **114** 369-375 (1992).

32. J.A. Snyman, *'A new and dynamic method for unconstrained minimization'*, Appl. Math. Modeling. **6** 449-462 (1982).

33. J.A. Snyman, *'An improved version of the original leap-frog method for unconstrained minimization'*, Appl. Math. Modeling. 7 216-218 (1983).

34. J.A. Snyman, *'The LFOPC leap-frog algorithm for constrained optimization'*, Technical Report, Department of Mechanical Engineering, University of Pretoria (1999) To appear in Computers and Mathematics with Applications.

35. J.A. Snyman, W.J. Roux and N. Stander, *'A dynamic penalty function method for the solution of structural optimization problems'*, Appl. Math. Modeling. **180** (15) 371-386 (1994).

36. R. Vijaykumar, K.J. Waldron and M.J. Tsai, *'Geometric optimization of serial chain structures for working volume and dexterity'*, Journal of Robotics Research. **5**(2) 91-103 (1986).

37. C.M. Gosselin, *'Stiffness mapping for parallel manipulators'*, IEEE Transactions on Robotics and Automation. **6**(3) 377-382 (1990).

38. C.M. Gosselin and J. Angeles, *'Singularity analysis of closed-loop kinematic chains'*, IEEE Transactions on Robotics and Automation. **6**(3) 281-290 (1990).

39. J. Sefrioui and C.M. Gosselin, *'Singularity analysis and representation of planar parallel manipulators'*, Robotics and Autonomous Systems. **10** 209-224 (1992).