Stony Brook University

Computer Aided Synthesis of Rational Motions under Kinematic Constraints

A Dissertation Presented

by

Zhe Jin

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Mechanical Engineering

Stony Brook University

May 2008

Stony Brook University

The Graduate School

Zhe Jin

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Qiaode Jeffrey Ge, Dissertation Advisor
Professor, Department of Mechanical Engineering

Peisen Huang, Chairperson of Defense
Associate Professor, Department of Mechanical Engineering

Anurag Purwar
Research Assistant Professor, Department of Mechanical Engineering

Yu Zhou
Assistant Professor, Department of Mechanical Engineering

Joseph Mitchell, Outside Member
Professor, Department of Applied Mathematics & Statistics

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

Computer Aided Synthesis of Rational Motions under Kinematic Constraints

by

Zhe Jin

Doctor of Philosophy

in

Mechanical Engineering

Stony Brook University

2008

This dissertation deals with the problem of synthesizing rational motions of a rigid body that satisfy kinematic constraints imposed by planar, spherical, and spatial kinematic chains. The dissertation brings together the well-known kinematics of various kinematic chains and the recently developed freeform rational motions to synthesize the constrained rational motions for Cartesian motion planning. The kinematic constraints under consideration are workspace related constraints that limit the position of the end link of open chains and the coupler link of closed chains.

Planar quaternions, quaternions, and dual quaternions are used to represent planar, spherical, and spatial displacements, respectively. In this way,

displacements of a rigid body in Cartesian space are mapped into points in quaternion space, and the kinematic constraints are transformed into geometric constraints, such as circle, circular ring, spherical and hyperboloidal shells in quaternion space. Thus, the problem of rational motion interpolation is transformed into that of rational curve interpolation, where the standard scheme for curve interpolation in Computer Aided Geometric Design (CAGD) can be applied. For the constrained curve interpolation this dissertation develops several efficient numerical algorithms that include smooth piecewise rational Bézier interpolation on a circle, smooth rational B-spline interpolation inside an n-spherical shell and within intersection of two hyperboloidal shells.

The last portion of the dissertation adopts a different approach for rational motion interpolation of planar chains in a parametric space defined by the elements of planar displacement matrix. This approach has the advantage of being direct and yields lower degree motions.

The results of this dissertation have applications in Cartesian motion planning in robotics and task specification for task driven design of robots and mechanisms.

To My Parents and My Wife

Table of Contents

List of Figures

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Professor Qiaode Jeffrey Ge, my research advisor, for his guidance, encouragement and patience throughout the work on this dissertation.

I would like to thank my committee members, Professor Peisen Huang, Professor Yu Zhou and Professor Joseph Mitchell for their review of this manuscript, and Professor Anurag Purwar for his valuable assistance throughout my research.

My thanks also go to my lab mates Carlos A. Trujillo and Jun Wu for their suggestions and friendship.

I would also like to thank my wife Minghua Cui for her understanding, endless supporting and love, without which I could not have concentrated on my research.

The text of this dissertation in part is a reprint of the materials as it appears in Jin and Ge [1, 2, 3] and Purwar and Jin [4, 5]. The co-author listed in the publications directed and supervised the research that forms the basis for this dissertation.

Finally, I gratefully acknowledge the National Science Foundation for the financial support of this work under grant number DMI-0500064.

Chapter 1

Introduction

This dissertation deals with the problem of synthesizing rational motions of a rigid body under kinematic constraints that are imposed by open and closed kinematic chains. In this introductory chapter, a general overview on background and the existing work in the area of motion planning is presented followed by the main contributions of this dissertation.

Kinematics (Reuleaux [6], Hunt [7], Bottema and Roth[8], Angeles[9], McCarthy [10]) deals with the phenomenon of motion without regard to the cause of the motion. The area of Computer Aided Geometric Design (CAGD) (Farin [11, 12], Farin et al. [13], Hoschek and Lasser [14], Piegl and Tiller [15], Gallier [16]) is concerned with the approximation and representation of curves and surfaces that arise when these objects have to be processed by computer. Since the seminal work of Shoemake [17], the past two decades have witnessed significant progress in merging of the field of Computer Aided Geometric Design and Kinematics for the development of rational Bézier and B-spline motions of rigid bodies. The idea behind such a synergy is that the problem of

designing rational motions of rigid bodies can be transformed into the problem of designing rational curves in a higher dimensional projective space via special mappings. By choosing quaternion representations for the planar, spherical, and spatial displacements, the problem is further reduced to designing curves in the space of planar quaternions, quaternions, and dual quaternions, respectively (Bottema and Roth [8] and McCarthy [10]). Rational motions, with applications spanning across areas such as motion animation in computer graphics, task specification in mechanism synthesis, and virtual reality systems as well as Cartesian motion planning in robotics, are an attractive proposition since they integrate well with the industry standard Non-uniform Rational B-spline (NURBS) based CAD/CAM system. Furthermore, from a computational perspective they can easily exploit fast and stable algorithms from CAGD.

The earliest instance of such a synergistic endeavor is found in Shoemake's [17] work, who extended the idea of linear interpolation in affine spaces to the spherical linear interpolation (*slerp*ing) of quaternions. Since then, there has been quite some work on the synthesis of spherical motions by designing quaternion based splines such as Pletinckx [18], Dam et al. [19], Duff [20], Kim et al. [21], Kim and Nam [22], Nielson [23, 24], Wang and Joe [25], and Barr et al [26]. Ge and Ravani [27, 28] extended this technique of motion synthesis to the domain of spatial motion interpolation by using a dual quaternion approach. Their work was further refined by Jüttler and Wagner [29], Wagner [30], and Purwar and Ge [31]. A comprehensive list of references on the

topic of free-form rational motion design can be found in a survey paper by Röschel [32], and the most updated ones can be found in Purwar [33].

Despite all the advances in the area of rational motion design, none of the aforementioned work deals with design of rational motions under kinematic constraints. The seemingly related work by Horsch and Jüttler [34] and Wagner and Ravani [35] seeks direct application of rational motions to Cartesian motion planning of robots and has not dealt with rational motions under kinematic constraints. Ge and Larochelle [36] presented a preliminary framework for approximating algebraic motions of four-bar spherical mechanisms with rational B-Spline spherical motions.

This dissertation deals with rational motion interpolation under kinematic constraints of open and closed kinematic chains. With the aid of quaternion representations of planar, spherical, and spatial displacements, the workspace constraints of the end link of open kinematic chains or the coupler link of closed kinematic chains can be transformed into geometric constraints (algebraic equations such as circles and circular rings). Thus, the problem of synthesizing the Cartesian rational motion under the kinematic constraints is converted into one that deals with geometric constraints in the space of quaternions (planar quaternions, quaternions, or spatial quaternions).

As mentioned above, the problem of rational motion interpolation under kinematic constraints of kinematic chains is reduced to that of rational curve interpolation under geometric constraints in quaternion space. Hofer and Pottmann have done some work on spline curve interpolation on surfaces

of arbitrary dimensions by variational approach (Hofer et al. [37, 38, 39] and Pottmann et al. [40]). They proposed an algorithm that minimizes an energy of curves on surfaces of arbitrary dimension and codimension. They also proposed a subdivision based algorithm for interpolation such that the chosen feature points of the moving object follow smooth trajectory. Pobegailo [41] gave a method to design motion by using B-spline interpolant and exploiting the affine connection, when the parameterized curve is given. If the geometric constraint transformed from its kinematic constraint is a surface in quaternion space, we can apply the methods developed by Hofer and Pottmann. But kinematic constraints of kinematic chains can be curves, constrained regions on a plane, or intersection of two geometric objects, etc. For example, in planar quaternion space, the kinematic constraint of planar 3R open chain is a circular ring and the kinematic constraint of planar 6R closed chain is the intersection of two hyperboloidal shells. Thus, we proposed different approaches in this dissertation for rational motion interpolation under kinematic constraints of kinematic chains (see Jin and Ge [1, 2, 3] and Purwar and Jin[4, 5]).

This dissertation deals with the following constrained motion interpolation problem:

Given: A set of the positions of the end link of a kinematic open chain or the coupler link of a kinematic closed chain in its workspace as well as the parameter values $u_i,\ i = 0, ..., L$ and the configuration of corresponding kinematic chain.

Find: A smooth rational motion of the end link of the kinematic open chain

or the coupler link of the kinematic closed chain that interpolates through the given positions at the parameter values and satisfies the kinematic constraints of corresponding kinematic chain.

Through the use of planar quaternions, it is shown that the problem of synthesizing the Cartesian rational motion of a planar 2R open chain can be reduced to that of circular interpolations in two separate planes. Piecewise rational Bézier interpolation on a circle can be applied (see Forrest (1968), Piegl and Tiller (1987, 1989) for details). Furthermore, the problem of synthesizing the Cartesian rational motion of a planar 3R arm can be reduced to that of circular interpolation in one plane and constrained spline interpolation in a circular ring. Meek and Ong [42] proposed an algorithm which constructs a G^2 interpolating curve on one side of a polyline or on the constrained polyline. This dissertation develops an algorithm which can produce a smooth (C^2 or higher) rational B-spline curve within an *n-spherical Shell* (when $n = 3$, the constraint is a three dimensional concentric spherical shell; when $n = 2$, the constraint is a two dimensional concentric circular ring and if $n = 1$, the constraint is a band). Due to the limitation of circular interpolation, only C^1 continuous rational motions are generated. For applications that require C^2 or higher continuous motions, the dissertation presents a method for generating smooth (C^2 or higher) motions that approximate the kinematic constraints for planar 2R and 3R open chains. For the planar 6R close chain, the kinematic constraint is transformed into the intersection of two hyperboloidal shells in planar quaternion space. In the past some work has been done on design-

ing splines on quadrics and hypersurfaces (see Gfrerrer [43] and Wang and Joe [44]) as well as on surfaces of arbitrary dimension and codimension (Hofer and Pottmann [39]). Given a set of positions within the workspace of a planar 6R closed chain, a freeform rational B-spline motion is used to interpolate the given positions. To ensure that the entire motion satisfies the kinematic constraints, an algorithm is developed that detects an extreme position on the rational motion that violates the kinematic constraints. This position is then modified so that it is in compliance with the kinematic constraints and is added to the list of positions to be interpolated. Since we require the modified new positions to be minimally away from the extreme positions, the issue of finding new positions can be seen as a normal distance minimization problem in the image space subject to certain constraints. Ravani and Roth [45] proposed a general algebraic method for approximate normal distance calculation between the image curve and a given position in the image space. Later on, Bodduluri and McCarthy [46] used their method for finite position synthesis of a spherical four-bar motion. In this dissertation we modify their methods to find new positions which are projected onto kinematic constraint manifolds from extreme positions. By repeating this process, one obtain a rational B-spline motion such that it fully satisfies the kinematic constraints of the planar 6R closed chain. As planar 5R and 4R closed chains can be obtained by holding one or two joints fixed, the above mentioned algorithm for planar 6R closed chain is also shown to be applicable to the problem of synthesizing rational motions for planar 5R and 4R chains.

Similarly, through the use of quaternions, it is shown that the problem of synthesizing the Cartesian rational motion of a spherical 2R open chain can be reduced to that of circular interpolations in two separate planes. The problem of synthesizing the Cartesian rational motion of a spherical 3R open chain can be reduced to that of constrained spline interpolation in two different planes. We show that in case of the spherical 3R open chain, it is possible to get an exact C^2 or higher continuous rational motion, while the same method can be used to synthesize a C^2 or higher continuous rational motion for a spherical 2R open chain that approximates the kinematic constraint within the limits of a user defined value. Furthermore, the kinematic constraint of a spherical 6R closed chain is transformed into geometric constraint, which is the intersection of two hyperboloidal shells in quaternion space. We can apply a numerical algorithm similar to the one developed for planar 6R closed chain. As spherical 5R and 4R closed chains can be obtained by holding one or two joints fixed, the above mentioned algorithm for spherical 6R closed chain is also shown to be applicable to the problem of synthesizing rational motions for spherical 5R and 4R closed chains.

The spatial displacement is represented by a dual quaternion (see Bottema and Roth [8] and McCarthy [10] for quaternion representations of displacements). In this way, the kinematic constraints of the ADEPT robot are transformed into geometric constraints of circle, circular ring, and band. Thus, the same algorithms for planar 2R and 3R open chains can be applied. For the case of spatial SS open chain, we develop an algorithm which is similar to the

one for planar or spherical 6R closed chain.

In Kinematics, it is well known that by using quaternion representation of displacement the elements of displacement matrix are quadratic in terms of the coordinates of quaternions (see Bottema and Roth [8] and McCarthy [10]). Therefore, if the interpolating curve in quaternion space is a polynomial function of degree n, then the corresponding displacement matrix represents a rational motion of degree $2n$, which means the degree of the Cartesian motion is doubled. To overcome this disadvantage, we formulate the kinematic constraints of planar open and closed chains in terms of the elements of planar displacement matrix, thus give rise to the constraint manifold in the parameter space of displacement matrix elements. In the space of planar displacement matrix elements, this manifold is given by algebraic equations and can be seen as describing geometric constraints. Iterative algorithms are developed which are similar to those proposed for planar open and closed kinematic chains. Advantages of directly using the elements of displacement matrix (as opposed to quaternions) for motion interpolation are that the interpolation process is straightforward and the resulting motion is of lower degree; e.g., cubic interpolation of planar quaternions produces a motion of degree six, while that of the elements of displacement matrices produces a motion of degree three only. We note that the degree mentioned here pertains to the motion interpolation parameter, usually associated with time.

The rest of the dissertation is organized as follows. Chapter 2 reviews representations of planar, spherical, and spatial displacements in terms of planar

quaternions, quaternions, and dual quaternions, respectively. Chapter 3 deals with the formulation of kinematic constraints of planar, spherical, and spatial mechanisms. Chapter 4 studies the problem of synthesizing rational motions of a rigid body under kinematic constraints that are imposed by planar and spherical 2R and 3R open chains. Chapter 5 deals with the synthesis of rational motions under kinematic constraints of planar and spherical 4R, 5R, and 6R closed chains. Chapter 6 studies the problem of synthesizing rational motions under kinematic constraints of spatial mechanisms, such as spatial SS open chain and ADEPT robot. Chapter 7 deals with the problem of synthesizing rational motions under kinematic constraints of planar open and closed kinematic chains by investigating the matrix representation of planar displacements directly. The final chapter summarizes the results of this research.

Chapter 2

Displacements and Quaternions

2.1 Introduction

This chapter deals, at the fundamental level, with the geometry of displacements as well as the quaternion representations of planar, spherical, and spatial displacements. Details about displacements and quaternions can be found in Bottema and Roth [8] and McCarthy [10], and Appendix A summarizes quaternion algebra.

The organization of this chapter is as follows. Section 2.2 reviews a formulation of spherical displacements using quaternions. Section 2.3 presents a formulation of planar displacements using planar quaternions. Section 2.4 shows a spatial displacement and its dual quaternion representation.

2.2 Spherical Displacements and Quaternions

The rotation of a three dimensional body, $\mathbf{M}$, with respect to a fixed body, $\mathbf{F}$, can be viewed as a displacement of the frame $\mathbf{M}$ from an initial position coin-

ciding with $\mathbf{F}$ to its final position. Let $\mathbf{x}$ and $\mathbf{X}$ be three dimensional vectors defining coordinates of a point P in $\mathbf{M}$ and $\mathbf{F}$, then under this displacement the point P is constrained to lie on a sphere and this displacement is termed a *spherical displacement*.

Any rotation in three-dimensional space has a rotation axis and a rotation angle about this axis. Let $\mathbf{s} = (s_x, s_y, s_z)$ denote a unit vector along the axis and θ denote the angle of rotation. They can be used to define the so-called *Euler-Rodrigues parameters:*

$$q_1 = s_x \sin(\theta/2),\ q_2 = s_y \sin(\theta/2),\ q_3 = s_z \sin(\theta/2),\ q_4 = \cos(\theta/2). \tag{2.1}$$

The Euler-Rodrigues parameters and the quaternion units, $1, \mathbf{i}, \mathbf{j}, \mathbf{k}$ can be combined to define a quaternion of rotation:

$$\mathbf{q} = q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} + q_4. \tag{2.2}$$

A quaternion $\mathbf{q}$, at times, is also written as an ordered quadruple (q_1, q_2, q_3, q_4). Since $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$, $\mathbf{q}$ is also called a unit quaternion. Details on quaternions are found in Bottema and Roth [8] and McCarthy [10], and Appendix A summarizes quaternion algebra.

If we consider $\mathbf{x}$ and $\mathbf{X}$ as the *vector quaternions* (no coefficient of 1), then the rotation is given by the quaternion equation

$$\mathbf{X} = \mathbf{q}\mathbf{x}\mathbf{q}^* \tag{2.3}$$

where $\mathbf{q}^* = q_4 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k}$ is the conjugate of $\mathbf{q}$.

We can apply homogeneous transform matrix form to represent the Eq. (2.3):

$$\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = [A] \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}, \tag{2.4}$$

where

$$[A] = \frac{1}{S^2} \begin{bmatrix} q_4^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_4q_3) & 2(q_1q_3 + q_4q_2) & 0 \\ 2(q_2q_1 + q_4q_3) & q_4^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_4q_1) & 0 \\ 2(q_3q_1 - q_4q_2) & 2(q_3q_2 + q_4q_1) & q_4^2 - q_1^2 - q_2^2 + q_3^2 & 0 \\ 0 & 0 & 0 & S^2 \end{bmatrix}, \tag{2.5}$$

where $S^2 = q_1^2 + q_2^2 + q_3^2 + q_4^2$. Details about quaternion algebra can be found in Appendix A.

Note that when q_i is replaced by $Q_i = wq_i$ $(i = 1, 2, 3, 4)$, where w is a nonzero scalar, the matrix $[A]$ is unchanged. Thus, the quaternion components of $\mathbf{q}$ can be considered as homogeneous coordinates of a rotation.

Quaternion algebra is also used for composing two successive rotations. Let $\mathbf{Q}_0, \mathbf{Q}_1$ denote two rotations. The composition of two rotations $\mathbf{Q}_1$ followed by $\mathbf{Q}_0$ is given by the quaternion product $\mathbf{Q}_0\mathbf{Q}_1$.

2.3 Planar Displacements and Planar Quaternions

It is shown that a spherical displacement can be represented by a quaternion in section 2.2. Similarly, a planar displacement can be represented by a planar quaternion (see Bottema and Roth [8] and McCarthy [10]). Planar quaternions have been used for designing planar open and closed chains (Ravani and Roth [45], Larochelle [47], Murray et al. [48], Perez and McCarthy [49]).

For a planar displacement shown in Figure 2.1, let d_1, d_2 denote the coordinates of the origin of the moving frame $\mathbf{M}$ in the fixed frame $\mathbf{F}$ and α denote the rotation angle of $\mathbf{M}$ relative to $\mathbf{F}$. Then a planar displacement can be represented by a planar quaternion, $\mathbf{Z} = Z_1\epsilon\mathbf{i} + Z_2\epsilon\mathbf{j} + Z_3\mathbf{k} + Z_4$, where $(\mathbf{i}, \mathbf{j}, \mathbf{k}, 1)$ form the quaternion basis and ϵ is the dual unit with the property $\epsilon^2 = 0$. The components of the planar quaternion, $\mathbf{Z} = (Z_1, Z_2, Z_3, Z_4)$, are given by

$$\begin{aligned} Z_1 &= (d_1/2)\cos(\alpha/2) + (d_2/2)\sin(\alpha/2), \\ Z_2 &= -(d_1/2)\sin(\alpha/2) + (d_2/2)\cos(\alpha/2), \\ Z_3 &= \sin(\alpha/2), \\ Z_4 &= \cos(\alpha/2). \end{aligned} \tag{2.6}$$

These four components can be identified as coordinates of a point in four dimensional space. The point $\mathbf{Z}$ is called the *image point of a planar displacement*. The set of image points that represent all planar displacements is called the *image space* of planar displacements and is denoted as Σ_p. In view of (2.6), the coordinates of an image point must satisfy the equation:

$$Z_3^2 + Z_4^2 = 1. \tag{2.7}$$

The above equation may be interpreted as defining a hyper-circular cylinder in four dimensions.

If the point $\mathbf{x} = (x, y)$ in $\mathbb{R}^2$ is identified with $\mathbf{x} = y\mathbf{i}\epsilon - x\mathbf{i}\epsilon + \mathbf{k}$, then the result of planar displacement of $\mathbf{x}$ is obtained by

$$\mathbf{X} = \mathbf{Z}\mathbf{x}\mathbf{Z}^*, \tag{2.8}$$

where $\mathbf{Z} = Z_4 - Z_1\epsilon\mathbf{i} - Z_2\epsilon\mathbf{j} - Z_3\mathbf{k}$ is the *conjugate* of $\mathbf{Z}$.

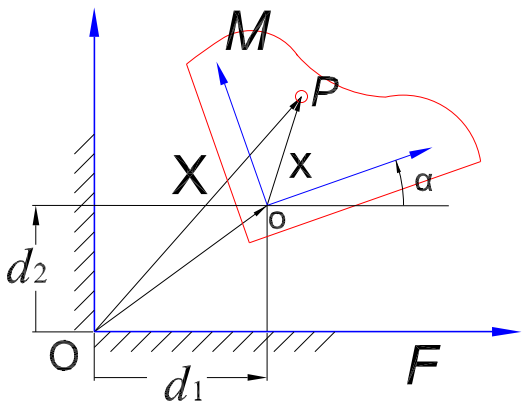

Figure 2.1: A planar displacement.

We can use homogeneous transform matrix to represent Eq. (2.8):

$$\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = [A] \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}, \tag{2.9}$$

where

$$[A] = \frac{1}{Z_3^2 + Z_4^2} \begin{bmatrix} Z_4^2 - Z_3^2 & -2Z_3Z_4 & 2(Z_1Z_4 - Z_2Z_3) \\ 2Z_3Z_4 & Z_4^2 - Z_3^2 & 2(Z_1Z_3 + Z_2Z_4) \\ 0 & 0 & Z_3^2 + Z_4^2 \end{bmatrix}. \tag{2.10}$$

Note that when Z_i $(i = 1, 2, 3, 4)$ is replaced by wZ_i, where w is a nonzero scalar, the matrix $[A]$ is unchanged. From this perspective, the four components of a planar quaternion can also be considered as a set of homogeneous coordinates for a planar displacement.

Quaternion algebra is also used for composing two successive planar displacements. Let $\mathbf{Z}_0, \mathbf{Z}_1$ denote two planar displacements. The composition of two planar displacements $\mathbf{Z}_1$ followed by $\mathbf{Z}_0$ is given by the quaternion product $\mathbf{Z}_0\mathbf{Z}_1$.

2.4 Spatial Displacements and Dual Quaternions

2.4.1 Spatial Displaements

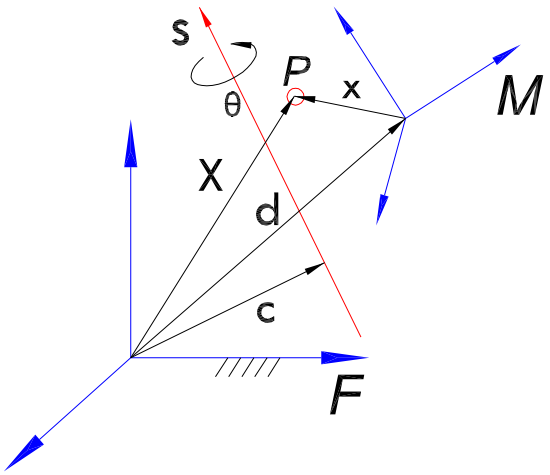

Figure 2.2: A spatial displacement.

A spatial displacement (see Figure 2.2) is most commonly represented as a rigid transformation from **M** to **F** in terms of point coordinates:

$$\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} & [R] & & \mathbf{d} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}, \tag{2.11}$$

where $\mathbf{X}$ and $\mathbf{x}$ are vectors whose scalar components are the Cartesian coordinates of the point as measured in **F** and **M**, respectively. The rotation matrix $[R]$ can be parameterized with quaternions $\mathbf{q} = (q_1, q_2, q_3, q_4)$ where $q_i, i = 1, 2, 3, 4$ are given by Eq. (2.1):

$$[R] = \frac{1}{S^2} \begin{bmatrix} q_4^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_4q_3) & 2(q_1q_3+q_4q_2) \\ 2(q_2q_1+q_4q_3) & q_4^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_4q_1) \\ 2(q_3q_1-q_4q_2) & 2(q_3q_2+q_4q_1) & q_4^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}, \tag{2.12}$$

where $S^2 = q_1^2 + q_2^2 + q_3^2 + q_4^2$.

The use of such matrix representation, however, is not convenient when dealing with the problem of synthesizing a rational motion that interpolates or approximates a set of displacements. One of the main obstacles is the issue of preserving the orthogonality of the rotation matrix in the interpolation/approximation process (Fillmore [50], Röschel [32]). It has been recognized that an effective way of dealing with the problem is to use dual quaternions (Ge and Ravani [27]). In what follows, we review the concepts dual quaternions in so far as necessary for the development of the current chapter.

2.4.2 Dual Quaternions

Dual quaternions $\hat{\mathbf{Q}} = \mathbf{Q} + \epsilon\mathbf{Q^0}$ can represent spatial displacements. The real part $\mathbf{Q} = Q_1\mathbf{i}+Q_2\mathbf{j}+Q_3\mathbf{k}+Q_4$ is defined by the homogeneous Euler parameters of rotation, see Eq. (2.1). The dual part, $\mathbf{Q^0}$, is given by the formula:

$$\begin{bmatrix} Q_1^0 \\ Q_2^0 \\ Q_3^0 \\ Q_4^0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -d_3 & d_2 & d_1 \\ d_3 & 0 & -d_1 & d_2 \\ -d_2 & d_1 & 0 & d_3 \\ -d_1 & -d_2 & -d_3 & 0 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} \tag{2.13}$$

The translation vector $\mathbf{d} = (d_1, d_2, d_3)$ can be recovered from Eq. (2.13) in terms of $(\mathbf{Q}, \mathbf{Q}^0)$ by using the following

$$\mathbf{d} = -\frac{2}{S^2} \begin{bmatrix} Q_4^0 Q_1 - Q_1^0 Q_4 + Q_2^0 Q_3 - Q_3^0 Q_2 \\ Q_4^0 Q_2 - Q_2^0 Q_4 + Q_3^0 Q_1 - Q_1^0 Q_3 \\ Q_4^0 Q_3 - Q_3^0 Q_4 + Q_1^0 Q_2 - Q_2^0 Q_1 \end{bmatrix} \tag{2.14}$$

We now recast Eq. (2.11) in terms of dual quaternions and the homogeneous

coordinates of a point $\mathbf{P} : (P_1, P_2, P_3, P_4)$ of the object (see Sirchia [51]):

$$\tilde{\mathbf{P}} = \mathbf{Q}\mathbf{P}\mathbf{Q}^* + P_4[(\mathbf{Q}^0)\mathbf{Q}^* - \mathbf{Q}(\mathbf{Q}^0)^*] \tag{2.15}$$

where $\mathbf{Q}^*$ and $(\mathbf{Q}^0)^*$ are conjugates of $\mathbf{Q}$ and $\mathbf{Q}^0$, respectively and $\tilde{\mathbf{P}}$ denotes homogeneous coordinates of the point after the displacement.

Similar to the spherical case, quaternion algebra is also used for composing two successive spatial displacements. Let $\hat{\mathbf{Q}}_0, \hat{\mathbf{Q}}_1$ denote two spatial displacements. The composition of the two spatial displacements is given by the quaternion product $\hat{\mathbf{Q}}_0\hat{\mathbf{Q}}_1$, in the order $\hat{\mathbf{Q}}_0$, after $\hat{\mathbf{Q}}_1$.

2.4.3 The Image Space

The unit dual quaternion $\hat{\mathbf{Q}} = \mathbf{Q} + \epsilon\mathbf{Q}^0$ that represent spatial displacements must satisfy the two constraints:

$$\mathbf{Q} \cdot \mathbf{Q} = Q_1^2 + Q_2^2 + Q_3^2 + Q_4^2 = 1, \tag{2.16}$$

and

$$\mathbf{Q} \cdot \mathbf{Q}^0 = Q_1Q_1^0 + Q_2Q_2^0 + Q_3Q_3^0 + Q_4Q_4^0 = 0. \tag{2.17}$$

The first ensures that $\mathbf{Q}$ is a unit quaternion and the second comes from the definition of the dual part $\mathbf{Q}^0$. The set of points $(\mathbf{Q}, \mathbf{Q}^0)$ in $\mathbb{R}^8$ is a six dimensional algebraic manifold of $\mathbb{R}^8$, and is termed the *image space of spatial displacements* (denoted as Σ).

Another way to view this manifold is considering the dual constraint equa-

tion

$$\begin{aligned} \hat{\mathbf{Q}} \cdot \hat{\mathbf{Q}} &= \hat{Q}_1^2 + \hat{Q}_2^2 + \hat{Q}_3^2 + \hat{Q}_4^2 \\ &= Q_1^2 + Q_2^2 + Q_3^2 + Q_4^2 + \epsilon(Q_1 Q_1^0 + Q_2 Q_2^0 + Q_3 Q_3^0 + Q_4 Q_4^0) = 1. \end{aligned} \tag{2.18}$$

This is the equation of a unit hypersphere in a space of four dual dimensions. Thus the dual four dimensional points that represent spatial displacements lie on a dual hypersphere.

Chapter 3

Kinematic Constraints of Planar, Spherical, and Spatial Mechanisms

3.1 Introduction

This chapter deals with in so far as necessary the formulation of kinematic constraints of planar, spherical, and spatial mechanisms using quaternion based representation found in McCarthy [10] and Ge [52].

The organization of this chapter is as follows. Section 3.2 presents the constraint manifolds of planar 2R, 3R open chains and planar 4R, 5R, and 6R closed chains using planar quaternions. Section 3.3 reviews kinematic constraints of spherical 2R, 3R open chains and spherical 4R, 5R, and 6R closed chains using quaternions. Section 3.4 shows the kinematic constraints of spatial SS open chain and ADEPT robot using dual quaternions.

3.2 Kinematic Constraints of Planar Mechanisms

This section reviews the formulation of kinematic constraints of planar 2R, 3R open chains and planar 4R, 5R, and 6R closed chains.

3.2.1 Planar 2R Open Chain

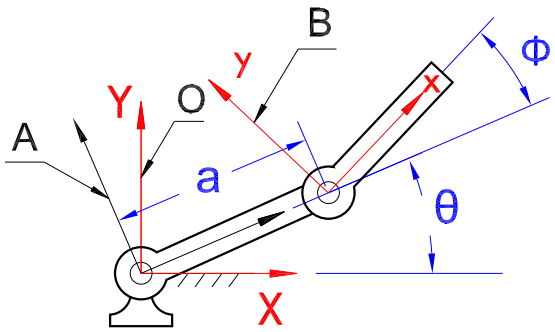

Figure 3.1: A planar 2R open chain.

A kinematic chain connected by two revolute joints with parallel axes is called a planar 2R open chain, see Figure 3.1. We attach a fixed frame **O** to the base and moving frames **A, B** to each of the links. The joint angles θ, ϕ are for the first and the second joints respectively and the link length is denoted as a.

A position of the second link is the composition of a rotation of frame **A** relative to the frame **O** followed by another displacement of **B** relative to **A**. The rotation of **A** is represented by the planar quaternion $\mathbf{Z}(\theta) = \sin(\theta/2)\mathbf{k} + \cos(\theta/2)$ which defines an image point $\mathbf{Z}(\theta) = (0, 0, \sin(\theta/2), \cos(\theta/2)$ in the

image space of planar displacements Σ_p. The displacement of **B** relative to **A** is a translation along **x**-axis by a distance a followed by a rotation about the **z**-axis by an angle ϕ. The translation is given by the planar quaternion $\mathbf{X}(a) = 1 + (a/2)\epsilon\mathbf{i}$ which defines an image point $\mathbf{X}(a) = (a/2, 0, 0, 1)$. The rotation is represented by the image point $\mathbf{Z}(\phi) = (0, 0, \sin(\phi/2), \cos(\phi/2))$. Their product is the image point defining a general position for the second link of the planar 2R open chain:

$$\mathbf{Z}(\theta, \phi) = \mathbf{Z}(\theta)\mathbf{X}(a)\mathbf{Z}(\phi). \tag{3.1}$$

Expanding Eq. (3.1), we obtain $\mathbf{Z}(\theta, \phi) = (Z_1, Z_2, Z_3, Z_4)$ where

$$\begin{array}{l} Z_1 = (a/2)\cos((\theta - \phi)/2), \\ Z_2 = (a/2)\sin((\theta - \phi)/2), \\ Z_3 = \sin((\theta + \phi)/2), \\ Z_4 = \cos((\theta + \phi)/2). \end{array} \tag{3.2}$$

It is clear from the above equations that the four components of the planar quaternion must satisfy the algebraic equations:

$$Z_1^2 + Z_2^2 = a^2/4. \tag{3.3}$$

$$Z_3^2 + Z_4^2 = 1. \tag{3.4}$$

Eqs.(3.3) and (3.4) characterize the kinematic constraints of a planar 2R open chain and are said to define the constraint manifold for the 2R open chain (McCarthy [10]).

3.2.2 Planar PR Open Chain

A displacement of the second link is the composition of a translation of frame **A** relative to the frame **O** followed by another rotation of **B** relative to **A**.

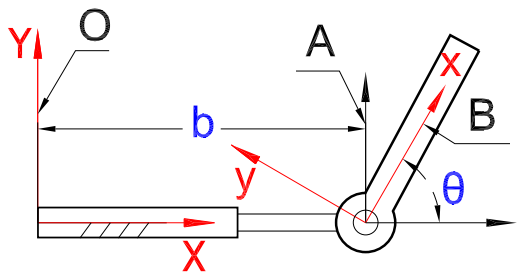

Figure 3.2: A planar PR open chain.

The translation of **A** is given by the planar quaternion $\mathbf{X}(b) = 1 + (b/2)\epsilon\mathbf{i}$. The rotation of **B** is represented by the planar quaternion $\mathbf{Z}(\theta) = \sin(\theta/2)\mathbf{k} + \cos(\theta/2)$. We obtain

$$\mathbf{Z}(\theta, b) = (\frac{b}{2}\cos\frac{\theta}{2}, -\frac{b}{2}\sin\frac{\theta}{2}, \sin\frac{\theta}{2}, \cos\frac{\theta}{2}). \tag{3.5}$$

Let b_1, b_2 denote the lower and upper limits for the range of travel of the prismatic joint, then in addition to the common circle constraint (2.7) we have the kinematic constraints for planar PR open chain:

$$\begin{aligned} Z_1 &= \frac{b}{2}Z_4, \\ Z_2 &= -\frac{b}{2}Z_3, \\ b_1 &\leq b \leq b_2. \end{aligned} \tag{3.6}$$

3.2.3 Planar RP Open Chain

Similar to a planar PR open chain, the planar quaternion for the end link of RP open chain is given by

$$\mathbf{Z}(\theta, b) = (\frac{b}{2}\cos\frac{\theta}{2}, \frac{b}{2}\sin\frac{\theta}{2}, \sin\frac{\theta}{2}, \cos\frac{\theta}{2}). \tag{3.7}$$

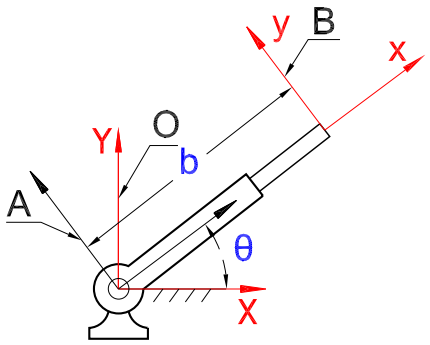

Figure 3.3: A planar RP open chain.

In addition to the common circle constraint (2.7), we have the kinematic constraints:

$$\begin{aligned} Z_1 &= \frac{b}{2} Z_4, \\ Z_2 &= \frac{b}{2} Z_3, \\ b_1 &\leq b \leq b_2. \end{aligned} \tag{3.8}$$

3.2.4 Planar 3R Open Chain

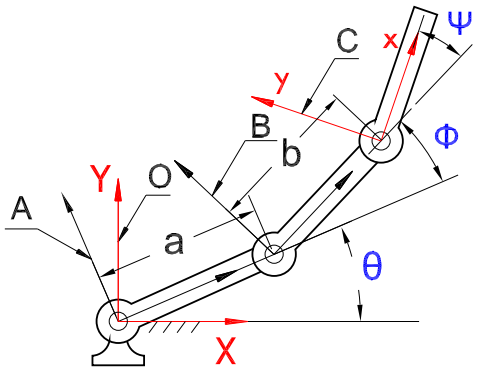

Figure 3.4: A planar 3R open chain.

Consider a planar 3R open chain with reference frames **O, A, B, C**, Fig-

ure 3.4. The length of the first link is a, the length of the second link is b and θ, ϕ, ψ are joint angles for three revolute joints respectively. The constraint manifold of a planar 3R open chain is obtained from that of the 2R open chain by another displacement of $\mathbf{C}$ relative to $\mathbf{B}$. The parameterized equation of the constraint manifold, $\mathbf{Z}(\theta, \phi, \psi)$, of a 3R robot open chain is obtained by multiplying Eq. (3.1) on the right by $\mathbf{X}(b) = (b/2, 0, 0, 1)$, which defines a translation of distance b in the direction of $\mathbf{x}$-axis, and $\mathbf{Z}(\psi) = (0, 0, \sin(\psi/2), \cos(\psi/2))$ which defines a rotation of angle ψ about $\mathbf{z}$-axis:

$$\mathbf{Z}(\theta, \phi, \psi) = \mathbf{Z}(\theta)\mathbf{X}(a)\mathbf{Z}(\phi)\mathbf{X}(b)\mathbf{Z}(\psi). \tag{3.9}$$

The coordinates of $\mathbf{Z}(\theta, \phi, \psi) = (Z_1, Z_2, Z_3, Z_4)$ can be obtained as:

$$\begin{aligned} Z_1 &= \frac{a}{2}\cos\frac{\theta - \phi - \psi}{2} + \frac{b}{2}\cos\frac{\theta + \phi - \psi}{2}, \\ Z_2 &= \frac{a}{2}\sin\frac{\theta - \phi - \psi}{2} + \frac{b}{2}\sin\frac{\theta + \phi - \psi}{2}, \\ Z_3 &= \sin\frac{\theta + \phi + \psi}{2}, \\ Z_4 &= \cos\frac{\theta + \phi + \psi}{2}. \end{aligned} \tag{3.10}$$

From Eq. (3.10), it is not difficult to show that the coordinates, Z_i, satisfy the following equations:

$$Z_1^2 + Z_2^2 = a^2/4 + b^2/4 + (ab/2)\cos(\phi). \tag{3.11}$$

$$Z_3^2 + Z_4^2 = 1. \tag{3.12}$$

Since the range of $\cos(\phi)$ is [-1 1], Eq. (3.11) can be reduced to:

$$(a - b)^2/4 \leq Z_1^2 + Z_2^2 \leq (a + b)^2/4. \tag{3.13}$$

Eqs. (3.12) and (3.13) characterize the kinematic constraints of the planar 3R open chain.

3.2.5 Planar RRP Open Chain

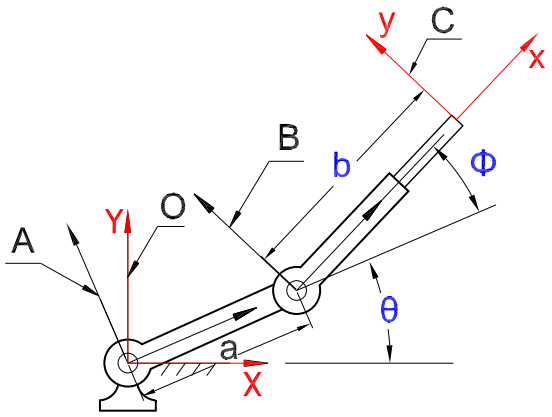

Figure 3.5: A planar RRP open chain.

The planar quaternion associated with the end link is given by

$$\mathbf{Z}(\theta, \phi, b) = \mathbf{Z}(\theta)\mathbf{Z}(\phi)\mathbf{X}(b), \tag{3.14}$$

where $\mathbf{Z}(\theta, \phi, b) = (Z_1, Z_2, Z_3, Z_4)$ are

$$\begin{aligned} Z_1 &= \frac{b}{2}\cos\frac{\theta+\phi}{2} + \frac{a}{2}\cos\frac{\theta-\phi}{2}, \\ Z_2 &= \frac{b}{2}\sin\frac{\theta+\phi}{2} + \frac{a}{2}\sin\frac{\theta-\phi}{2}, \\ Z_3 &= \sin\frac{\theta+\phi}{2}, \\ Z_4 &= \cos\frac{\theta+\phi}{2}. \end{aligned} \tag{3.15}$$

It is clear from the above equation that the components of the planar

quaternion must satisfy the algebraic equation:

$$(Z_1 - \frac{b}{2}Z_4)^2 + (Z_2 - \frac{b}{2}Z_3)^2 = \frac{a^2}{4}. \tag{3.16}$$

Eq.(3.16) and the range of b ($b_1 \leq b \leq b_2$) guarantee the motion is within workspace.

3.2.6 Planar RPR Open Chain

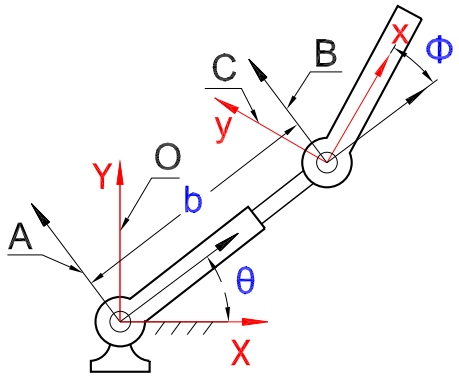

Figure 3.6: A planar RPR open chain.

The planar quaternion of the end link is given by

$$\mathbf{Z}(\theta, b, \phi) = \mathbf{Z}(\theta)\mathbf{X}(b)\mathbf{Z}(\phi), \tag{3.17}$$

where

$$\begin{aligned} Z_1 &= \frac{b}{2}\cos\frac{\theta-\phi}{2}, \\ Z_2 &= \frac{b}{2}\sin\frac{\theta-\phi}{2}, \\ Z_3 &= \sin\frac{\theta+\phi}{2}, \\ Z_4 &= \cos\frac{\theta+\phi}{2}. \end{aligned} \tag{3.18}$$

It follows that

$$\frac{b_1^2}{4} \leq Z_1^2 + Z_2^2 = \frac{b^2}{4} \leq \frac{b_2^2}{4}, \tag{3.19}$$

where we assume that $b_1 \leq b \leq b_2$.

3.2.7 Planar PRR Open Chain

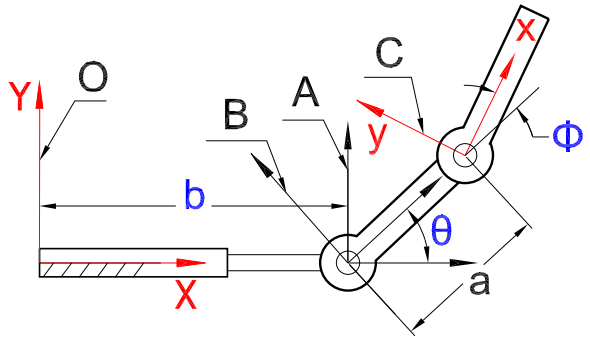

Figure 3.7: A planar PRR open chain.

The planar quaternion for the end link is given by

$$\mathbf{Z}(b, \theta, \phi) = \mathbf{X}(b)\mathbf{Z}(\theta)\mathbf{Z}(\phi), \tag{3.20}$$

where

$$\begin{aligned} Z_1 &= \frac{a}{2}\cos\frac{\theta-\phi}{2} + \frac{b}{2}\cos\frac{\theta+\phi}{2}, \\ Z_2 &= \frac{a}{2}\sin\frac{\theta-\phi}{2} - \frac{b}{2}\sin\frac{\theta+\phi}{2}, \\ Z_3 &= \sin\frac{\theta+\phi}{2}, \\ Z_4 &= \cos\frac{\theta+\phi}{2}. \end{aligned} \tag{3.21}$$

Thus the kinematic constraint associated with the PRR chain is given by

$$(Z_1 - \frac{b}{2})^2 + (Z_2 + \frac{b}{2}Z_3)^2 = \frac{a^2}{4}, \qquad b_1 \leq b \leq b_2. \tag{3.22}$$

3.2.8 Planar PRP Open Chain

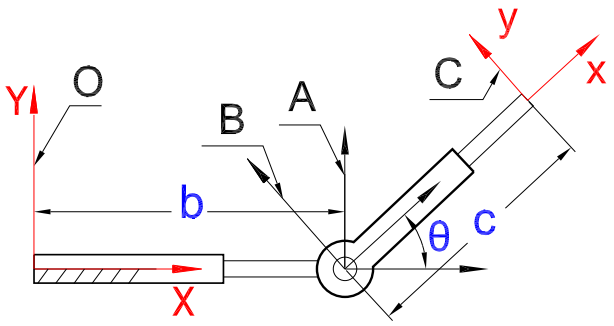

Figure 3.8: A planar PRP open chain.

The planar quaternion for the end link is given by

$$\mathbf{Z}(b, \theta, c) = \mathbf{X}(b)\mathbf{Z}(\theta)\mathbf{X}(c). \tag{3.23}$$

where

$$\begin{aligned} Z_1 &= \frac{b+c}{2}\cos\frac{\theta}{2}, \\ Z_2 &= \frac{c-b}{2}\sin\frac{\theta}{2}, \\ Z_3 &= \sin\frac{\theta}{2}, \\ Z_4 &= \cos\frac{\theta}{2}. \end{aligned} \tag{3.24}$$

The kinematic constraints for the chain are given by

$$\begin{aligned} Z_1 &= \frac{b+c}{2} Z_4, \\ Z_2 &= \frac{c-b}{2} Z_3, \\ b_1 &\le b \le b_2, \\ c_1 &\le c \le c_2. \end{aligned} \tag{3.25}$$

3.2.9 Planar 4R Closed Chain

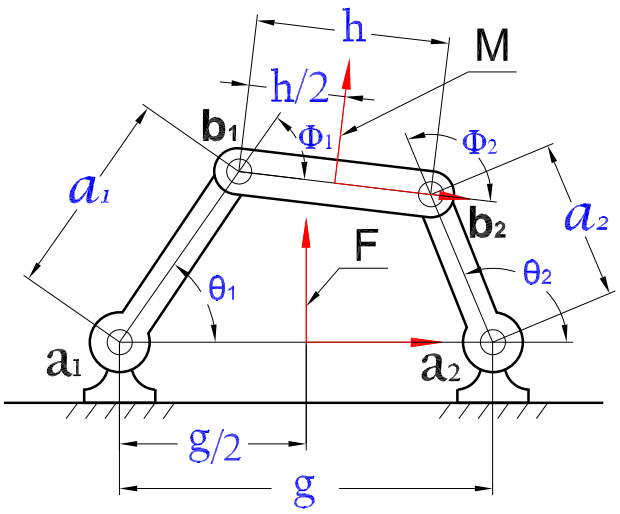

Figure 3.9: A planar 4R closed chain.

A planar 4R closed chain is composed of a pair of planar 2R open chains with their end links rigidly connected, see Figure 3.9. This planar 4R closed chain has one degree of freedom so the constraint manifold is a curve, which is the intersection of two constraint surfaces governed by two planar 2R open chains (see Jin and Ge [53, 54]).

Select the fixed frame $\mathbf{F}$ so that it is at the midpoint between $\mathbf{a}_1$ and $\mathbf{a}_2$, and the moving frame $\mathbf{M}$ at the midpoint between $\mathbf{b}_1$ and $\mathbf{b}_2$. The position of

the moving frame $\mathbf{M}$ is composed of a translation from $\mathbf{F}$ to $\mathbf{O}_1$ with a planar quaternion $\mathbf{G}_1 = (-g/4, 0, 0, 1)$, a displacement of $\mathbf{B}_1$ relative to $\mathbf{O}_1$ with $\mathbf{Z}(\theta_1, \phi_1)$, and another translation from $\mathbf{B}_1$ to $\mathbf{M}$ with a planar quaternion $\mathbf{H}_1 = (h/4, 0, 0, 1)$. The planar quaternion $\mathbf{Z}(\theta_1, \phi_1)$ is given by Eq. (3.2) (see Jin and Ge [53, 54]). Combining all these displacements, we obtain the following transformation from $\mathbf{F}$ to $\mathbf{M}$:

$$\mathbf{Y}_1(\theta_1, \phi_1) = \mathbf{G}_1\mathbf{Z}(\theta_1, \phi_1)\mathbf{H}_1 = [C_1]\mathbf{Z}(\theta_1, \phi_1), \tag{3.26}$$

where

$$[C_1] = [G_1^+][H_1^-] = \begin{bmatrix} 1 & 0 & 0 & -\tau \\ 0 & 1 & \sigma & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.27}$$

and

$$\sigma = (g + h)/4, \quad \tau = (g - h)/4. \tag{3.28}$$

Similarly the planar quaternion representing the other planar 2R open chain is given by:

$$\mathbf{Y}_2(\theta_2, \phi_2) = [C_2]\mathbf{Z}(\theta_2, \phi_2), \tag{3.29}$$

where $[C_2]$ is the inverse of $[C_1]$:

$$[C_2] = \begin{bmatrix} 1 & 0 & 0 & \tau \\ 0 & 1 & -\sigma & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.30}$$

The constraint curve for the planar 4R closed chain is the intersection of the constraint surfaces given by (3.26) and (3.29), that is $\mathbf{Y}_1 = \mathbf{Y}_2 = \mathbf{Y} =$

(Y_1, Y_2, Y_3, Y_4). The algebraic equations for the intersection are obtained by inverting (3.26) and (3.29):

$$\mathbf{Z}(\theta_1, \phi_1) = [C_2]\mathbf{Y}(\theta_1, \phi_1), \quad \mathbf{Z}(\theta_2, \phi_2) = [C_1]\mathbf{Y}(\theta_2, \phi_2). \tag{3.31}$$

This leads to the kinematic constraints in terms of planar quaternions:

$$\begin{aligned} \mathbf{Y}^T[Q_1]\mathbf{Y} &= 0, \\ \mathbf{Y}^T[Q_2]\mathbf{Y} &= 0. \end{aligned} \tag{3.32}$$

where $[Q_1],\ [Q_2]$ are:

$$[Q_1] = \begin{bmatrix} 1 & 0 & 0 & \tau \\ 0 & 1 & -\sigma & 0 \\ 0 & -\sigma & \sigma^2 - a_1^2/4 & 0 \\ \tau & 0 & 0 & \tau^2 - a_1^2/4 \end{bmatrix}, \tag{3.33}$$

and

$$[Q_2] = \begin{bmatrix} 1 & 0 & 0 & -\tau \\ 0 & 1 & \sigma & 0 \\ 0 & \sigma & \sigma^2 - a_2^2/4 & 0 \\ -\tau & 0 & 0 & \tau^2 - a_2^2/4 \end{bmatrix}. \tag{3.34}$$

The kinematic constraints as given by Eq. (3.32) can be rewritten as:

$$F_1(Y_1, Y_2, Y_3, Y_4) = \frac{a_1^2}{4}, \tag{3.35}$$

and

$$F_2(Y_1, Y_2, Y_3, Y_4) = \frac{a_2^2}{4}, \tag{3.36}$$

where $F_1(Y_1, Y_2, Y_3, Y_4)$ and $F_2(Y_1, Y_2, Y_3, Y_4)$ are given by:

$$F_1(Y_1, Y_2, Y_3, Y_4) = \frac{(Y_1 + \tau Y_4)^2 + (Y_2 - \sigma Y_3)^2}{Y_3^2 + Y_4^2}, \tag{3.37}$$

and

$$F_2(Y_1, Y_2, Y_3, Y_4) = \frac{(Y_1 - \tau Y_4)^2 + (Y_2 + \sigma Y_3)^2}{Y_3^2 + Y_4^2}. \tag{3.38}$$

The constraint equations of the form, (3.35) and (3.36), will be used for rational motion synthesis.

3.2.10 Planar 5R Closed Chain

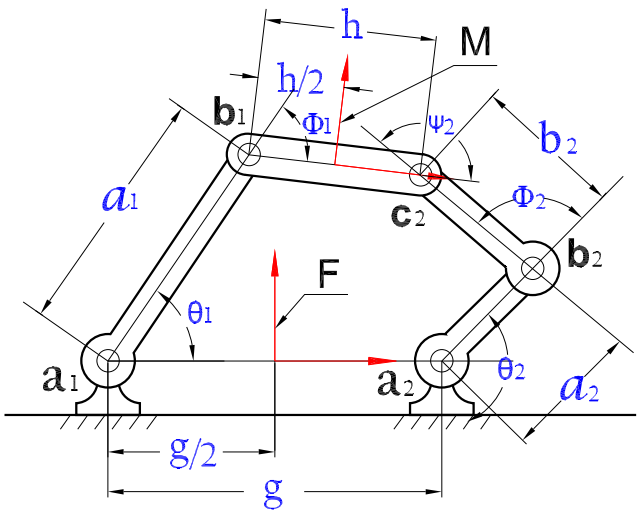

Figure 3.10: A planar 5R closed chain.

Consider a planar 5R closed chain, see Figure 3.10. The constraint manifold for this planar 5R closed chain is a portion of the constraint surface of a planar 2R open chain cut by the inner and outer boundaries of the constraint manifold of a planar 3R open chain.

The constraint surface of a planar 2R open chain is given by Eq. (3.26). The constraint manifold of a planar 3R open chain is given by (see Jin and

Ge [53]):

$$\mathbf{Y}_2(\theta_2, \phi_2, \psi_2) \;\; = \;\; [C_2]\mathbf{Z}(\theta_2, \phi_2, \psi_2), \tag{3.39}$$

where $[C_2]$ is given by Eq. (3.30) and $\mathbf{Z}(\theta_2, \phi_2, \psi_2)$ is given by Eq. (3.10).

The algebraic equations for the constraint manifolds associated with planar 2R and 3R chains can be obtained as:

$$\mathbf{Y}^T[Q_1]\mathbf{Y} = 0, \;\; \mathbf{Y}^T[Q_2(\phi_2)]\mathbf{Y} = 0. \tag{3.40}$$

where $[Q_1]$ is given by Eq. (3.33) and $[Q_2(\phi_2)]$ is given by:

$$[Q_2(\phi_2)] = \begin{bmatrix} 1 & 0 & 0 & -\tau \\ 0 & 1 & \sigma & 0 \\ 0 & \sigma & \sigma^2 - R_2^2(\phi_2)/4 & 0 \\ -\tau & 0 & 0 & \tau^2 - R_2^2(\phi_2)/4 \end{bmatrix}, \tag{3.41}$$

and

$$\begin{gathered} R_2^2(\phi_2) = a_2^2 + b_2^2 + 2a_2b_2\cos(\phi_2), \\ |a_2 - b_2| \leq R_2(\phi_2) \leq a_2 + b_2. \end{gathered} \tag{3.42}$$

We can rewrite Eq. (3.40) as

$$F_1(Y_1, Y_2, Y_3, Y_4) = \frac{a_1^2}{4}, \tag{3.43}$$

$$\frac{(a_2 - b_2)^2}{4} \leq F_2(Y_1, Y_2, Y_3, Y_4) \leq \frac{(a_2 + b_2)^2}{4}. \tag{3.44}$$

where $F_1(Y_1, Y_2, Y_3, Y_4)$ and $F_2(Y_1, Y_2, Y_3, Y_4)$ are given by Eqs. (3.37) and (3.38) respectively.

Eqs. (3.43) and (3.44) characterize the kinematic constraints of the planar 5R closed chain.

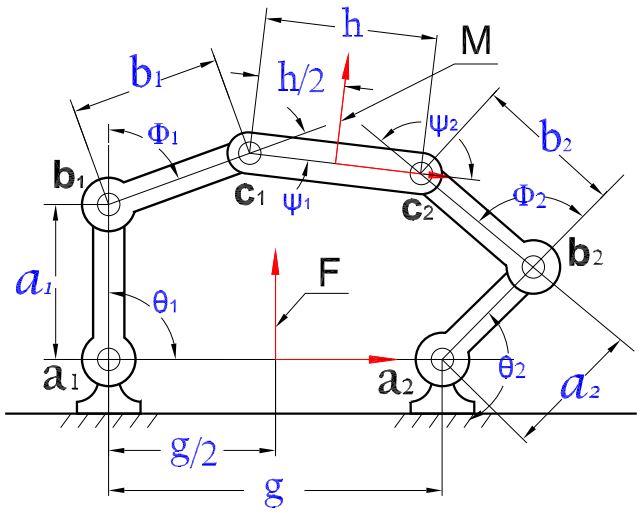

Figure 3.11: A planar 6R closed chain.

3.2.11 Planar 6R Closed Chain

Consider a planar 6R closed chain, see Figure 3.11. The constraint manifold for this planar 6R closed chain is the intersection of constraint manifolds of two planar 3R open chains. Similar to the case of a planar 5R closed chain, the two constraint manifolds for the 6R chain are given by

$$\begin{aligned} \mathbf{Y}_1(\theta_1, \phi_1, \psi_1) &= [C_1]\mathbf{Z}(\theta_1, \phi_1, \psi_1), \\ \mathbf{Y}_2(\theta_2, \phi_2, \psi_2) &= [C_2]\mathbf{Z}(\theta_2, \phi_2, \psi_2). \end{aligned} \tag{3.45}$$

The algebraic equations for the two manifolds are:

$$\mathbf{Y}^T[Q_1(\phi_1)]\mathbf{Y} = 0, \quad \mathbf{Y}^T[Q_2(\phi_2)]\mathbf{Y} = 0, \tag{3.46}$$

where $[Q_2(\phi_2)]$ is given by (3.41) and $[Q_1(\phi_1)]$ is:

$$[Q_1(\phi_1)] = \begin{bmatrix} 1 & 0 & 0 & \tau \\ 0 & 1 & -\sigma & 0 \\ 0 & -\sigma & \sigma^2 - R_1^2(\phi_1)/4 & 0 \\ \tau & 0 & 0 & \tau^2 - R_1^2(\phi_1)/4 \end{bmatrix}, \tag{3.47}$$

and

$$\begin{gathered} R_1^2(\phi_1) = a_1^2 + b_1^2 + 2a_1 b_1 \cos(\phi_1), \\ |a_1 - b_1| \leq R_1(\phi_1) \leq a_1 + b_1. \end{gathered} \tag{3.48}$$

We can rewrite Eq. (3.46) as:

$$\frac{(a_1 - b_1)^2}{4} \leq F_1(Y_1, Y_2, Y_3, Y_4) \leq \frac{(a_1 + b_1)^2}{4}, \tag{3.49}$$

$$\frac{(a_2 - b_2)^2}{4} \leq F_2(Y_1, Y_2, Y_3, Y_4) \leq \frac{(a_2 + b_2)^2}{4}. \tag{3.50}$$

where $F_1(Y_1, Y_2, Y_3, Y_4)$ and $F_2(Y_1, Y_2, Y_3, Y_4)$ are given by Eqs. (3.37) and (3.38).

Eqs. (3.49) and (3.50) characterize the kinematic constraints of the planar 6R closed chain.

3.3 Kinematic Constraints of Spherical Mechanisms

This section reviews the formulation of kinematic constraints of spherical 2R, 3R open chains and spherical 4R, 5R, and 6R closed chains.

3.3.1 Spherical 2R Open Chain

In this subsection, we review the kinematic constraint of a spherical 2R open chain that specifies the orientations obtainable by the end link of the arm.

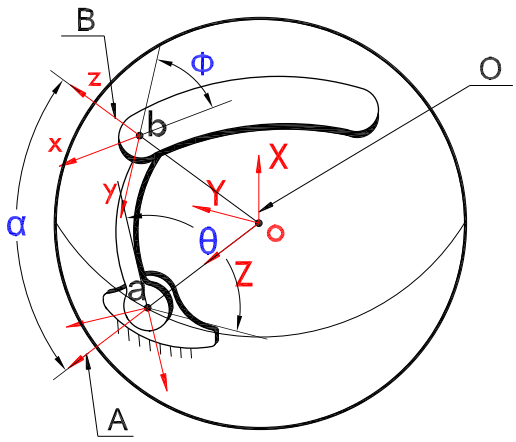

Figure 3.12: A Spherical 2R open chain.

Consider a spherical 2R open chain (see Figure 3.12) with joint axes **a**, **b** that intersect at a fixed point O. The joint axes make an angle α with respect to each other. We attach a fixed frame **O** to O and moving frames **A**, **B**, to each link. The joint angles are denoted as θ and ϕ for the first and the second joint, respectively. For details on the orientation of the moving frames, we refer the reader to McCarthy [10].

The orientation of the end link is given by the composition of a rotation θ about the **z**-axis of **A**, a rotation α about the displaced **x**-axis, followed by another rotation ϕ about the **z**-axis of **B**. These three rotations are given by the following quaternions:

$$\begin{aligned} \mathbf{Z}(\theta) &= (0, 0, \sin(\theta/2), \cos(\theta/2)), \\ \mathbf{X}(\alpha) &= (\sin(\alpha/2), 0, 0, \cos(\alpha/2)), \\ \mathbf{Z}(\phi) &= (0, 0, \sin(\phi/2), \cos(\phi/2)). \end{aligned} \tag{3.51}$$

The product of these quaternions gives a quaternion defining a general position

for the end link of the spherical 2R open chain:

$$\mathbf{q}(\theta, \alpha, \phi) = \mathbf{Z}(\theta)\mathbf{X}(\alpha)\mathbf{Z}(\phi). \tag{3.52}$$

Expanding Eq. (3.52), we obtain $\mathbf{q}(\theta, \alpha, \phi) = (q_1, q_2, q_3, q_4)$, where

$$\begin{aligned} q_1 &= \sin(\alpha/2)\cos((\theta-\phi)/2), \\ q_2 &= \sin(\alpha/2)\sin((\theta-\phi)/2), \\ q_3 &= \cos(\alpha/2)\sin((\theta+\phi)/2), \\ q_4 &= \cos(\alpha/2)\cos((\theta+\phi)/2). \end{aligned} \tag{3.53}$$

From Eq. (3.53), we can derive the following algebraic equation:

$$\frac{q_1^2 + q_2^2}{q_3^2 + q_4^2} = \tan^2(\alpha/2). \tag{3.54}$$

This equation characterizes the kinematic constraint of a spherical 2R open chain and is said to define the constraint manifold for the 2R open chain (McCarthy [10]).

The Eq. (3.54) is equivalent to:

$$\frac{q_1^2 + q_2^2}{q_1^2 + q_2^2 + q_3^2 + q_4^2} = \sin^2(\alpha/2). \tag{3.55}$$

or

$$\frac{q_3^2 + q_4^2}{q_1^2 + q_2^2 + q_3^2 + q_4^2} = \cos^2(\alpha/2). \tag{3.56}$$

Note that when q_i $(i = 1, 2, 3, 4)$ is replaced by $Q_i = wq_i$, where w is a scalar, Eq. (3.54), (3.55) and (3.56) are unchanged.

From Eqs. (3.55) and (3.56) it is clear that if a unit quaternion is used, these equations reduce to the equations of circle in two different planes. Thus, the problem of interpolating orientations of the end link of a spherical 2R

open chain can be reduced to that of circular interpolations in two separate planes. We note here that even though the kinematic constraint expressed by Eqs. (3.54), (3.55), and (3.56) are equivalent to one another, we still need to perform circular interpolation in two different planes. This is so because Eq. (3.55) and Eq. (3.56) each contribute only half of the four quaternion coordinates. As we will show later, the problem is greatly simplified since the circular interpolation is a standard problem in CAGD and has been effectively solved.

3.3.2 Spherical 3R Open Chain

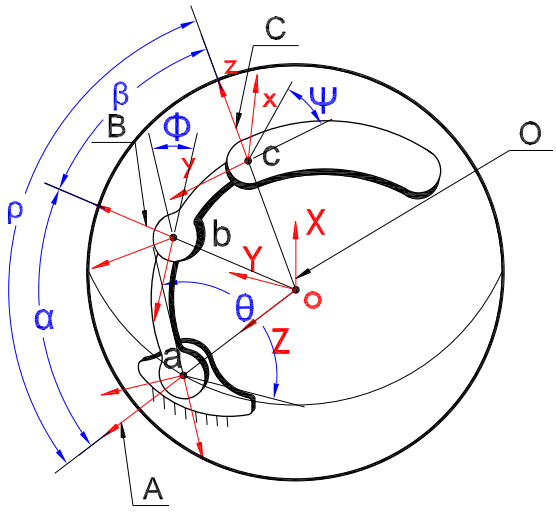

Figure 3.13: A spherical 3R open chain.

Consider a spherical 3R open chain (see Figure 3.13) with joint axes **a, b, c** intersecting a fixed point O. The axes **a, b** make an angle α and **b, c** make an angle β. We attach a fixed frame **O** to O and moving frames **A, B, C** to each links. The joint angles are denoted as θ, ϕ, and ψ for the successive joints.

For details on the orientation of the moving frames, we again refer the reader to McCarthy [10]. The orientation of the end link is given by the following quaternion product:

$$\mathbf{q}(\theta, \phi, \psi) = \mathbf{Z}(\theta)\mathbf{X}(\alpha)\mathbf{Z}(\phi)\mathbf{X}(\beta)\mathbf{Z}(\psi), \tag{3.57}$$

where,

$$\begin{aligned}
&\mathbf{Z}(\theta) = (0, 0, \sin(\theta/2), \cos(\theta/2)),\\
&\mathbf{X}(\alpha) = (\sin(\alpha/2), 0, 0, \cos(\alpha/2)),\\
&\mathbf{Z}(\phi) = (0, 0, \sin(\phi/2), \cos(\phi/2)),\\
&\mathbf{X}(\beta) = (\sin(\beta/2), 0, 0, \cos(\beta/2)),\\
&\mathbf{Z}(\psi) = (0, 0, \sin(\psi/2), \cos(\psi/2)).
\end{aligned} \tag{3.58}$$

By expanding the product in Eq. (3.57) we obtain $\mathbf{q}(\theta, \phi, \psi) = (q_1, q_2, q_3, q_4)$, where

$$\begin{aligned}
q_1 &= \cos(\frac{\phi}{2})\sin(\frac{\alpha+\beta}{2})\cos(\frac{\theta-\psi}{2}) + \sin(\frac{\phi}{2})\sin(\frac{\alpha-\beta}{2})\sin(\frac{\theta-\psi}{2}),\\
q_2 &= \cos(\frac{\phi}{2})\sin(\frac{\alpha+\beta}{2})\sin(\frac{\theta-\psi}{2}) - \sin(\frac{\phi}{2})\sin(\frac{\alpha-\beta}{2})\cos(\frac{\theta-\psi}{2}),\\
q_3 &= \cos(\frac{\phi}{2})\cos(\frac{\alpha+\beta}{2})\sin(\frac{\theta+\psi}{2}) + \sin(\frac{\phi}{2})\cos(\frac{\alpha-\beta}{2})\cos(\frac{\theta+\psi}{2}),\\
q_4 &= \cos(\frac{\phi}{2})\cos(\frac{\alpha+\beta}{2})\cos(\frac{\theta+\psi}{2}) - \sin(\frac{\phi}{2})\cos(\frac{\alpha-\beta}{2})\sin(\frac{\theta+\psi}{2}).
\end{aligned} \tag{3.59}$$

Equation (3.59) further gives:

$$\begin{aligned}
q_1^2 + q_2^2 &= \cos^2(\frac{\phi}{2})\sin^2(\frac{\alpha+\beta}{2}) + \sin^2(\frac{\phi}{2})\sin^2(\frac{\alpha-\beta}{2}),\\
q_3^2 + q_4^2 &= \cos^2(\frac{\phi}{2})\cos^2(\frac{\alpha+\beta}{2}) + \sin^2(\frac{\phi}{2})\cos^2(\frac{\alpha-\beta}{2}).
\end{aligned} \tag{3.60}$$

Since α, β satisfy the condition $0 < \alpha, \beta < \pi$, Eq. (3.60) reduces to the following inequality:

$$\tan^2((\alpha-\beta)/2) \leq \frac{q_1^2 + q_2^2}{q_3^2 + q_4^2} \leq \tan^2((\alpha+\beta)/2). \tag{3.61}$$

This inequality characterizes the kinematic constraint of a spherical 3R open chain. The Eq. (3.61) is equivalent to:

$$\sin^2((\alpha-\beta)/2) \leq \frac{q_1^2+q_2^2}{q_1^2+q_2^2+q_3^2+q_4^2} \leq \sin^2((\alpha+\beta)/2), \tag{3.62}$$

or

$$\cos^2((\alpha+\beta)/2) \leq \frac{q_3^2+q_4^2}{q_1^2+q_2^2+q_3^2+q_4^2} \leq \cos^2((\alpha-\beta)/2). \tag{3.63}$$

Note once again that when q_i $(i = 1, 2, 3, 4)$ is replaced by $Q_i = wq_i$, where w is a scalar, Eq. (3.61), (3.62) and (3.63) are unchanged.

We assert that the kinematic constraint given by Eq. (3.61) is frame independent. See **Appendix C** for a proof that uses spherical trigonometry.

3.3.3 Spherical 4R Closed Chain

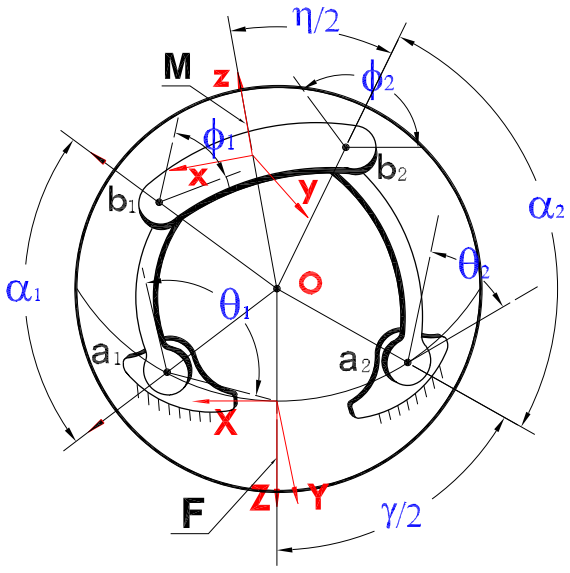

Figure 3.14: A spherical 4R closed chain

A 4R spherical closed chain can be assembled by joining the end links of two spherical 2R robot arms. Thus, the kinematic constraints of the 4R spherical closed chain can be obtained by first, appropriately transforming the location of the fixed and the moving frame, writing out the kinematic constraints of the end link from the perspective of the two 2R robot arms, and then equating those two perspectives.

Consider a spherical 4R robot arm (see Figure 3.14) with moving pivots at $\mathbf{b}_1$ and $\mathbf{b}_2$, and fixed pivots at $\mathbf{a}_1$ and $\mathbf{a}_2$. The joint axes at all the pivots intersect at a fixed point $\mathbf{O}$. The angle between the fixed pivots is denoted γ and the fixed frame $\mathbf{F}$ ($\mathbf{XYZ}$) is positioned midway along this angle with its x-axis normal to the plane defined by the fixed joints $\mathbf{a}_1$ and $\mathbf{a}_2$ in the direction $(\mathbf{a}_1 - \mathbf{O}) \times (\mathbf{a}_2 - \mathbf{O})$. The z-axis of the fixed frame $\mathbf{F}$ points in the direction of the bisector of the angle γ between vectors $(\mathbf{a}_1 - \mathbf{O})$ and $(\mathbf{a}_2 - \mathbf{O})$. The angle between the moving pivots is denoted η and the moving frame $\mathbf{M}$ ($\mathbf{xyz}$) is positioned midway along this angle. Similar to the orientation of the fixed frame, its x-axis is normal to the plane defined by the moving joints $\mathbf{b}_1$ and $\mathbf{b}_2$ in the direction $(\mathbf{b}_1 - \mathbf{O}) \times (\mathbf{b}_2 - \mathbf{O})$ and the z-axis bisects the angle η between the vectors $(\mathbf{b}_1 - \mathbf{O})$ and $(\mathbf{b}_2 - \mathbf{O})$. The joint angles for the first and the second 2R robot arm are denoted θ_1, ϕ_1 and θ_2, ϕ_2, respectively, while the angular spans of the two links joining the coupler are α_1 and α_2. For the first 2R robot arm, the moving frame $\mathbf{M}$ can be positioned with respect to the fixed frame $\mathbf{F}$ by the following quaternion product (see **Appendix A** for

quaternion products):

$$\mathbf{Y}_1(\theta_1, \phi_1) = \mathbf{X}(-\gamma/2)\mathbf{D}(\theta_1, \phi_1)\mathbf{X}(\eta/2), \tag{3.64}$$

where $\mathbf{X}(-\gamma/2)$ and $\mathbf{X}(\eta/2)$ represent rotations about $\mathbf{X}$ axis by angle $-\gamma/2$ and $\eta/2$, respectively. Using local coordinate transformation approach to building transformations for robotic chains, these two transformations bring the origin of the fixed frame $\mathbf{F}$ to the fixed pivot $\mathbf{a}_1$ and bring the moving pivot $\mathbf{b}_1$ to the origin of the moving frame $\mathbf{M}$, while $\mathbf{D}(\theta_1, \phi_1)$ represents the transformation of a spherical 2R chain. Equation (3.64) is given in the matrix form by

$$\mathbf{Y}_1(\theta_1, \phi_1) = [C_1]\mathbf{D}(\theta_1, \phi_1), \tag{3.65}$$

where, $[C_1]$ is a 4×4 matrix and $\mathbf{D}(\theta_1, \phi_1)$ is written as a 4×1 column vector. We note here that for brevity we use the same symbol to denote both a quaternion and a column vector and rely on the context to reveal which use is intended. Thus, for example, $\mathbf{D}(\theta_1, \phi_1)$ is written in the quaternion form in the quaternion product given by Eq. (3.64), while the same symbol in Eq. (3.65) involving multiplication with a matrix $[C_1]$ denotes a 4×1 column vector.

Similarly, for the second 2R robot arm, the moving frame $\mathbf{M}$ can be positioned with respect to the fixed frame $\mathbf{F}$ by the following transformation:

$$\mathbf{Y}_2(\theta_2, \phi_2) = \mathbf{X}(\gamma/2)\mathbf{D}(\theta_2, \phi_2)\mathbf{X}(-\eta/2), \tag{3.66}$$

Equation (3.66) is given in the matrix form by

$$\mathbf{Y}_2(\theta_2, \phi_2) = [C_2]\mathbf{D}(\theta_2, \phi_2), \tag{3.67}$$

where $\mathbf{D}(\theta_i, \phi_i); i = 1, 2$ have been derived in Purwar et. al. [4], and $\mathbf{X}(\pm\gamma/2) = (\pm\sin(\gamma/4), 0, 0, \cos(\gamma/4)); \mathbf{X}(\pm\eta/2) = (\pm\sin(\eta/4), 0, 0, \cos(\eta/4))$.

Matrices $[C_1]$ and $[C_2]$ in the Eqs. (3.65) and (3.67) are orthogonal matrices, given by

$$[C_1] = [\mathbf{X}(-\gamma/\mathbf{2})^+][\mathbf{X}(\eta/\mathbf{2})^-] = \begin{bmatrix} \cos\tau & 0 & 0 & -\sin\tau \\ 0 & \cos\sigma & \sin\sigma & 0 \\ 0 & -\sin\sigma & \cos\sigma & 0 \\ \sin\tau & 0 & 0 & \cos\tau \end{bmatrix}, \quad (3.68)$$

$$[C_2] = [\mathbf{X}(\gamma/\mathbf{2})^+][\mathbf{X}(-\eta/\mathbf{2})^-] = \begin{bmatrix} \cos\tau & 0 & 0 & \sin\tau \\ 0 & \cos\sigma & -\sin\sigma & 0 \\ 0 & \sin\sigma & \cos\sigma & 0 \\ -\sin\tau & 0 & 0 & \cos\tau \end{bmatrix}, \quad (3.69)$$

where $[\mathbf{X}(\pm\gamma/\mathbf{2})^+]\mathbf{and}[\mathbf{X}(\pm\eta/\mathbf{2})^-]$ are the matrix representations of their respective quaternion form (see McCarthy [10] for matrix representation of quaternion products) and

$$\sigma = (\gamma + \eta)/4, \tau = (\gamma - \eta)/4. \quad (3.70)$$

Equations (3.65) and (3.67) describe the constraint surfaces of the two spherical 2R robot arms. Their intersection gives us the kinematic constraints of the 4R spherical closed chain, that is $\mathbf{Y}_1 = \mathbf{Y}_2 = \mathbf{Y} = (Y_1, Y_2, Y_3, Y_4)$. The algebraic equations for the intersection are obtained by inverting Eqs. (3.65) and (3.67) and by using the fact that $[C_1][C_2] = [I]$:

$$\mathbf{D}(\theta_1, \phi_1) = [C_2]\mathbf{Y}_1(\theta_1, \phi_1), \quad \mathbf{D}(\theta_2, \phi_2) = [C_1]\mathbf{Y}_2(\theta_2, \phi_2). \quad (3.71)$$

Substituting (3.71) into the equation for the constraint surface of spherical 2R robot arm chain (see Purwar et. al. [4]), we obtain the equations for two quadrics:

$$\begin{aligned} \mathbf{Y}^T[Q_1]\mathbf{Y} &= 0, \\ \mathbf{Y}^T[Q_2]\mathbf{Y} &= 0. \end{aligned} \tag{3.72}$$

where $[Q_1],\ [Q_2]$ are:

$$[Q_1] = \begin{bmatrix} c^2\frac{\alpha_1}{2} - s^2\tau & 0 & 0 & s\tau c\tau \\ 0 & c^2\frac{\alpha_1}{2} - s^2\sigma & -s\sigma c\sigma & 0 \\ 0 & -s\sigma c\sigma & -s^2\frac{\alpha_1}{2} + s^2\sigma & 0 \\ s\tau c\tau & 0 & 0 & -s^2\frac{\alpha_1}{2} + s^2\tau \end{bmatrix}, \tag{3.73}$$

and

$$[Q_2] = \begin{bmatrix} c^2\frac{\alpha_2}{2} - s^2\tau & 0 & 0 & -s\tau c\tau \\ 0 & c^2\frac{\alpha_2}{2} - s^2\sigma & s\sigma c\sigma & 0 \\ 0 & s\sigma c\sigma & -s^2\frac{\alpha_2}{2} + s^2\sigma & 0 \\ -s\tau c\tau & 0 & 0 & -s^2\frac{\alpha_2}{2} + s^2\tau \end{bmatrix}. \tag{3.74}$$

In the above matrices, c and s represent the cosine and sine functions, respectively.

We expand the equation of the two quadrics given by Eq. (3.72) to:

$$F_1(Y_1, Y_2, Y_3, Y_4) = \cos^2\frac{\alpha_1}{2}, \tag{3.75}$$

$$F_2(Y_1, Y_2, Y_3, Y_4) = \cos^2\frac{\alpha_2}{2}. \tag{3.76}$$

where $F_1(Y_1, Y_2, Y_3, Y_4)$ and $F_2(Y_1, Y_2, Y_3, Y_4)$ are given by:

$$F_1(Y_1, Y_2, Y_3, Y_4) = \frac{(Y_1\sin\tau - Y_4\cos\tau)^2 + (Y_2\sin\sigma + Y_3\cos\sigma)^2}{Y_1^2 + Y_2^2 + Y_3^2 + Y_4^2}, \tag{3.77}$$

and

$$F_2(Y_1, Y_2, Y_3, Y_4) = \frac{(Y_1 \sin\tau + Y_4 \cos\tau)^2 + (Y_2 \sin\sigma - Y_3 \cos\sigma)^2}{Y_1^2 + Y_2^2 + Y_3^2 + Y_4^2}. \qquad (3.78)$$

Equations (3.75) and (3.76) characterize the kinematic constraints of a spherical 4R closed chain (McCarthy [10]) and are said to define Clifford-quadrics (Müller [55]). These kinematic constraints are homogeneous in nature since multiplying $Y_i (i = 1 \ldots 4)$ with a common scalar does not change these equations.

3.3.4 Spherical 5R Closed Chain

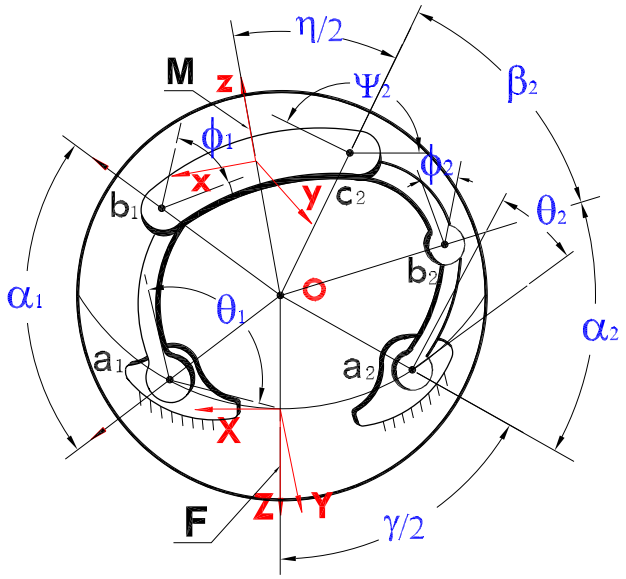

Figure 3.15: A spherical 5R closed chain

Consider a spherical 5R closed chain; see Figure 3.15. The constraint manifold for this spherical 5R closed chain is the intersection of the constraint

surface of a spherical 2R robot arm with the constraint solid of a spherical 3R robot arm.

The constraint surface of a spherical 2R open chain is given by Eq. (3.65). The constraint solid of a spherical 3R robot arm is given by transforming $\mathbf{D}(\theta_2, \phi_2, \psi_2)$ (see Purwar et.al. [4]) by matrix $[C_2]$:

$$\begin{aligned} \mathbf{Y}_1(\theta_1, \phi_1) &= [C_1]\mathbf{D}(\theta_1, \phi_1), \\ \mathbf{Y}_2(\theta_2, \phi_2, \psi_2) &= [C_2]\mathbf{D}(\theta_2, \phi_2, \psi_2). \end{aligned} \tag{3.79}$$

The algebraic equations for the intersection of above manifolds are:

$$\mathbf{Y}^T[Q_1]\mathbf{Y} = 0, \quad \mathbf{Y}^T[Q_2(\phi_2)]\mathbf{Y} = 0, \tag{3.80}$$

where $[Q_1]$ is given by Eq. (3.73) and $[Q_2(\phi_2)]$ is obtained by substituting α_2 in Eq. (3.74) with ρ_2. The angle $\rho_2 = \rho_2(\phi_2)$ defines the distance between the fixed pivot $\mathbf{a}_2$ and the moving pivot $\mathbf{c}_2$ of the second 3R spherical robot arm and varies according to joint angle ϕ_2 and the angular spans of the links of the 3R arm as follows:

$$\begin{gathered} \cos\rho_2 = \cos\alpha_2 \cos\beta_2 - \sin\alpha_2 \sin\beta_2 \cos\phi_2, \\ \cos(\alpha_2 + \beta_2) \leq \cos\rho_2 \leq \cos(\alpha_2 - \beta_2). \end{gathered} \tag{3.81}$$

The above inequality is always true since the link lengths given by α_2 and β_2 are never more than π. We can get the kinematic constraints of a spherical 5R closed chain by expanding Eq. (3.80):

$$F_1(Y_1, Y_2, Y_3, Y_4) = \cos^2\frac{\alpha_1}{2}, \tag{3.82}$$

$$\cos^2(\frac{\alpha_2 + \beta_2}{2}) \leq F_2(Y_1, Y_2, Y_3, Y_4) \leq \cos^2(\frac{\beta_2 - \alpha_2}{2}). \tag{3.83}$$

where $F_1(Y_1, Y_2, Y_3, Y_4)$ and $F_2(Y_1, Y_2, Y_3, Y_4)$ are given by Eqs. (3.77) and (3.78).

Equations (3.82) and (3.83) characterize the kinematic constraints of a spherical 5R closed chain and are said to define the constraint manifold for the spherical 5R closed chain (McCarthy [10]).

3.3.5 Spherical 6R Closed Chain

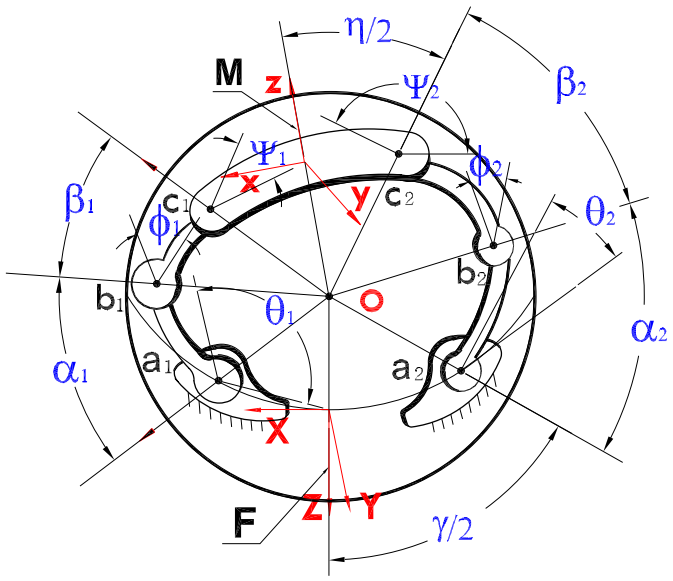

Figure 3.16: A spherical 6R closed chain

Consider a spherical 6R closed chain; see Figure 3.16. The constraint manifold for this spherical 6R closed chain is the intersection of constraint solids of two spherical 3R robot arms. In the figure, angles θ_1, ϕ_1, ψ_1, and θ_2, ϕ_2, ψ_2 denote the joint angles for the first and the second spherical 3R robot arms, respectively.

The constraint solids of two spherical 3R closed chains can be obtained by transforming $\mathbf{D}(\theta_1, \phi_1, \psi_1)$ by matrix $[C_1]$ and transforming $\mathbf{D}(\theta_2, \phi_2, \psi_2)$ by matrix $[C_2]$ (see Purwar et.al. [4] for expressions of the parameterized solids given by $\mathbf{D}(\theta_i, \phi_i, \psi_i); i = 1, 2$):

$$\begin{aligned} \mathbf{Y}_1(\theta_1, \phi_1, \psi_1) &= [C_1]\mathbf{D}(\theta_1, \phi_1, \psi_1), \\ \mathbf{Y}_2(\theta_2, \phi_2, \psi_2) &= [C_2]\mathbf{D}(\theta_2, \phi_2, \psi_2). \end{aligned} \tag{3.84}$$

The algebraic equations for the intersection of above manifolds are:

$$\mathbf{Y}^T[Q_1(\phi_1)]\mathbf{Y} = 0, \quad \mathbf{Y}^T[Q_2(\phi_2)]\mathbf{Y} = 0, \tag{3.85}$$

where $[Q_2(\phi_2)]$ has been already defined and $[Q_1(\phi_1)]$ is obtained by substituting α_1 in Eq. (3.73) with $\rho_1(\phi_1)$. The angle $\rho_1 = \rho_1(\phi_1)$ defines the distance between the fixed pivot $\mathbf{a}_1$ and the moving pivot $\mathbf{c}_1$ of the first 3R spherical robot arm and varies according to joint angle ϕ_1 and the angular spans of the links of the first 3R robot arm as follows:

$$\cos\rho_1 = \cos\alpha_1 \cos\beta_1 - \sin\alpha_1 \sin\beta_1 \cos\phi_1, \tag{3.86}$$

We can derive the kinematic constraints of a spherical 6R closed chain from Eq. (3.85):

$$\cos^2(\frac{\alpha_1 + \beta_1}{2}) \leq F_1(Y_1, Y_2, Y_3, Y_4) \leq \cos^2(\frac{\beta_1 - \alpha_1}{2}), \tag{3.87}$$

$$\cos^2(\frac{\alpha_2 + \beta_2}{2}) \leq F_2(Y_1, Y_2, Y_3, Y_4) \leq \cos^2(\frac{\beta_2 - \alpha_2}{2}), \tag{3.88}$$

where $F_1(Y_1, Y_2, Y_3, Y_4)$ and $F_2(Y_1, Y_2, Y_3, Y_4)$ are given by Eqs. (3.77) and (3.78).

Equations (3.87) and (3.88) characterize the kinematic constraints of a spherical 6R closed chain and are said to define the constraint manifold for the spherical 6R closed chain (McCarthy [10]).

3.4 Kinematic Constraints of Spatial Mechanisms

This section reviews the formulation of kinematic constraints of the spatial SS open chain, the ADEPT robot, and the PUMA robot.

3.4.1 Spatial SS Open Chain

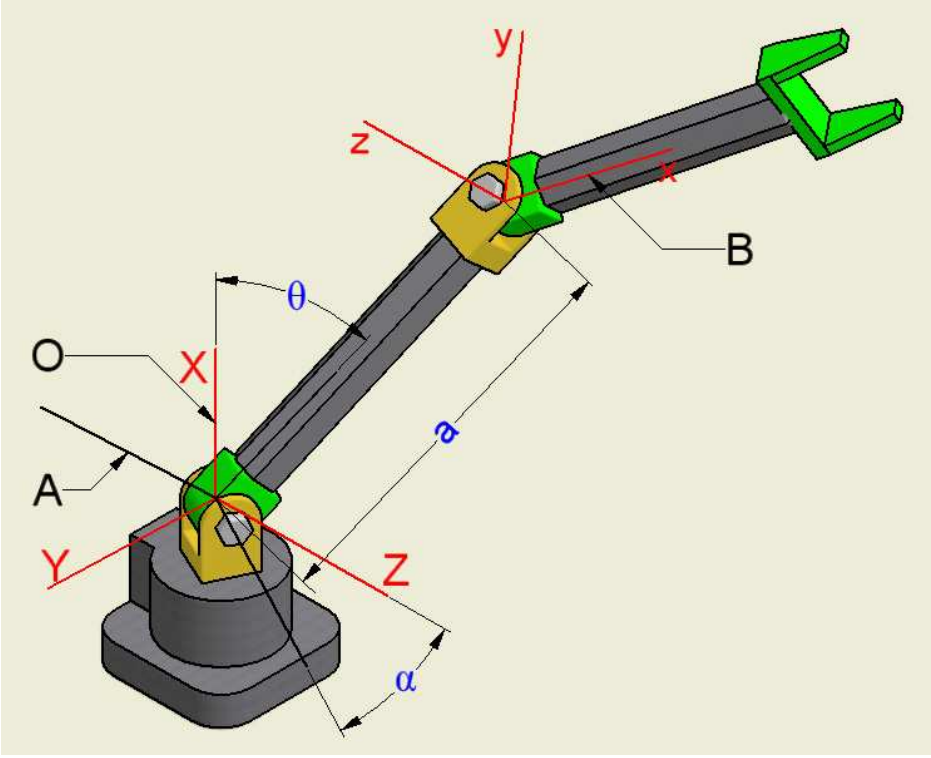

Figure 3.17: A spatial SS dyad.

A kinematic open chain connected by two spherical joints is called a *spatial SS open chain*, see Figure 3.17. We attach a fixed frame **O** to the base and moving frames **A, B** to each of the links. Initially the frames **A** and **O** coincide and **B** is located at the center of joint **b** with the same orientation with **O**. The orientation of **A** relative to **O** is prescribed by the angles α, θ, and δ about the x-axis, the displaced z, x-axis of **A**, respectively. We set $\delta = 0$ since this rotation does not contribute to the final configuration of the end link. Similarly, the orientation of **B** relative to **A** is prescribed by the angles β, ϕ, and γ. The position of **B** relative **A** is a translation along x-axis of **A** by the link length a. Thus the final configuration of the end link **B** relative to the base **O** is determined by angles α, θ, β, ϕ, γ and link length a.

The dual quaternion representing the transformation from **O** to **B** is:

$$\hat{\mathbf{q}}(\alpha, \theta, a, \beta, \phi, \gamma) = \mathbf{q} + \epsilon\mathbf{q}^0 = \mathbf{X}(\alpha)\mathbf{Z}(\theta)\mathbf{X}(a)\mathbf{X}(\beta)\mathbf{Z}(\phi)\mathbf{X}(\gamma), \tag{3.89}$$

where $\mathbf{X}(\alpha)$, $\mathbf{Z}(\theta)$, $\mathbf{X}(a)$, $\mathbf{X}(\beta)$, $\mathbf{Z}(\phi)$, and $\mathbf{X}(\gamma)$ are defined by:

$$\begin{aligned}
\mathbf{X}(\alpha) &= \sin(\frac{\alpha}{2})\mathbf{i} + \cos(\frac{\alpha}{2}), \\
\mathbf{Z}(\theta) &= \sin(\frac{\theta}{2})\mathbf{k} + \cos(\frac{\theta}{2}), \\
\mathbf{X}(a) &= (\frac{a}{2})\epsilon\mathbf{i} + 1, \\
\mathbf{X}(\beta) &= \sin(\frac{\beta}{2})\mathbf{i} + \cos(\frac{\beta}{2}), \\
\mathbf{Z}(\phi) &= \sin(\frac{\phi}{2})\mathbf{k} + \cos(\frac{\phi}{2}), \\
\mathbf{X}(\gamma) &= \sin(\frac{\gamma}{2})\mathbf{i} + \cos(\frac{\gamma}{2}).
\end{aligned} \tag{3.90}$$

Expand this product to obtain $\mathbf{q} = (q_1, q_2, q_3, q_4)$ and $\mathbf{q}^0 = (q_1^0, q_2^0, q_3^0, q_4^0)$,

where

$$
\begin{aligned}
q_1 &= \cos(\frac{\theta}{2})\cos(\frac{\phi}{2})\sin(\frac{\alpha+\beta+\gamma}{2}) - \sin(\frac{\theta}{2})\sin(\frac{\phi}{2})\sin(\frac{\alpha-\beta+\gamma}{2}), \quad (3.91)\\
q_2 &= -\cos(\frac{\theta}{2})\sin(\frac{\phi}{2})\sin(\frac{\alpha+\beta-\gamma}{2}) - \sin(\frac{\theta}{2})\cos(\frac{\phi}{2})\sin(\frac{\alpha-\beta-\gamma}{2}),\\
q_3 &= \cos(\frac{\theta}{2})\sin(\frac{\phi}{2})\cos(\frac{\alpha+\beta-\gamma}{2}) + \sin(\frac{\theta}{2})\cos(\frac{\phi}{2})\cos(\frac{\alpha-\beta-\gamma}{2}),\\
q_4 &= \cos(\frac{\theta}{2})\cos(\frac{\phi}{2})\cos(\frac{\alpha+\beta+\gamma}{2}) - \sin(\frac{\theta}{2})\sin(\frac{\phi}{2})\cos(\frac{\alpha-\beta+\gamma}{2}),\\
q_1^0 &= \frac{a}{2}\cos(\frac{\theta}{2})\cos(\frac{\phi}{2})\cos(\frac{\alpha+\beta+\gamma}{2}) + \frac{a}{2}\sin(\frac{\theta}{2})\sin(\frac{\phi}{2})\cos(\frac{\alpha-\beta+\gamma}{2}),\\
q_2^0 &= -\frac{a}{2}\cos(\frac{\theta}{2})\sin(\frac{\phi}{2})\cos(\frac{\alpha+\beta-\gamma}{2}) + \frac{a}{2}\sin(\frac{\theta}{2})\cos(\frac{\phi}{2})\cos(\frac{\alpha-\beta-\gamma}{2}),\\
q_3^0 &= -\frac{a}{2}\cos(\frac{\theta}{2})\sin(\frac{\phi}{2})\sin(\frac{\alpha+\beta-\gamma}{2}) + \frac{a}{2}\sin(\frac{\theta}{2})\cos(\frac{\phi}{2})\sin(\frac{\alpha-\beta-\gamma}{2}),\\
q_4^0 &= -\frac{a}{2}\cos(\frac{\theta}{2})\cos(\frac{\phi}{2})\sin(\frac{\alpha+\beta+\gamma}{2}) - \frac{a}{2}\sin(\frac{\theta}{2})\sin(\frac{\phi}{2})\sin(\frac{\alpha-\beta+\gamma}{2}).
\end{aligned}
$$

The components of $\mathbf{q}$ and $\mathbf{q}^0$ satisfy the relation

$$(a^2/4)(q_1^2+q_2^2+q_3^2+q_4^2) - ((q_1^0)^2+(q_2^0)^2+(q_3^0)^2+(q_4^0)^2) = 0. \quad (3.92)$$

Eq.(3.92) as well as the fundamental dual quaternion equations (2.16) and (2.17) characterize the kinematic constraints of a spatial SS open chain are said to define the constraint manifold for the spatial SS open chain (Ge [56]).

3.4.2 An ADEPT robot

Consider the ADEPT robot shown in Figure 3.18. We attach a fixed frame **O** to the base and moving frame **A**, **B**, **C**, and **D** to each links. The joint angles are denoted as θ, ϕ, and ψ. The displacement from the fixed fame **O** to the moving frame **D** is the composition of a rotation of **A** relative to **O**,

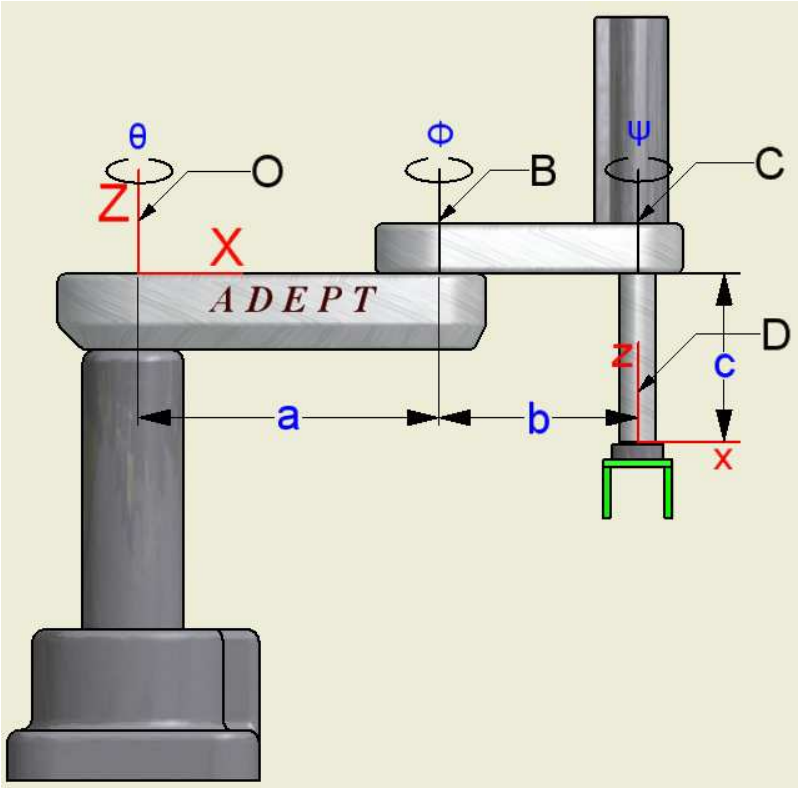

Figure 3.18: An ADEPT robot.

a translation from the frame **A** to the frame **B**, a rotation of **C** relative to **B**, a translation from the frame **B** to the frame **C**, a rotation of the frame **D** relative to **C**, and a translation from the frame **C** to the frame **D**. Their product is the dual quaternion defining a general position of the moving frame **D**:

$$\hat{\mathbf{q}}(\theta, a, \phi, b, \psi, c) = \mathbf{q} + \epsilon\mathbf{q}^0 = \mathbf{Z}(\theta)\mathbf{X}(a)\mathbf{Z}(\phi)\mathbf{X}(b)\mathbf{Z}(\psi)\mathbf{Z}(c), \tag{3.93}$$

where $\mathbf{Z}(\theta)$, $\mathbf{X}(a)$, $\mathbf{Z}(\phi)$, $\mathbf{X}(b)$, $\mathbf{Z}(\psi)$, and $\mathbf{Z}(c)$ are given by:

$$\begin{aligned}
\mathbf{Z}(\theta) &= \sin(\frac{\theta}{2})\mathbf{k} + \cos(\frac{\theta}{2}), \\
\mathbf{X}(a) &= (\frac{a}{2})\epsilon\mathbf{i} + 1, \\
\mathbf{Z}(\phi) &= \sin(\frac{\phi}{2})\mathbf{k} + \cos(\frac{\phi}{2}), \\
\mathbf{X}(b) &= (\frac{b}{2})\epsilon\mathbf{i} + 1, \\
\mathbf{Z}(\psi) &= \sin(\frac{\psi}{2})\mathbf{k} + \cos(\frac{\psi}{2}), \\
\mathbf{Z}(c) &= (\frac{c}{2})\epsilon\mathbf{k} + 1,
\end{aligned} \tag{3.94}$$

Expand this product to obtain $\mathbf{q} = (q_1, q_2, q_3, q_4)$ and $\mathbf{q}^0 = (q_1^0, q_2^0, q_3^0, q_4^0)$, where

$$\begin{aligned}
q_1 &= 0, \\
q_2 &= 0, \\
q_3 &= \sin(\frac{\alpha+\phi+\psi}{2}), \\
q_4 &= \cos(\frac{\alpha+\phi+\psi}{2}), \\
q_1^0 &= \frac{a}{2}\cos(\frac{\alpha-\phi-\psi}{2}) + \frac{b}{2}\cos(\frac{\alpha+\phi-\psi}{2}), \\
q_2^0 &= \frac{a}{2}\sin(\frac{\alpha-\phi-\psi}{2}) + \frac{b}{2}\sin(\frac{\alpha+\phi-\psi}{2}), \\
q_3^0 &= \frac{c}{2}\cos(\frac{\alpha+\phi+\psi}{2}), \\
q_4^0 &= -\frac{c}{2}\sin(\frac{\alpha+\phi+\psi}{2}).
\end{aligned} \tag{3.95}$$

The components of $\mathbf{q}$ and $\mathbf{q}^0$ satisfy the relation

$$q_3^2 + q_4^2 = 1, \tag{3.96}$$

$$\frac{(a-b)^2}{4} \leq (q_1^0)^2 + (q_2^0)^2 = \frac{a^2}{4} + \frac{b^2}{4} + \frac{ab}{2}\cos(\phi) \leq \frac{(a+b)^2}{4}, \tag{3.97}$$

$$q_3^0 = \frac{c}{2} q_4, \tag{3.98}$$

$$q_4^0 = -\frac{c}{2} q_3. \tag{3.99}$$

Eqs. (3.96), (3.97), (3.98), and (3.99) characterize the kinematic constraints of a ADEPT robot.

Chapter 4

Constrained Motion Interpolation for Planar and Spherical 2R and 3R Open Chains

4.1 Introduction

The purpose of this chapter is to study the problem of synthesizing rational motions of a rigid body under kinematic constraints that are imposed by planar and spherical 2R and 3R open chains. Through the use of planar quaternions, it is shown that the problem of rational motion interpolation under the kinematic constraints of a planar 2R open chain can be reduced to that of circular interpolations in two separate planes. Furthermore, the problem of synthesizing the Cartesian rational motion of a planar 3R open chain can be reduced to that of circular interpolation in one plane and constrained spline interpolation in a circular ring in the other plane. Similarly, through the use of quaternions, it is shown that the problem of synthesizing the Cartesian rational motion of

a spherical 2R open chain can be reduced to that of circular interpolations in two separate planes. The problem of synthesizing the Cartesian rational motion of a spherical 3R open chain can be reduced to that of constrained spline interpolation in two different planes.

The organization of this chapter is as follows. Section 4.2 presents an algorithm for C^1 piecewise rational Bézier interpolation on a circle. Section 4.3 shows an algorithm for rational B-spline interpolation within an n-spherical shell. Section 4.4 presents algorithms for rational motion interpolation under kinematic constrains of planar and spherical 2R and 3R open chains.

4.2 Piecewise Rational Bézier Interpolation on a Circle

In this section, we deal with the problem of interpolating a given set of two dimensional points on a circle using a piecewise rational Bézier representation. The resulting algorithm will be used in later sections for generating Cartesian rational motions for planar and spherical open chains.

We first review the construction of a circular arc using a rational Bézier representation (see Forrest [57], Piegl and Tiller [58, 59] for details).

Let $\mathbf{b}_0,\ \mathbf{b}_2$, denote the start and end points of a circular arc of radius r, since the middle Bézier point $\mathbf{b}_1$ should be the intersection point of two lines that are tangent to the circle at the points $\mathbf{b}_0$, $\mathbf{b}_2$ respectively, we have

$$\mathbf{b}_1 = \frac{r^2(\mathbf{b}_0 + \mathbf{b}_2)}{r^2 + \mathbf{b}_0 \cdot \mathbf{b}_2}. \tag{4.1}$$

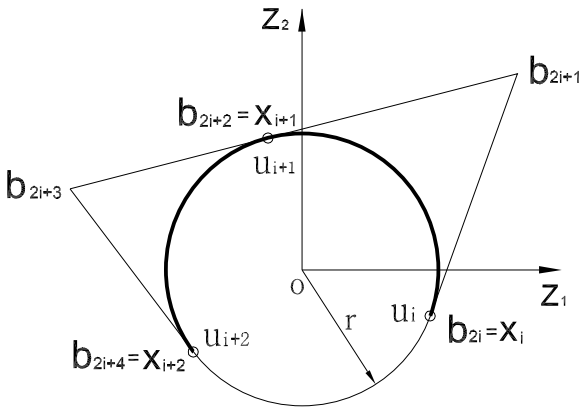

Figure 4.1: Quadratic NURB circular arc.

where "·" denotes the dot product of two vectors. In addition, the relationship of the weights w_i related to Bézier points $\mathbf{b}_i$ should be:

$$w_1^2 = \frac{w_0 w_2}{2}[1 + (\frac{\mathbf{b}_0}{r}) \cdot (\frac{\mathbf{b}_2}{r})]. \tag{4.2}$$

We now present a simple algorithm for interpolating a set of points on a circle with a C^1 piecewise rational Bézier curve (Figure 4.1). The problem is defined as follows:

Given: A set of points $\mathbf{x}_0, ..., \mathbf{x}_L$ on a circle with radius r and the corresponding parameter values (or knots) $u_0, ..., u_L$.

Find: A piecewise quadratic rational Bézier circular arc $\mathbf{b}(u)$ that interpolates the given data points at the specified parameter values, i.e., $\mathbf{b}(u_i) = \mathbf{x}_i$.

In view of Eq. (4.1), Bézier control points of the piecewise rational Bézier curve are given by:

$$\mathbf{b}_{2i} = \mathbf{x}_i, \quad \mathbf{b}_{2i+1} = \frac{r^2(\mathbf{x}_i + \mathbf{x}_{i+1})}{r^2 + \mathbf{x}_i \cdot \mathbf{x}_{i+1}}, \quad \mathbf{b}_{2i+2} = \mathbf{x}_{i+1}. \tag{4.3}$$

Let $\triangle_i = u_{i+1} - u_i$ and let $s_{2i} = w_{2i+1}/w_{2i}$ denote the weight ratio, where $i = 0, ..., L-1$. The weights (or weight ratios) must be selected such that each rational Bézier segment is a circular arc:

$$s_{2i} = \frac{s_{2i+1}}{2}[1 + (\frac{\mathbf{x}_i}{r}) \cdot (\frac{\mathbf{x}_{i+1}}{r})], \ i = 0, ..., L-1. \tag{4.4}$$

and that two adjacent Bézier segments have C^1 continuity at the junction point $\mathbf{b}_{2(i+1)} = \mathbf{x}_{i+1}$:

$$\frac{|\mathbf{b}_{2i+3} - \mathbf{b}_{2i+2}|}{|\mathbf{b}_{2i+2} - \mathbf{b}_{2i+1}|} = \frac{\triangle_{i+1}}{\triangle_i s_{2i+1} s_{2i+2}}, \ i = 0, ..., L-2. \tag{4.5}$$

where $|\cdot|$ denotes the magnitude of a vector. Without loss of generality, we choose $s_0 = 1$ ($w_0 = 1, \ w_1 = 1$).

In summary, the algorithm for C^1 piecewise rational Bézier circular interpolation is given by:

Algorithm 4.2

1. Find control points $\mathbf{b}_{2i}$ from the given points $\mathbf{x}_i$ using Eq. (4.3).

2. For i^{th} piece of Bézier circular arc use $[u_i, \ u_{i+1}]$ as the range for the parameter u and calculate the weights associated with each of the Bézier points from Eqs. (4.4) and (4.5).

3. Generate the C^1 interpolating non-uniform rational Bézier (NURB) circular arc from its piecewise rational Bézier form.

4.3 Rational B-spline Interpolation inside an n-spherical Shell

In this section, we deal with the problem of interpolating a given set of points within an *n-spherical shell* using a rational B-spline curve, where *n-spherical shell* means in-between of two n dimensional concentric hyperspheres. The resulting algorithm will be used in later sections for generating Cartesian rational motions for planar and spherical open chains.

Given: A set of points $\mathbf{x}_i,\ i = 0, ..., L$ within an n-spherical shell in $\mathbb{R}^n$ as well as corresponding parameter values u_i. Let r_{min} and r_{max} denote the inner and the outer radius of the n-spherical shell respectively.

Find: A rational B-spline curve $\mathbf{b}(u)$ that interpolates through the given points at the corresponding parameter values.

Algorithm 4.3

1. Interpolate the given points $\mathbf{x}_i$ with a B-spline curve $\mathbf{b}(u)$.

2. Of all the points on the curve $\mathbf{b}(u)$, find those that have the maximum and minimum distances to the origin and record the corresponding parameter values.

3. Verify if the magnitudes (distances to the origin) of those extreme points are in the range of $[r_{min}, r_{max}]$.

4. If yes, stop; if not, connect a line from the origin to the extreme point and then select a new point on this line such that the new point is within

the range of $[r_{min}, r_{max}]$. Add this new point as well as the associated parameter value to the list of points to be interpolated and go to step 1.

Remark 4.3.1. In step 1 of above algorithm, a standard scheme for B-spline interpolation is used (see, for example, Piegl and Tiller [15]). However, this scheme does not guarantee that the entire curve would stay within the n-spherical shell.

Remark 4.3.2. When we add a new point in step 4 of above algorithm, if the distance of the extreme point is smaller than r_{min}, then use $r_{min} + \delta$ to generate a new point, and if the distance of the extreme point is greater than r_{max}, then use $r_{max} - \delta$ to generate a new point. Here, δ is a specified tolerance that defines the minimum distance of the new point from the boundary of the n-spherical shell.

Remark 4.3.3. If $n = 2$, then the n-spherical shell becomes circular ring; when $n = 1$ we interpolate given one dimensional points within the boundary $[r_{min}, r_{max}]$. If we consider the parameter value as the other coordinate, then the problem becomes interpolation of two dimensional points within a band with the boundary $[r_{min}, r_{max}]$.

We now address the issue of how to determine the extreme points on the rational B-spline curve that have maximum or minimum distance to the origin. For each of the Bézier segments, the distance from a point with parameter u to the origin is given by

$$|\mathbf{s}_i(u)| = |\sum_{j=0}^{3} B_j^3(\frac{u - u_i}{u_{i+1} - u_i})\mathbf{b}_{3i+j}|;\ i = 0, ...,\ L-1;\ u \in [u_i, u_{i+1}). \qquad (4.6)$$

i	1	2	3	4	5	6
Z_{i1}	3.0	2.0	0.0	-2.0	-2.5	0.0
Z_{i2}	0.0	1.2	3.0	2.4	1.0	-2.1
u_i	0.00	0.14	0.38	0.52	0.67	1.00

Table 4.1: **Input data for Algorithm 4.3**.

where $B_j^3(\frac{u-u_i}{u_{i+1}-u_i})$ are *Bernstein Polynomials* of degree 3. The parameter u associated with the extreme point is given by the following condition:

$$\frac{\partial}{\partial u}(|\mathbf{s}_i(u)|)^2 = 0. \tag{4.7}$$

This results in a quintic equation in terms of u, which can be solved either analytically or numerically for the parameter value u in the range $[u_i, u_{i+1})$.

We give an example to demonstrate the **Algorithm 4.3**. A set of two dimensional points is given in Table 4.1. Figure 4.2 shows that a portion of the cubic B-spline curve is outside the circular ring and thus violates the circular ring constraint. As the results of **Algorithm 4.3**, Figure 4.3 shows the constrained interpolation with $\delta = 1$.

4.4 Rational Motions of Planar and Spherical 2R and 3R Open Chains

In this section, we present algorithms for synthesizing piecewise rational motions of planar and spherical 2R and 3R open chains that interpolate a set of given positions of the end link.

The given positions of the end link of a planar open chain can be specified using either Cartesian based parameters $(d_{1i}, d_{2i}, \alpha_i)$ or joint coordinates

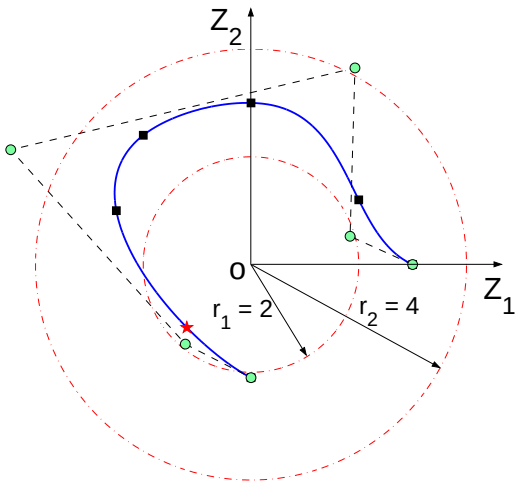

Figure 4.2: An unconstrained C^2 cubic B-spline interpolation of a set of points. In this case, a portion of the curve is outside of the ring and thus violates the constraint. The symbol "$\star$" represents the extreme point of B-spline curve that is outside the circular ring and has minimum distance to the center; the symbol "$\blacksquare$" represents the coordinates of the given points; the symbol "$\bullet$" represents the control points of B-spline curve.

(θ_i, ϕ_i) of the open chain. In the case of Cartesian parameters, we use Eq. (2.6) to convert them into planar quaternions; in the case of joint parameters, we convert them to planar quaternions by using Eq. (3.2) for planar 2R open chain and Eq.(3.10) for planar 3R open chain.

The orientations of the end link of a spherical 2R open chain can be given by either Cartesian coordinates $(s_{xi}, s_{yi}, s_{zi}, \theta_i)$ or joint coordinates (θ_i, ϕ_i). Similarly, the orientations of the end link of a spherical 3R open chain can be given by either Cartesian coordinates $(s_{xi}, s_{yi}, s_{zi}, \theta_i)$ or joint coordinates $(\theta_i, \phi_i, \psi_i)$. Eq. (2.1) can be used to convert Cartesian coordinates into quaternions, while Eq. (3.53) or Eq. (3.59) can be used to convert joint coordinates into quaternions for spherical 2R and 3R open chains, respectively.

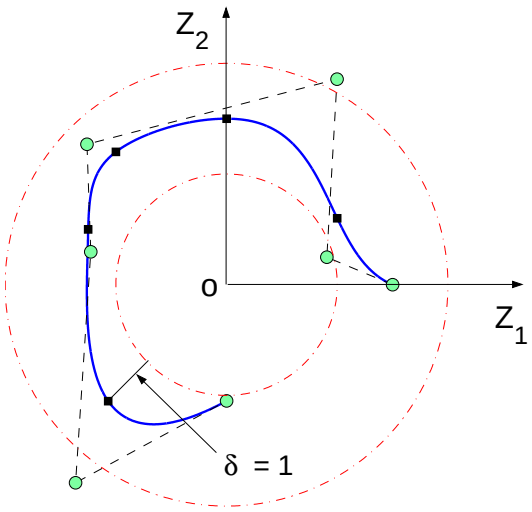

Figure 4.3: A constrained C^2 cubic B-spline interpolation in a circular ring with $\delta = 1$.

4.4.1 C^1 Interpolating Rational Motions for Planar and Spherical 2R Open Chains

This section presents an algorithm for the following constrained motion interpolation problem:

Given: A set of the positions of the end link of a planar or a spherical 2R open chain in its workspace as well as the parameter values $u_i,\ i = 0, ..., L$ and the length of the first link a of the planar 2R open chain and the length of the first link α of the spherical 2R open chain.

Find: A C^1 rational motion of the end link that interpolates through the given positions at the parameter values and satisfies the kinematic constraints of the planar or the spherical 2R open chain.

Algorithm 4.4.1

1. Convert the given positions of the second link into the planar quaternions

$\mathbf{Z}_i = (Z_{i1}, Z_{i2}, Z_{i3}, Z_{i4})$ using Eq. (2.6) or Eq. (3.2).

2. Separate the planar quaternions into: $\mathbf{E}_i = (Z_{i1}, Z_{i2});\ \mathbf{F}_i = (Z_{i3}, Z_{i4})$.

3. Since all given positions are assumed to be within the workspace of the planar 2R open chain, $\mathbf{E}_i$ must satisfy Eq. (3.3). This means the points $\mathbf{E}_i$ lie on a circle of radius $a/2$. Let $\mathbf{x}_i = \mathbf{E}_i,\ r = a/2$ and we use **Algorithm 4.2** to obtain a C^1 interpolating circular NURB curve $\mathbf{E}(u)$.

4. Similarly, $\mathbf{F}_i$ must satisfy Eq. (3.4) and we let $\mathbf{x}_i = \mathbf{F}_i,\ r = 1$ and use **Algorithm 4.2** once again to obtain another C^1 interpolating circular NURB curve $\mathbf{F}(u)$.

5. The image curve, $\mathbf{Z}(u) = (\mathbf{E}(u), \mathbf{F}(u))$, is a C^1 continuous quadratic rational curve. Substituting $\mathbf{Z}(u)$ into Eq. (2.10), we obtain a C^1 continuous quartic rational motion for the planar 2R open chain that interpolates the given set of positions of the end link.

Remark 4.4.1. From Eqs. (3.55) and (3.56) we can see that the kinematic constraints of a spherical 2R open chain are similar to those of a planar 2R open chain, the only difference is the radius of circle constraint. We can use **Algorithm 4.4.1** for the rational motion interpolation of a spherical 2R open chain with the modification $r = \sin(\alpha/2)$ in the step 3 and $r = \cos(\alpha/2)$ in the step 4.

In the following example, we are given five key positions of the second link for planning the motion of a planar 2R open chain. These positions

i	1	2	3	4	5
$\theta_i(^\circ)$	10	50	90	130	160
$\phi_i(^\circ)$	0	-10	0	20	20
u_i	0.0	0.2	0.5	0.8	1.0

Table 4.2: **Input data for a planar 2R open chain** (a = 4).

as well as the dimensions of the planar open chain are given in Table 4.2. We apply **Algorithm 4.4.1** to interpolate $\mathbf{E}_i$ and $\mathbf{F}_i$ with two quadratic circular arcs $\mathbf{E}(u)$ and $\mathbf{F}(u)$ respectively, see Figure 4.4 and Figure 4.5. Then $\mathbf{Z}(u) = (\mathbf{E}(u), \mathbf{F}(u))$ is the C^1 rational motion that interpolates given key positions of the second link.

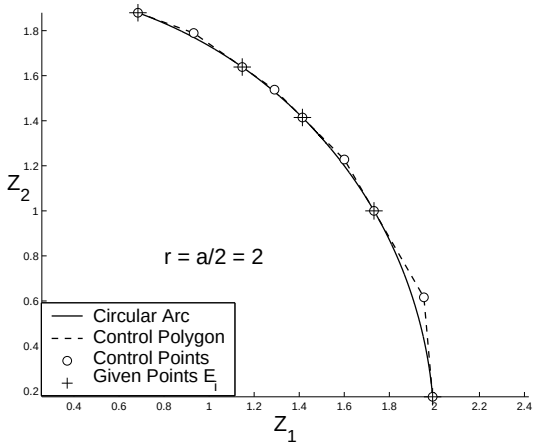

Figure 4.4: Interpolation of points $\mathbf{E}_i$ (planar 2R open chain).

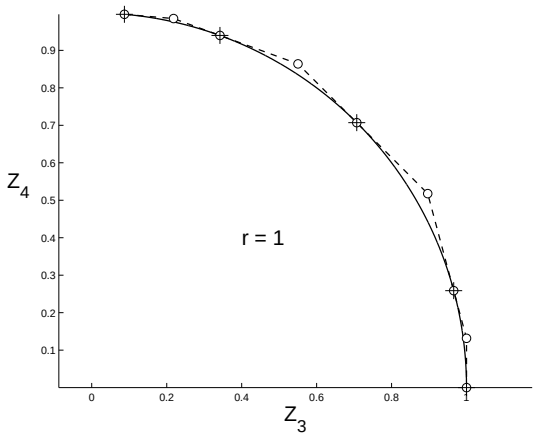

Figure 4.5: Interpolation of points $\mathbf{F}_i$ (planar 2R open chain).

4.4.2 C^2 Interpolating Rational Motions that Approximate the Kinematic Constraints for Planar 2R and 3R Open Chains

The approach advocated in section 4.4.1 has a limitation in that it can generate only C^1 (i.e., velocity-continuous) piecewise rational motions. In high-speed applications, a C^2 (i.e., acceleration-continuous) motion is often desired.

For this purpose we consider the kinematic constraints Eqs. (3.3), (3.4), and (3.12) (equation of circle) as very "thin" circular rings. Then we apply **Algorithm 4.3** to generate C^2 rational motions that approximate the kinematic constraints for planar 2R and 3R open chains. We still separate the planar quaternions $\mathbf{Z}_i = (Z_{i1}, Z_{i2}, Z_{i3}, Z_{i4})$ into: $\mathbf{E}_i = (Z_{i1}, Z_{i2})$; $\mathbf{F}_i = (Z_{i3}, Z_{i4})$, and use the following range $[r_{min}, r_{max}]$ when applying **Algorithm 4.3**:

- Planar 2R open chain:
  - Interpolating $\mathbf{E}_i$: $r_{min} = a/2 - \delta/2$, $r_{max} = a/2 + \delta/2$

– Interpolating $\mathbf{F}_i : r_{min} = 1 - \delta/2,\ r_{max} = 1 + \delta/2$

- Planar 3R open chain:

  – Interpolating $\mathbf{E}_i : r_{min} = |a - b|/2,\ r_{max} = (a + b)/2$

  – Interpolating $\mathbf{F}_i : r_{min} = 1 - \delta/2,\ r_{max} = 1 + \delta/2$

Here δ is a deviation value, it can be decided by the diametral clearance of open chains. In reality there is diametral clearance in each revolute joints of planar 2R and 3R open chains, see Figure 4.6. Because of diametral clearance c_{d1}, c_{d2} for the first and the second joints, length of the first link is within the range $[a - c_{d1} - c_{d2}, a + c_{d1} + c_{d2}]$, so if we choose the deviation value $\delta = c_{d1} + c_{d2}$ for the planar 2R open chain then the generated C^2 rational motion is obtainable by this planar 2R robot open chain which is perfect.

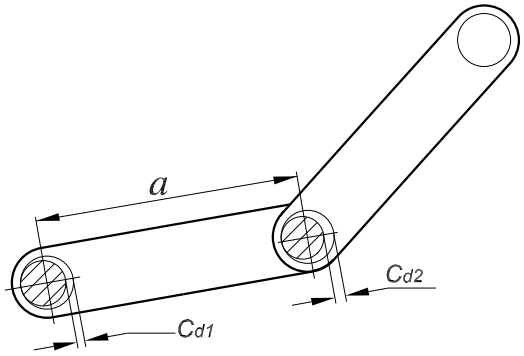

Figure 4.6: Diametral clearance of planar 2R open chain.

In the following example, we still use five given positions of a planar 3R open chain as listed in Table 4.3. We apply **Algorithm 4.3** to interpolate both $\mathbf{E}_i$ and $\mathbf{F}_i$ with a cubic B-spline curve, here we use the deviation value

i	1	2	3	4	5
$\theta_i(^\circ)$	20	60	90	120	170
$\phi_i(^\circ)$	0	30	40	50	80
$\psi_i(^\circ)$	10	20	30	40	50
u_i	0.0	0.2	0.5	0.8	1.0

Table 4.3: **Input data for planar 3R open chain** (a = 4, b = 3).

$\delta = 0.02$, see Figures 4.7, 4.8, and 4.9. The approximation converges quickly, it is achieved with two iterations.

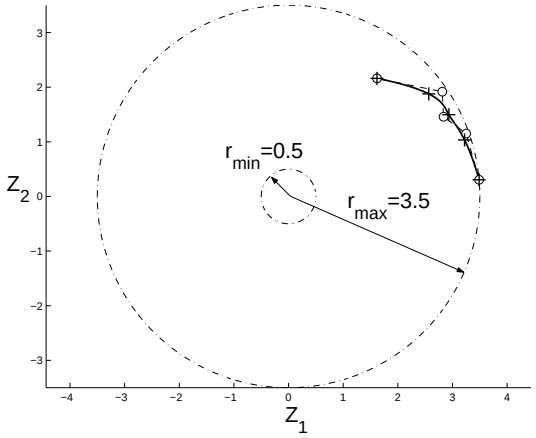

Figure 4.7: Interpolation of points $\mathbf{E}_i$ (planar 3R open chain).

4.4.3 C^2 Interpolating Rational Motions that Approximate the Kinematic Constraints for Spherical 2R and 3R Open Chains

Spherical 3R Open Chain

This subsection presents an algorithm for the following constrained motion interpolation problem:

Given: A set of orientations of the end link of a spherical 3R open chain in

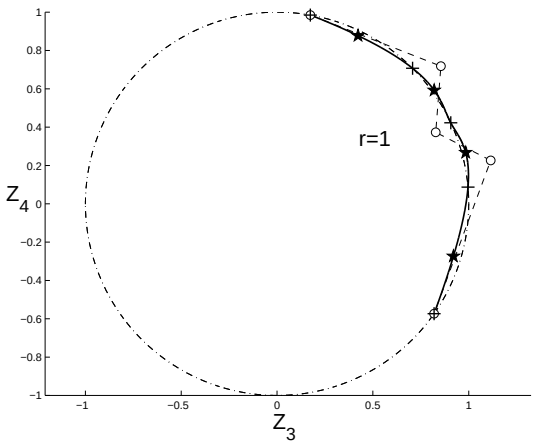

Figure 4.8: Unconstrained interpolation of points $\mathbf{F}_i$ (planar 3R open chain).

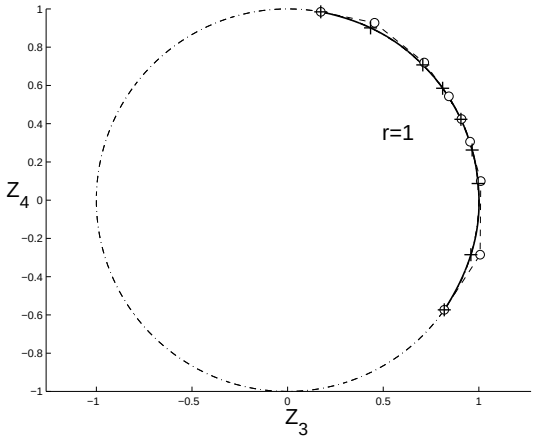

Figure 4.9: Constrained interpolation of points $\mathbf{F}_i$ (planar 3R open chain).

its workspace, the corresponding parameter values u_i, the associated weights $w_i,\ i = 0 \ldots L$, and link lengths α and β of the first and second link, respectively.

Find: A smooth (C^2 or higher) rational motion of the end link that interpolates the given orientations at the corresponding parameter values subject to the kinematic constraints of the spherical 3R open chain.

Then kinematic constraint for spherical 3R open chain is given by inequalities in Eq. (3.62) or Eq. (3.63). Geometrically, the limits in these inequalities represent two rings of varying radius in two different planes. The radius of these two rings change along the parameter u because the denominator in the inequalities is not necessarily unity. Since the motion of the end link is required to satisfy these kinematic constraints, we can reduce the problem of designing a smooth rational motion of the end link to that of constructing two separate spline curves confined to lie in the region defined by the two rings. In the succeeding discussion, we assume that the given orientations of the end link have been converted into unit quaternions $\mathbf{q}_i = (q_{i1}, q_{i2}, q_{i3}, q_{i4})$ using either Eq. (2.1) or Eq. (3.59), and thereafter into the non-unit quaternions $\mathbf{Q}_i (= (Q_{i1}, Q_{i2}, Q_{i3}, Q_{i4})) = w_i \mathbf{q}_i$.

We now present a sketch of the algorithm: We divide the quaternion components $\mathbf{Q}_i = (Q_{i1}, Q_{i2}, Q_{i3}, Q_{i4})$ in two groups $(\mathbf{E}_i, \mathbf{F}_i)$, where $\mathbf{E}_i = (Q_{i1}, Q_{i2})$ and $\mathbf{F}_i = (Q_{i3}, Q_{i4})$. Then, we construct two C^2 cubic B-spline curves[1] $\mathbf{E}(u)$ and $\mathbf{F}(u)$ that interpolate $\mathbf{E}_i$ and $\mathbf{F}_i$ at parameter values u_i, respectively. Let us call the plane defined by the first two quaternion coordinates (q_1, q_2) as $s_1 s_2$ plane, and by the last two quaternion coordinates (q_3, q_4) as $s_3 s_4$ plane. The curve $\mathbf{E}(u)$ lies in the $s_1 s_2$ plane, while the curve $\mathbf{F}(u)$ lies in the $s_3 s_4$ plane. However, these two curves are not guaranteed to lie inside the rings described earlier. In the Cartesian space, this means that the kinematic constraint given by Eq. (3.61) are not satisfied at all values of the parameter u. The Kine-

[1]Designing a C^2 B-spline curve is a standard scheme in CAGD (Farin [11], Hoschek and Lasser [14], and Piegl and Tiller [15]).

matic constraint (Eq. (3.61)) for a continuous valued parametric curve can be rewritten as:

$$\tan^2((\alpha-\beta)/2) \leq g(u) \leq \tan^2((\alpha+\beta)/2), \tag{4.8}$$

where

$$g(u) = \frac{\mathbf{E}(u)\cdot\mathbf{E}(u)}{\mathbf{F}(u)\cdot\mathbf{F}(u)}.$$

The idea behind the algorithm to be presented shortly is to first detect all the extrema of the function $g(u)$ by using piecewise Bézier representation of the B-spline curves. Then we test if an extremum violates the constraint. If it does, then we replace such an extremum with a new point that satisfies the constraint. We call an extremum that violates the constraint as an extreme point. In adding a new point, we impose following restrictions on it: 1) it should be inside the rings, 2) it should be minimally away from the extreme point, and 3) it should be added for the same parameter value as that of the extreme point. The idea is to deform the curve minimally in the vicinity of the extreme point by the addition of this new point that is just inside the ring. This new point is added to the set of points to be interpolated and new C^2 B-spline curves that interpolate these points are constructed. This process is repeated until no extreme points are detected. At the end, we have a C^2 B-spline constrained curve that does not violate the constraint. We note here that even though the two curves detect a common extreme point since the constraints given by Eqs. (3.62) and (3.63) are equivalent to each other, it is still necessary to construct two B-spline curves in each of the s_1s_2 and s_3s_4 plane. The curves $\mathbf{E}(u)$ and $\mathbf{F}(u)$ contribute two coordinates each to make up

the complete quaternion curve $\mathbf{Q}(u) = (\mathbf{E}(u), \mathbf{F}(u))$.

Now we show how the extreme and new points are calculated. We calculate the extrema of function $g(u)$ to detect violations of the kinematic constraint. The extrema of $g(u)$ can be calculated easily by taking the derivative of the piecewise Bézier form of $\mathbf{E}(u)$ and $\mathbf{F}(u)$. For the i^{th} pair of Bézier segments, $\mathbf{E}_i(u)$ and $\mathbf{F}_i(u)$, with the parameter value $u \in [u_i, u_{i+1})$, we determine the extrema of the following function:

$$g_i(u) = \frac{\mathbf{E}_i(u) \cdot \mathbf{E}_i(u)}{\mathbf{F}_i(u) \cdot \mathbf{F}_i(u)}. \tag{4.9}$$

By differentiating (4.9) and setting the resulting derivative to zero we obtain:

$$\frac{\mathbf{E}_i(u) \cdot \dot{\mathbf{E}}_i(u)}{\mathbf{E}_i(u) \cdot \mathbf{E}_i(u)} = \frac{\mathbf{F}_i(u) \cdot \dot{\mathbf{F}}_i(u)}{\mathbf{F}_i(u) \cdot \mathbf{F}_i(u)}. \tag{4.10}$$

where $\dot{\mathbf{E}}_i(u)$ and $\dot{\mathbf{F}}_i(u)$ denote the derivatives of $\mathbf{E}_i(u)$ and $\mathbf{F}_i(u)$ with respect to u. Equation (4.10) is a polynomial equation of degree 11 but due to the cancellation of the highest order term, it actually reduces to a equation of degree 10, and can be solved numerically for the parameter value u in the range $[u_i, u_{i+1})$. This procedure may yield many local extrema for different Bézier segments.

Let us assume that $g(u)$ has an extremum at $u = u^*$ that violates the constraint. Thus, we would like to add a new point $(\mathbf{E}_j, \mathbf{F}_j)$ at the parameter value u^*. We connect a line from the origin to $\mathbf{E}(u^*)$ and select $\mathbf{E}_j$ on this line; similarly we connect another line from the origin to $\mathbf{F}(u^*)$ and then select $\mathbf{F}_j$ on this line, such that $g(u^*)$ satisfies the inequality given in Eq. (4.8). We use following rules for locating $\mathbf{E}_j$ and $\mathbf{F}_j$ on the respective lines:

If

$$\frac{|\mathbf{E}(u^*)|}{|\mathbf{F}(u^*)|} < |\tan((\alpha-\beta)/2)|,$$

then

$$\frac{|\mathbf{E}_j|}{|\mathbf{F}_j|} = |\tan((\alpha-\beta)/2)| + \delta, \tag{4.11}$$

and if

$$\frac{|\mathbf{E}(u^*)|}{|\mathbf{F}(u^*)|} > |\tan((\alpha+\beta)/2)|,$$

then

$$\frac{|\mathbf{E}_j|}{|\mathbf{F}_j|} = |\tan((\alpha+\beta)/2)| - \delta, \tag{4.12}$$

where δ is a user defined tolerance value, which can be chosen as small as possible. If an extremum is found at the limits of the inequality in Eq. (4.8), the constraints are considered satisfied and thus, no new points are added in that case. We note here that due to limitations of numerical computing, we actually determine if the difference between the extremum of function $g(u)$ and the limit values is smaller than a very small number, instead of directly comparing the two numbers for equality. The tolerance parameter δ controls the change in the shape of the curve between successive iterations. These rules seek to add a new point that is minimally away from the extreme point and satisfies the kinematic constraint. We note that choosing a truly inner point, instead of a new point that is just inside a boundary of the ring may yield a curve that is unlikely to go out of the ring, this would not satisfy the desired condition of minimal change in the curve. The new point $\mathbf{E}_j, \mathbf{F}_j$ should also satisfy:

$$\mathbf{E}_j \cdot \mathbf{E}_j + \mathbf{F}_j \cdot \mathbf{F}_j = w_j^2. \tag{4.13}$$

Here, without any loss of generality, we choose $w_j = 1$.

Thus, the magnitude of new point $\mathbf{E}_j, \mathbf{F}_j$ can be calculated from Eq. (4.11) or (4.12), and Eq. (4.13). Since we already know the slope of the lines joining origin with the extreme point in two planes, we can uniquely locate the new point. Now, we present the algorithm:

Algorithm 4.4.3

1. Convert the given orientations of the end link into unit quaternions $\mathbf{q}_i = (q_{i1}, q_{i2}, q_{i3}, q_{i4})$ using either Eq. (2.1) or Eq. (3.59). Convert unit quaternions $\mathbf{q}_i$ into non-unit quaternions $\mathbf{Q}_i (= (Q_{i1}, Q_{i2}, Q_{i3}, Q_{i4})) = w_i \mathbf{q}_i$.

2. Group quaternion components as: $\mathbf{E}_i = (Q_{i1}, Q_{i2})$; $\mathbf{F}_i = (Q_{i3}, Q_{i4})$.

3. Construct a C^2 cubic B-spline curve $\mathbf{E}(u)$ that interpolates $\mathbf{E}_i$ and construct a C^2 cubic B-spline curve $\mathbf{F}(u)$ that interpolates $\mathbf{F}_i$ at parameter values u_i.

4. Calculate the extrema of the function $g(u)$ (Eq. (4.10)). For each extremum, test if $(\tan^2((\alpha - \beta)/2) \leq \text{extremum } (g(u)) \leq \tan^2((\alpha + \beta)/2)$. If yes, then the kinematic constraint of a spherical 3R open chain are satisfied. Go to step 5. If an extremum point fails the test, say at $u = u^*$, do the following and repeat for each extreme point:

   (a) Connect a line from the origin to $\mathbf{E}(u^*)$ and add $\mathbf{E}_j$ on this line. Calculate the magnitude of $\mathbf{E}_j$ from Eq. (4.11) or (4.12), and Eq. (4.13).

(b) Connect a line from the origin to $\mathbf{F}(u^*)$ and add $\mathbf{F}_j$ on this line. Calculate the magnitude of $\mathbf{F}_j$ from Eq. (4.11) or (4.12), and Eq. (4.13).

(c) Locate $\mathbf{E}_j$ and $\mathbf{F}_j$ on the lines.

(d) Add the new point $(\mathbf{E}_j, \mathbf{F}_j)$ at the parameter value u^* to the list of points to be interpolated and go to step 3.

5. The image curve $\mathbf{Q}(u) = (\mathbf{E}(u), \mathbf{F}(u))$ defines the C^2 interpolating piecewise rational motion of degree 6 after the substitution into Eq. (2.5).

Spherical 2R Open Chain

In this subsection, we show that we can use a slightly modified form of **Algorithm 4.4.3** to plan the C^2 rational motion for a spherical 2R open chain that approximates the kinematic constraints.

First, we transform the kinematic constraints of the spherical 2R open chain to a form similar to that of spherical 3R open chain. The idea is to transform the circular constraints of spherical 2R open chain to the ring constraints. Considering that in practice, there is always some clearance at the revolute joints of the open chain, the link length α is not exact but actually a function of diametric clearance at both the joints. Thus, the kinematic constraint of the spherical 2R open chain (Eq. (3.54)) can be modified as:

$$\tan^2(\alpha/2) - \delta \leq h(u) \leq \tan^2(\alpha/2) + \delta, \tag{4.14}$$

where

$$h(u) = \frac{q_1^2 + q_2^2}{q_3^2 + q_4^2}, \tag{4.15}$$

and δ is an indicator of the clearance at joints. The value of δ can be calculated from the diametric clearance at the joints. For a smaller value of δ, the kinematic constraint is better approximated. This form of the modified constraint equation is similar to the one derived for spherical 3R open chain and thus, the **Algorithm 4.4.3** can be applied. For a given value of δ, the constraints are considered violated, if the extrema of $h(u)$ do not satisfy the inequality in Eq. (4.14).

Let us assume that $h(u)$ has an extremum at $u = u^*$ that violates the constraint. Thus, we would like to add a new point $(\mathbf{E}_j, \mathbf{F}_j)$ at the parameter value u^*. Following the kinematic constraints of 2R open chains given by Eq. (3.55) and (3.56), the new point $(\mathbf{E}_j, \mathbf{F}_j)$ should satisfy:

$$|\mathbf{E}_j| = w_j \sin(\alpha/2);\ |\mathbf{F}_j| = w_j \cos(\alpha/2). \tag{4.16}$$

Here, again without any loss of generality, we choose $w_j = 1$.

Examples

In this section, we present two examples – one for the C^2 interpolating motion of a spherical 2R open chain, and the other for the C^2 interpolating motion of spherical 3R open chain. These examples demonstrate the working of the algorithms presented before. Tables 4.4 and 4.5 give the values of the joint coordinates used in the examples for five orientations along with their parameter values for spherical 2R and 3R open chain, respectively.

In the first example we have five orientations for Cartesian rational motion planning of a spherical 2R open chain (see Table 4.4 for input data). We

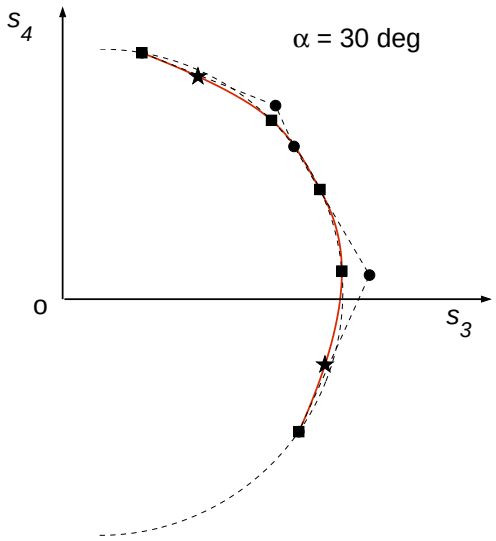

Figure 4.10: An unconstrained C^2 cubic B-spline interpolation of a set of points in the s_3s_4 plane.

use **Algorithm 4.4.3** to plan a C^2 rational motion for the same spherical 2R open chain and the same set of five orientations. This is demonstrated in Figures 4.10 and 4.11, which show the results of constraining the curve $\mathbf{F}(u)$ in the s_3s_4 plane, with $\delta = 0.002$. Note that in all the figures shown in this section, the symbol "■" represents first or the last two coordinates (i.e., $\mathbf{E}_i = (Q_{i1}, Q_{i2})$ or $\mathbf{F}_i = (Q_{i3}, Q_{i4})$) of the quaternions associated with the given five orientations. These are the orientations to be interpolated. The symbol "•" represents the deboor control points of the resulting B-spline curve; the symbol "⋆" represents the extreme points of the B-spline curve. In first iteration, the algorithm finds two extreme points (Figure 4.10), one each on the two Bézier segments. Thus, the algorithm adds two new points corresponding

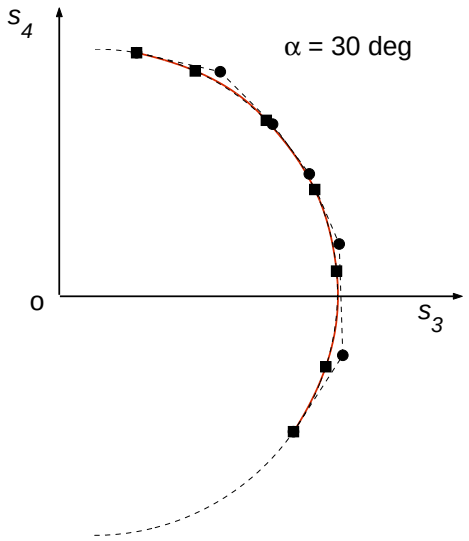

Figure 4.11: A constrained C^2 cubic B-spline interpolation of a set of points in the s_3s_4 plane.

i	1	2	3	4	5
α(deg)	30				
θ_i(deg)	0	45	90	125	165
ϕ_i(deg)	0	15	30	40	30
u_i	0.0	0.3	0.6	0.8	1.0

Table 4.4: **Joint coordinates for spherical 2R open chain motion planning.**

to the parameter value of the extreme points and in the next iteration re-interpolates seven (five originally given, and two newly added) orientations. In the second iteration, the kinematic constraints are satisfied approximately (Figure 4.11). The same procedure is repeated for the curve $\mathbf{E}(u)$ as well. In the end, we have a constrained curve $\mathbf{Q}(u)$ that satisfies the constraint of a spherical 2R robot open chain approximately.

i	1	2	3	4	5
α(deg)	45				
β(deg)	30				
θ_i(deg)	20	70	90	120	170
ϕ_i(deg)	0	30	40	50	80
ψ_i(deg)	10	30	40	60	90
u_i	0.0	0.2	0.5	0.7	1.0

Table 4.5: **Joint coordinates for spherical 3R open chain motion planning.**

	$\mathbf{Q}(u) = (\mathbf{E}(u), \mathbf{F}(u))$
i	$\mathbf{D}_i$
0	$(0.6064, 0.0531, 0.2053, 0.7663)$
1	$(0.4951, 0.1167, 0.6587, 0.5544)$
2	$(0.5915, 0.2021, 0.7772, 0.0725)$
3	$(0.4953, 0.1791, 0.8327, 0.1707)$
4	$(0.3862, 0.2059, 0.5419, -0.7175)$
5	$(0.4112, 0.2355, 0.0559, -0.8788)$

Table 4.6: **Control points of B-spline curve $\mathbf{Q}(u) = (\mathbf{E}(u), \mathbf{F}(u))$.**

In the second example, we use five given orientations for motion planning of a spherical 3R open chain (see Table 4.5 for input data). Figures 4.12, 4.13, 4.14 and 4.15 demonstrate the **Algorithm 4.4.3**. In Figures 4.12 and 4.14, the "wiggly" curves $r_1(u)$ and $r_2(u)$ depict the lower and the upper limits, respectively of the inequality in Eq. (3.62), while in Figures 4.13 and 4.15, curves $r_3(u)$ and $r_4(u)$ depict the lower and the upper limits, respectively of the inequality in Eq. (3.63). Figure 4.12 shows the unconstrained curve $\mathbf{E}(u)$ in the s_1s_2 plane and the Figure 4.13 shows the unconstrained curve $\mathbf{F}(u)$ in the s_3s_4 plane. The algorithm detects one extreme point outside the limits of the inequality (Eq. (4.8)) at $u^* = 0.6241$. This extreme point is common to the curves $\mathbf{E}(u)$ and $\mathbf{F}(u)$. The algorithm adds a new point on the line joining origin and this extreme point in each plane and re-interpolates. We note here that the equivalence of the inequalities given by Eq. (3.62) and Eq. (3.63) gives rise to a common extreme point for both the curves but both the curves still need to be constructed and constrained independently since they each contribute only half of the quaternion coordinates of $\mathbf{Q}(u)$. Similarly, the new point can be generated only by combining the coordinates of point E_j and F_j from coordinate planes s_1s_2 and s_3s_4, respectively. The curve is constrained in the next iteration; see Fig 4.14 and Fig 4.15. The algorithm converges in two iterations for a value of $\delta = 0.02$. Table 4.6 gives the deboor control orientations D_i of the constrained C_2 B-spline curve.

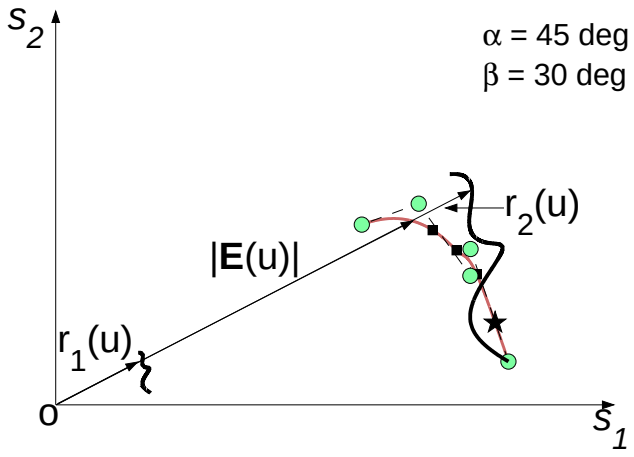

Figure 4.12: An unconstrained C^2 cubic B-spline interpolation of a set of points in s_1s_2 plane; $r_1(u) = |w\sin((\alpha-\beta)/2)|$, $r_2(u) = |w\sin((\alpha+\beta)/2)|$, $w^2 = Q_1^2(u) + Q_2^2(u) + Q_3^2(u) + Q_4^2(u)$.

4.5 Conclusions

In this chapter, we studied the problem of synthesizing piecewise rational motions subject to the kinematic constraints of the planar and spherical 2R and 3R open chains. We presented several algorithms for synthesizing exact C^1 continuous piecewise rational motions for planar and spherical 2R and 3R open chains. We also showed an algorithm for synthesizing exact C^2 continuous piecewise rational motions for a spherical 3R open chain and an algorithm for synthesizing C^2 continuous piecewise rational motions that approximate the kinematic constraints for a planar 3R open chain.

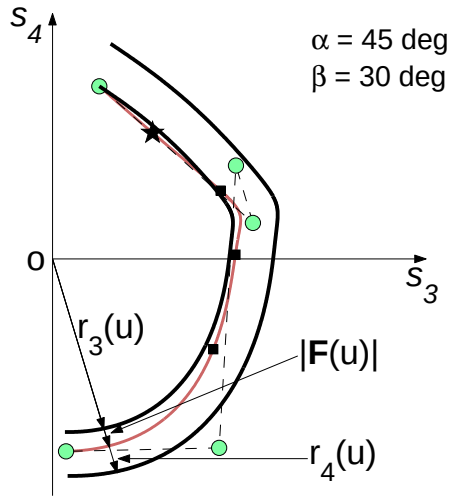

Figure 4.13: An unconstrained C^2 cubic B-spline interpolation of a set of points in s_3s_4 plane; $r_3(u) = |w\cos((\alpha+\beta)/2)|$, $r_4(u) = |w\cos((\alpha-\beta)/2)|$, $w^2 = Q_1^2(u) + Q_2^2(u) + Q_3^2(u) + Q_4^2(u)$.

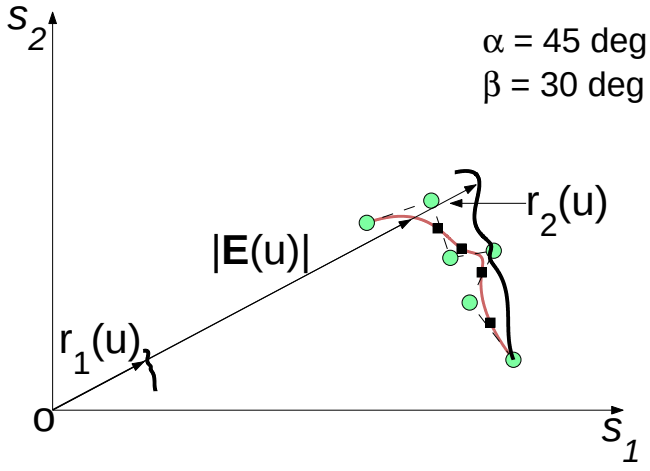

Figure 4.14: A constrained C^2 cubic B-spline interpolation of a set of points in s_1s_2 plane; $r_1(u) = |w\sin((\alpha-\beta)/2)|$, $r_2(u) = |w\sin((\alpha+\beta)/2)|$, $w^2 = Q_1^2(u) + Q_2^2(u) + Q_3^2(u) + Q_4^2(u)$.

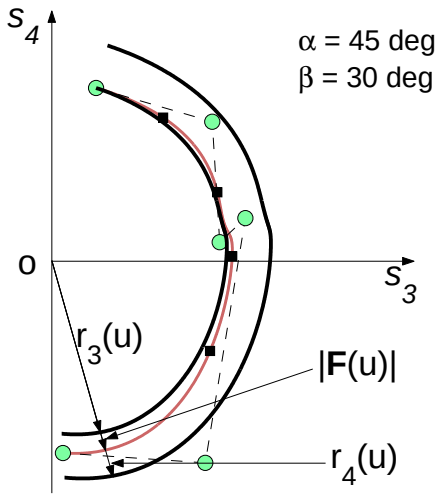

Figure 4.15: A constrained C^2 cubic B-spline interpolation of a set of points in s_3s_4 plane; $r_3(u) = |w\cos((\alpha+\beta)/2)|$, $r_4(u) = |w\cos((\alpha-\beta)/2)|$, $w^2 = Q_1^2(u) + Q_2^2(u) + Q_3^2(u) + Q_4^2(u)$.

Chapter 5

Constrained Motion Interpolation for Planar and Spherical 4R, 5R and 6R Closed Chains

5.1 Introduction

The purpose of this chapter is to study the synthesis of rational motions under kinematic constraints of planar and spherical 4R, 5R, and 6R closed chains. The spherical displacement of the coupler link is represented by a quaternion and the planar displacement is represented by a planar quaternion (see Bottema and Roth [8] and McCarthy [10] for quaternion representation of displacements). In this way, the problem of rational motion interpolation is transformed into that of rational curve interpolation, and the kinematic constraints of planar and spherical 6R closed chains are transformed into geometric constraints for the rational interpolation. Thus, given a series of coupler's positions in Cartesian space, the problem of synthesizing the smooth interpolating

rational motion of a planar and a spherical 6R closed chain is reduced to that of designing a smooth rational spline constrained to lie on the constraint manifolds of the planar or the spherical closed chain. To solve this problem, first a free-form smooth B-spline quaternion curve is used to interpolate the given positions. To ensure that the entire motion satisfies the kinematic constraints, an algorithm is developed that detect extreme positions on the rational motion that violates the kinematic constraints. These extreme positions are then modified so that they are in compliance with the kinematic constraints and are added to the list of positions to be interpolated. By repeating this process, one obtain a rational B-spline motion such that it fully satisfies the kinematic constraints of the planar 6R closed chain or the spherical 6R closed chain. As planar and spherical 5R, 4R closed chains can be obtained by holding one or two joints fixed, the above mentioned algorithm is also shown to be applicable to the problem of synthesizing rational motions for the planar and the spherical 5R, 4R closed chains.

The organization of this chapter is as follows. Section 5.2 reviews the idea of synthesis of free-form rational planar and spherical motions. Section 5.3 presents algorithms for the rational motion interpolation of planar 4R, 5R, and 6R closed chains. Section 5.4 shows algorithms for the rational motion interpolation of spherical 4R, 5R, and 6R closed chains.

5.2 Unconstrained Rational Planar and Spherical Motions

This section reviews the idea of synthesis of unconstrained (also known as, free-form) rational planar and spherical motions.

5.2.1 Unconstrained Rational Planar Motion

Planar quaternions have been used for developing unconstrained planar rational B-Spline motions (Wagner [60]).

Let $\mathbf{Z}_i,\ i = 0, ..., n$ be planar quaternions, then the following represents a Bézier curve in the space of planar quaternions:

$$\mathbf{Z}(u) = \sum_{i=0}^{n} B_i^n(u)\mathbf{Z}_i. \tag{5.1}$$

where $B_i^n(u)$ are the Bernstein polynomials.

Similarly, a B-Spline planar quaternion curve is given by:

$$\mathbf{Z}(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{Z}_i. \tag{5.2}$$

where $N_{i,p}(u)$ are pth-degree basis functions.

A representation for the rational Bézier motion and rational B-Spline motion in the Cartesian space can be obtained by substituting Eq. (5.1) and Eq. (5.2) into the homogeneous matrix $[A]$. From Eq. (2.10), it can be seen that if the Bézier or B-Spline curve $\mathbf{Z}(u)$ is expressed as a polynomial function of degree n, then the matrix $[A]$ represents a rational Bézier or B-Spline motion of degree $2n$.

5.2.2 Unconstrained Rational Spherical Motion

Ravani and Roth [61] considered the components of $\mathbf{q}$ as defining a point in a projective three-space, called the *Image Space* of spherical kinematics. The image space of spherical kinematics is a three-dimensional Cayley-Klein space with elliptic metric. (see Müller [55])

A polynomial curve in the Image Space corresponds to a rational spherical motion in the Cartesian space. By applying CAGD techniques for designing curves in the Image Space, we obtain rational motions in the Cartesian space.

Let $\mathbf{Q}_i,\ i = 0, ..., n$ be given quaternions, then the following represents a B-Spline quaternion curve in the space of quaternions:

$$\mathbf{Q}(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{Q}_i, \tag{5.3}$$

where $N_{i,p}(u)$ are pth-degree basis functions. See Farin [11], Hoschek and Lasser [14], and Piegl and Tiller [15] for details on the B-spline curves.

A representation for the rational B-Spline motion in the Cartesian space can be obtained by substituting the coordinates of $\mathbf{Q}(u)$ given by Eq. (5.3) into the homogeneous matrix $[A]$ (Eq. (2.5)). It is easy to see that if the B-Spline curve $\mathbf{Q}(u)$ is expressed as a polynomial function of degree p with parameter u treated as time, then the matrix $[A]$ represents a rational B-Spline motion of degree $2p$.

5.3 Rational Motion Interpolation of Planar 4R, 5R and 6R Closed Chains

In this section, we present algorithms for synthesizing C^2 continuous piecewise ration motions of planar 4R, 5R, and 6R close chains under the kinematic constraints derived in chapter 3.

5.3.1 Rational Motion Interpolation of Planar 6R Closed Chain

In this subsection, we present an algorithm for the following constrained motion interpolation problem:

Given: A set of positions of the coupler link of a planar 6R closed chain in its workspace as well as the parameter sequence $u_i,\ i = 1, ..., n$ for the given positions.

Find: A rational B-spline motion of the coupler link that interpolates through the given positions at the parameter values subject to the kinematic constraints of the planar 6R closed chain.

With the use of planar quaternions, this problem is transformed into that of interpolating a set of points in the space of planar quaternions subject to the kinematic constraints (3.49) and (3.50). In this section, we use cubic B-spline interpolation algorithm for points interpolation. This algorithm can be found in CAGD texts (see for example, Farin [11], Hoschek and Lasser [14], and Piegl and Tiller [15]).

We first interpolate the set of planar quaternions without any kinematic

constraint. We then try to find the extreme point on the B-spline planar quaternion curve that has the maximum deviation from the constraint manifold of the 6R chain. This extreme point is then "projected" into the constraint manifold. This results in a new point within the constraint manifold. This new point is then added to the list of points (or planar quaternions) to be interpolated. We repeat this process until there is no extreme point on the interpolating curve that violates the kinematic constraints.

Algorithm 5.3.1

1. Convert given positions into planar quaternions $\mathbf{Y}_i = (Y_{i1}, Y_{i2}, Y_{i3}, Y_{i4})$ using Eq. (3.45), if given positions are in joint angles.

2. Interpolate these planar quaternions $\mathbf{Y}_i$ with a cubic B-spline curve in the space of planar quaternions.

3. Find extreme points on the interpolating curve which violate the kinematic constraint Eq. (3.49) or Eq. (3.50).

4. If no extreme points are found, then stop and the resulting motion satisfies all the kinematic constraints of the planar 6R closed chain. If there are extreme points that violate the kinematic constraints then add new points (which satisfy the kinematic constraints) as well as the associated parameter values to the list of points to be interpolated and go to step 2.

To find extreme points on the interpolating curve we rewrite Eq. (3.37) and Eq. (3.38) as follows:

$$\mathbf{F}_1(u) = \frac{(Y_1(u) + \tau Y_4(u))^2 + (Y_2(u) - \sigma Y_3(u))^2}{Y_3^2(u) + Y_4^2(u)}, \tag{5.4}$$

$$\mathbf{F}_2(u) = \frac{(Y_1(u) - \tau Y_4(u))^2 + (Y_2(u) + \sigma Y_3(u))^2}{Y_3^2(u) + Y_4^2(u)}. \tag{5.5}$$

The maximum and minimum values of $\mathbf{F}_1(u)$ and $\mathbf{F}_2(u)$ can be obtained from

$$\frac{d\mathbf{F}_1(u)}{du} = 0, \quad \frac{d\mathbf{F}_2(u)}{du} = 0. \tag{5.6}$$

Then the extreme points are those points which have maximum or minimum values of $\mathbf{F}_1(u)$ or $\mathbf{F}_2(u)$, and reside outside of the range governed by Eqs. (3.49) and (3.50).

We solve the following modified kinematic constraint equations to find new points which satisfy the kinematic constraints in the step 4 of the algorithm:

$$(Y_1 + \tau Y_4)^2 + (Y_2 - \sigma Y_3)^2 = r_1^2(Y_3^2 + Y_4^2), \tag{5.7}$$

$$(Y_1 - \tau Y_4)^2 + (Y_2 + \sigma Y_3)^2 = r_2^2(Y_3^2 + Y_4^2). \tag{5.8}$$

In general, one can project the extreme point onto the constraint manifold using the concept of normal distance in the image space introduced by Ravani and Roth [45]. In this chapter, however, we use a much simpler approach in order to obtain a more efficient algorithm. We use the values of Y_3, Y_4 of the extreme point and then obtain new values for Y_1, Y_2 by solving (5.7) and (5.8) simultaneously. Kinematically, this means that we keep the same orientation of the coupler link associated with the extreme point but shift the position

of the coupler link so that the kinematic constraints of the coupler link are satisfied.

In order to make sure that the new point stays inside of the constraint manifold, we select r_1, r_2 by:

$$r_1 = \begin{cases} |a_1 - b_1|/2 + \delta_1, & \text{if } \mathbf{F}_1(u^*) < (a_1 - b_1)^2/4; \\ (a_1 + b_1)/2 - \delta_1, & \text{if } \mathbf{F}_1(u^*) > (a_1 + b_1)^2/4; \\ \mathbf{F}_1(u^*), & \text{otherwise.} \end{cases} \tag{5.9}$$

and

$$r_2 = \begin{cases} |a_2 - b_2|/2 + \delta_2, & \text{if } \mathbf{F}_2(u^*) < (a_2 - b_2)^2/4; \\ (a_2 + b_2)/2 - \delta_2, & \text{if } \mathbf{F}_2(u^*) > (a_2 + b_2)^2/4; \\ \mathbf{F}_2(u^*), & \text{otherwise.} \end{cases} \tag{5.10}$$

where u^* is the parameter value corresponding to the extreme point. The parameters, δ_1 and δ_2 define how far inside the new point is supposed to be inside of the constraint manifold. When they equal zero, the new point lies on the boundary of the constraint manifold.

From Eqs. (5.7) and (5.8) we can see that they represent two circles when Y_3 and Y_4 are held constant. In our algorithm we choose the one of two intersection points of two circles which is closer to the extreme point as new point.

We now give an example to demonstrate the algorithm. A set of positions of the coupler link of a planar 6R closed chain are given in Table 5.1 and the corresponding planar quaternions are given in Table 5.2. The kinematic constraints of the given planar 6R closed chain is shown in Table 5.3. The results of the advocated algorithm are shown in Figures 5.1, 5.2 and 5.3. For the ease of visualization, we have projected all objects from planar quaternion space

i	1	2	3	4	5
θ_i(deg)	63.3	68.5	97.1	169.7	-129.1
ϕ_i(deg)	-84.6	-74.8	-16.6	-108.8	-136.9
ψ_i(deg)	21.3	36.3	-95.5	-45.9	-63.0
u_i	0.0	2.0	5.0	7.0	10.0

Table 5.1: **Input joint angles for a planar 6R closed chain** ($a_1 = 1$, $b_1 = 3$, $a_2 = 4$, $b_2 = 2$).

onto $Y_4 = 1$ hyperplane. We have rotated and zoomed Figure 5.1 to obtain Figures 5.2 and 5.3, which allow illustration of the results more clearly. While the green and red hyperboloids represent the kinematic constraints Eqs. (3.49) and (3.50), the given positions are marked by pink squares, the extreme points are shown as red star points, and the new points that satisfy the kinematic constraints are denoted as yellow squares. The initial curve for unconstrained interpolation is shown in black and the final curve for constrained interpolation is shown in blue. We set the values $\delta_1 = \delta_2 = 0.1$ in Eqs. (5.9) and (5.10). From Figure 5.2 we find that one portion of the initial black curve contains one extreme point that violates the kinematic constraint Eq. (3.49) ($\mathbf{F}_1 \geq 1$). Figure 5.3 shows us that another part of the initial curve, which also has one extreme point that violates the kinematic constraint Eq. (3.49) ($\mathbf{F}_1 \leq 4$). The blue curve in each of the two figures represents the new planar quaternion curve that satisfies the kinematic constraints of given planar 6R closed chain.

i	$\mathbf{Z}_i = (Z_{i1}, Z_{i2}, Z_{i3}, Z_{i4})$
1	$(1.0224, -0.0970, 0.0000, 1.0000)$
2	$(1.1153, 0.4791, 0.2588, 0.9659)$
3	$(-0.6683, 1.6696, -0.1305, 0.9914)$
4	$(-0.1761, 1.6698, 0.1305, 0.9914)$
5	$(-0.6875, 1.8228, 0.2676, 0.9635)$

Table 5.2: **Planar quaternions of given positions of the coupler link of planar 6R closed chain.**

i	Kinematic constraints $\mathbf{F}_i$
1	$1.0 \leq \mathbf{F}_1 \leq 4.0$
2	$1.0 \leq \mathbf{F}_2 \leq 9.0$

Table 5.3: **Kinematic constraints of given planar 6R closed chain.**

5.3.2 Rational Motion Interpolation of Planar 4R Closed Chain

This section shows that the algorithm described in the previous section may be applied for the rational motion interpolation under the kinematic constraints of the planar 4R closed chain.

To use the algorithm mentioned in the section 5.3.1, we modify the kinematic constraints Eqs. (3.35) and (3.36) to be:

$$\left(\frac{a_1 - \delta_1}{2}\right)^2 \leq \mathbf{F}_1(Y_1, Y_2, Y_3, Y_4) \leq \left(\frac{a_1 + \delta_1}{2}\right)^2, \tag{5.11}$$

and

$$\left(\frac{a_2 - \delta_2}{2}\right)^2 \leq \mathbf{F}_2(Y_1, Y_2, Y_3, Y_4) \leq \left(\frac{a_2 + \delta_2}{2}\right)^2. \tag{5.12}$$

where δ_1 and δ_2 are allowable tolerances indicating how closely the kinematic constraints (3.35) and (3.36) of a 4R chain should be approximated.

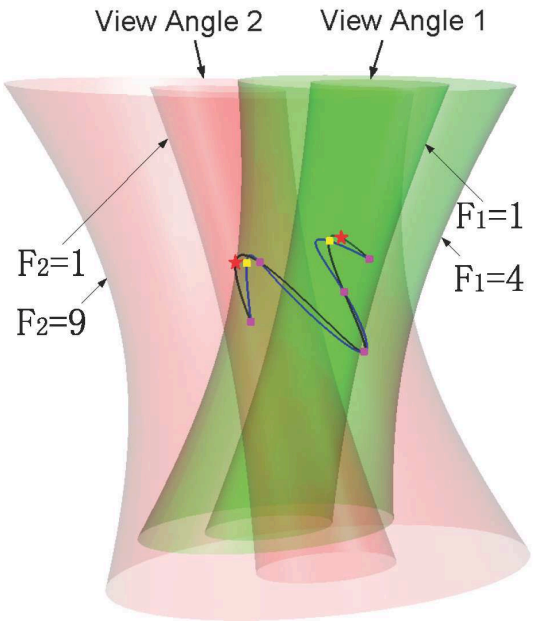

Figure 5.1: A constrained interpolation of given points.

With the modified kinematic constraints Eqs. (5.11) and (5.12), we can directly use the same algorithm described in the section 5.3.1 except that when we find new points in the step 4 of the algorithm the r_1 and r_2 of Eqns (5.9) and (5.10) are given by:

$$r_1 = \frac{a_1}{2}, \quad r_2 = \frac{a_2}{2}. \tag{5.13}$$

5.3.3 Rational Motion Interpolation of Planar 5R Closed Chain

For a planar 5R closed chain, we can use the same algorithm as for the planar 6R closed chain. We need to replace the first kinematic constraint Eq. (3.43) with Eq. (5.11). The value of r_1 is given by Eq. (5.13) in the step 4 of the **Algorithm 5.3.1**.

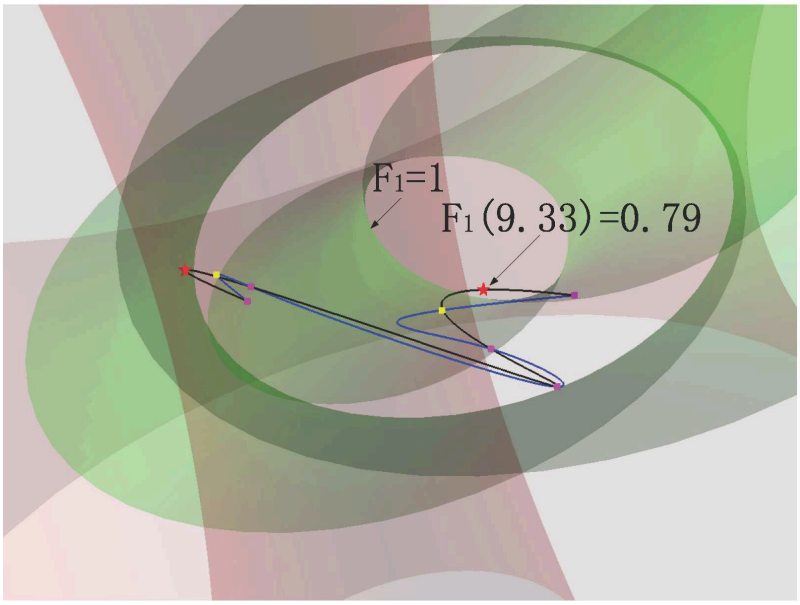

Figure 5.2: Zoomed image from view angle 1 of Figure 5.1. One extreme point violates the kinematic constraint $1.0 \leq \mathbf{F}_1 \leq 4.0$).

5.4 Rational Motion Interpolation of Spherical 4R, 5R and 6R Closed Chains

In this section, we first present an algorithm for synthesizing C^2 continuous piecewise rational motions of spherical 6R closed chain under the kinematic constraints derived in chapter 3. Next we will show that the same algorithm can be applied for the synthesis of interpolating rational motion of spherical 4R and 5R closed chains with the kinematic constraints being satisfied only approximately, however within a user-defined tolerance.

The positions of the coupler of a spherical 6R closed chain can be given by either Cartesian coordinates $(s_{xi}, s_{yi}, s_{zi}, \theta_i)$ or joint coordinates $(\theta_i, \phi_i, \psi_i)$. Eq. (2.1) can be used to convert Cartesian coordinates into quaternions, while

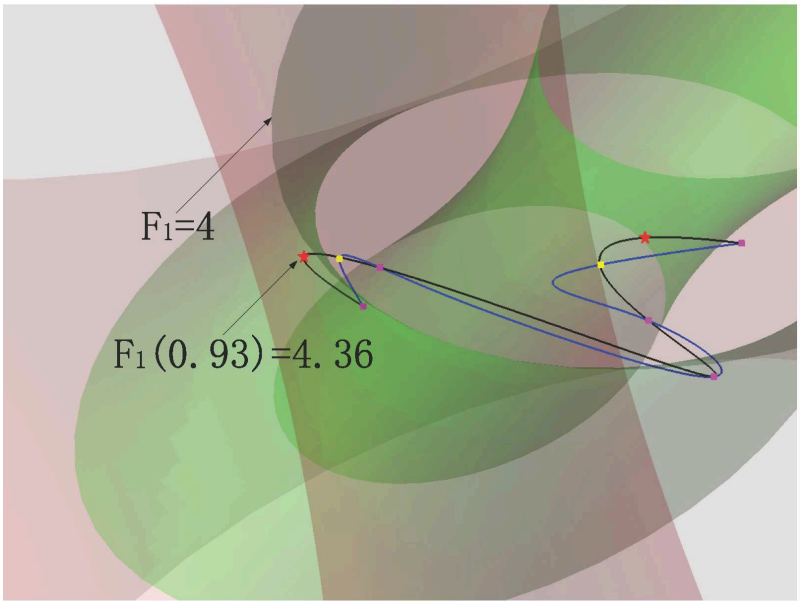

Figure 5.3: Zoomed image from view angle 2 of Figure 5.1. One extreme point violates the kinematic constraint $1.0 \leq \mathbf{F}_1 \leq 4.0$).

Eq. (3.84) can be used to convert joint coordinates into quaternions. If the given positions of the coupler are not connectable with a continuous trajectory, then the mechanism has to be disassembled and re-assembled to interpolate the positions. This is commonly known as "assembly mode defect". Recently, Schröcker and Husty [62] and Schröker et al. [63] presented fast numerical algorithms for determining if two task positions of planar and spherical four-bar mechanisms suffer from this defect. The algorithm presented below assumes that the given positions are on the same branch.

5.4.1 Rational Motion Interpolation for Spherical 6R Closed Chain

In this subsection, we present an algorithm for the following constrained motion interpolation problem:

Given: A set of positions of the coupler link of a spherical 6R closed chain in its workspace, the corresponding parameter values u_i $(i = 1 \ldots n)$, angular spans of the links given by α_i, β_i $(i = 1, 2)$, and the angular distance between the two moving pivots and between the two fixed pivots given by η and γ, respectively.

Find: A rational motion of the coupler link that interpolates the given positions at the parameter values subject to the kinematic constraints of the spherical 6R closed chain.

In the ensuing discussion over a general description of the algorithm, we assume that the given positions of the coupler link are represented by quaternion coordinates given by $\mathbf{Y}_i = (Y_{i1}, Y_{i2}, Y_{i3}, Y_{i4});\ i = 1 \ldots n$. These quaternions map into points in the space of quaternions.

Detection of Kinematic Constraints Violation

Finding a motion that satisfies the requirements stated earlier in this section begins with the construction of a C^2 cubic B-spline image curve [1] $\mathbf{Y}(u)$ that interpolates $\mathbf{Y}_i$ at parameter values u_i. Such a curve does not automatically satisfy the kinematic constraints given by Eqs. (3.87) and (3.88). Specifically,

[1] Designing a C^2 B-spline curve is a standard scheme in CAGD (see Farin [11], Hoschek and Lasser [14], and Piegl and Tiller [15]).

kinematic constraints are considered violated at those points of the synthesized image curve $\mathbf{Y}(u)$, where the following functions (rewritten from Eqs. (3.77) and (3.78)) do not satisfy the inequality given in Eqs. (3.87) and (3.88):

$$F_1(u) = \frac{(Y_1(u)\sin\tau - Y_4(u)\cos\tau)^2 + (Y_2(u)\sin\sigma + Y_3(u)\cos\sigma)^2}{Y_1^2(u) + Y_2^2(u) + Y_3^2(u) + Y_4^2(u)}, \quad (5.14)$$

$$F_2(u) = \frac{(Y_1(u)\sin\tau + Y_4(u)\cos\tau)^2 + (Y_2(u)\sin\sigma - Y_3(u)\cos\sigma)^2}{Y_1^2(u) + Y_2^2(u) + Y_3^2(u) + Y_4^2(u)}. \quad (5.15)$$

To detect violations of the kinematic constraints, extrema of functions $F_1(u)$ and $F_2(u)$ are calculated using the following equations:

$$\frac{dF_1(u)}{du} = 0, \quad \frac{dF_2(u)}{du} = 0. \quad (5.16)$$

If an extremum falls outside the limits in any of the inequalities (Eqs. (3.87) and (3.88)), the constraints are considered violated. We call a point on the image curve corresponding to an extremum of $F_1(u)$ or $F_2(u)$ an extreme point. A C^2 cubic B-spline curve can be easily converted into a piecewise Bézier form using standard methods (see Farin [11]), where each Bézier segment has an algebraic form. The piecewise Bézier form allows easy evaluation of Eq. (5.16) for each of the L Bézier segments, $\mathbf{Y}_i(u)$; $i = 0, 1, \ldots, L-1$, that make up the B-spline curve $\mathbf{Y}(u)$.

Modifying the Curve to Satisfy Kinematic Constraints

By restricting to operate in the image space, we have transformed the kinematic constraints into geometric constraints. Thus, we can take a purely geometric approach to modify the synthesized image curve in such a way that

the kinematic constraints are satisfied. Our approach is to replace the extreme point that violates the kinematic constraints with a new point that satisfies kinematic constraints, thereby dragging the curve to the inside of the constraint manifold. To ensure that the image curve changes minimally, the new point should be closest possible to the extreme point. This approach is realized in a two step iterative process: first, a new point is generated that satisfies the kinematic constraints and is minimally away from the extreme point, then this new point is added (at the parameter value corresponding to the extreme point) to the initial set of given points and a new C^2 cubic B-spline image curve is generated that interpolates the new point as well. If the new image curve detects any further violation of the kinematic constraints as outlined in the previous subsection, this two-step process is repeated until no extreme points that violate the kinematic constraints are found. This process may yield many new points before the kinematic constraints are satisfied completely. A visual interpretation of this approach is that the image curve is repeatedly "bent-in" or "bent-out" in the vicinity of the constraints violating extreme points in its image space, while still interpolating through the given positions, until it fits in the constraint manifold defined by the kinematic constraints.

Generating New Points

Since we require the new points to be minimally away from the extreme points, the issue of finding new points can be seen as a normal distance minimization problem in the image space subject to certain constraints. Ravani and Roth [45] proposed a general algebraic method for approximate normal dis-

tance calculation between the image curve and a given position in the image space. Later on, Bodduluri and McCarthy [46] used their method for finite position synthesis of a spherical four-bar motion. In what follows, we outline this method and conform the issue of finding new points to the framework of this method for solution.

We first rewrite the kinematic constraint equations from Eqs. (3.77), (3.78), and Eqs. (3.87), (3.88) as follows:

$$(Y_1 \sin\tau - Y_4 \cos\tau)^2 + (Y_2 \sin\sigma + Y_3 \cos\sigma)^2 = r_1^2 \sum_{i=1}^{4} Y_i^2, \qquad (5.17)$$

$$(Y_1 \sin\tau + Y_4 \cos\tau)^2 + (Y_2 \sin\sigma - Y_3 \cos\sigma)^2 = r_2^2 \sum_{i=1}^{4} Y_i^2. \qquad (5.18)$$

Here, r_1 and r_2 are variables which should satisfy the following inequalities:

$$\begin{aligned} |\cos(\frac{\alpha_1+\beta_1}{2})| \le r_1 \le |\cos(\frac{\alpha_1-\beta_1}{2})| \\ |\cos(\frac{\alpha_2+\beta_2}{2})| \le r_2 \le |\cos(\frac{\alpha_2-\beta_2}{2})| \end{aligned} \qquad (5.19)$$

The new point should satisfy Eqs. (5.17), (5.18) as well as the following:

$$Y_1^2 + Y_2^2 + Y_3^2 + Y_4^2 = w^2, \qquad (5.20)$$

where w is a non-zero scalar (or, weight) for the quaternion coordinates of the new point. Here, without any loss of generality, we choose $w = 1$, since as we showed before in section 5.2.2, the coordinates of $\mathbf{Y}$ are homogeneous coordinates.

In the context of spherical four-bar mechanism synthesis, the method used by Bodduluri and McCarthy [46] solves the design problem of determining an

image curve that passes through or near a set of given points. The image curve is algebraically given by intersection of two quadric hypersurfaces (constraint surfaces) and is expressed as functions of the design parameters, such as link lengths and the pivot locations. They seek the values of design parameters which reduce the error between the image curve and the given points. The error at each given position is defined as the normal distance to the image curve. They define total error as the sum of squares of each position error. The method is approximate in the sense that the constraint surfaces are approximated by their tangent hyperplane in the vicinity of the desired position. We use the same approach to find a new point $\mathbf{Y}$ that satisfies the constraints given by Eqs. (5.17), (5.18), and (5.20) and is also at the minimum distance from the extreme point ($\mathbf{Y}^*$) that violates the kinematic constraints. In our case, we define $\mathbf{e} = \mathbf{Y} - \mathbf{Y}^*$ as the normal error vector at the position $\mathbf{Y}^*$. Minimization of the square of the magnitude of $\mathbf{e}$ subject to Eq. (5.19) would yield r_1 and r_2. This in turn would give the new point $\mathbf{Y} = \mathbf{e}^* + \mathbf{Y}^*$, where $\mathbf{e}^*$ is the optimized normal distance. This new point $\mathbf{Y}$ may not satisfy the kinematic constraints because the constraint surfaces are only approximated in this approach. In that case, this new point $\mathbf{Y}$ is set as the new extreme point $\mathbf{Y}^*$ and the process described above is repeated.

The optimization procedure outlined above involves the tangent hyperplane approximation to the hypersurfaces. For that, we first rewrite Eqs. (5.17), (5.18), and (5.20) as follows:

$$H_1(\mathbf{Y}) : (Y_1 \sin\tau - Y_4 \cos\tau)^2 + (Y_2 \sin\sigma + Y_3 \cos\sigma)^2 - r_1^2 = h_1, \qquad (5.21)$$

$$H_2(\mathbf{Y}) : (Y_1 \sin\tau + Y_4 \cos\tau)^2 + (Y_2 \sin\sigma - Y_3 \cos\sigma)^2 - r_2^2 = h_2, \tag{5.22}$$

$$G(\mathbf{Y}) : Y_1^2 + Y_2^2 + Y_3^2 + Y_4^2 - 1 = g, \tag{5.23}$$

where, $h_1 = h_2 = g = 0$.

These surfaces can be approximated by the tangent hyperplanes at a point $\mathbf{Y}^* = (Y_1^*, Y_2^*, Y_3^*, Y_4^*)$ by following:

$$\begin{aligned} 0 &= H_1(\mathbf{Y}^*) + \sum_{i=1}^{4} \frac{\partial H_1(\mathbf{Y}^*)}{\partial Y_i} \Delta Y_i, \\ 0 &= H_2(\mathbf{Y}^*) + \sum_{i=1}^{4} \frac{\partial H_2(\mathbf{Y}^*)}{\partial Y_i} \Delta Y_i, \\ 0 &= \sum_{i=1}^{4} 2Y_i^* \Delta Y_i. \end{aligned} \tag{5.24}$$

These equations can be assembled as follows:

$$\begin{bmatrix} \frac{\partial H_1}{\partial Y_1} & \frac{\partial H_1}{\partial Y_2} & \frac{\partial H_1}{\partial Y_3} & \frac{\partial H_1}{\partial Y_4} \\ \frac{\partial H_2}{\partial Y_1} & \frac{\partial H_2}{\partial Y_2} & \frac{\partial H_2}{\partial Y_3} & \frac{\partial H_2}{\partial Y_4} \\ 2Y_1^* & 2Y_2^* & 2Y_3^* & 2Y_4^* \end{bmatrix} \begin{bmatrix} \Delta Y_1 \\ \Delta Y_2 \\ \Delta Y_3 \\ \Delta Y_4 \end{bmatrix} = \begin{bmatrix} -H_1 \\ -H_2 \\ 0 \end{bmatrix}. \tag{5.25}$$

The above can also be written as

$$[J]\mathbf{e} = \mathbf{v}. \tag{5.26}$$

Bodduluri and McCarthy [46] solve for the normal error vector $\mathbf{e}$ by minimizing the Lagrangian function given as follows:

$$L(\mathbf{e}, \mathbf{A}) = \mathbf{e}^T\mathbf{e} + \mathbf{A}^T([J]\mathbf{e} - \mathbf{v}), \tag{5.27}$$

where $\mathbf{A} = (A_1, A_2, A_3)$ is a vector of Lagrange multipliers, $\mathbf{e} = (Y_1 - Y_1^*, Y_2 - Y_2^*, Y_3 - Y_3^*, Y_4 - Y_4^*)$, $\mathbf{v} = (-H_1(\mathbf{Y}^*), -H_2(\mathbf{Y}^*), 0)$. Putting the condition for a minimum as

$$\frac{\partial L}{\partial e_i} = 0, \qquad i = 1, 2, 3, 4 \tag{5.28}$$

where (e_1, e_2, e_3, e_4) are the coordinates of the vector $\mathbf{e}$ and assembling the solution equations, we obtain

$$2\mathbf{e}^T + \mathbf{A}^T[J] = 0. \tag{5.29}$$

Thus, if $\mathbf{e}^*$ designates the solution to the error vector (or, the normal distance) then it should satisfy the equations

$$\begin{aligned} [J]\mathbf{e}^* &= \mathbf{v}, \\ 2\mathbf{e}^* + [J]^T\mathbf{A} &= 0. \end{aligned} \tag{5.30}$$

Equation (5.30) gives an explicit formula for the solution error vector $\mathbf{e}^*$ in terms of variables r_1 and r_2 (see Bodduluri and McCarthy [46]):

$$\mathbf{e}^* = [J]^T([J][J]^T)^{-1}\mathbf{v}. \tag{5.31}$$

Thus, we can determine the variables r_1 and r_2 by optimizing the function

$$E(\mathbf{Y}, r_1, r_2) = (\mathbf{e}^*)^T\mathbf{e}^*, \tag{5.32}$$

subject to constraints given by Eq. (5.19).

With normal error vector $\mathbf{e}^*$ known, the new point is given by

$$\mathbf{Y} = \mathbf{e}^* + \mathbf{Y}^*. \tag{5.33}$$

Since the constraint surfaces have been approximated by their tangent hyperplanes in the vicinity of the extreme point $\mathbf{Y}^*$, this new point $\mathbf{Y}$ may or may not lie inside the constraint solids given by Eqs. (3.87) and (3.88). If the new point does not satisfy the constraints, the newly obtained point $\mathbf{Y}$ is set as a new $\mathbf{Y}^*$ and the procedure described above is repeated from Eq. (5.24) until a new point $\mathbf{Y}$ is obtained that satisfies the kinematic constraints. This new point not only satisfies the kinematic constraints but is also at the shortest distance from the original extreme point $\mathbf{Y}^*$. With this new point added to the set of initial points in the quaternion space, a new C^2 B-spline is interpolated to the points. If the new image curve detects any further violation of the kinematic constraints, the optimization process is repeated until no further extreme points that violate the kinematic constraints are found. This process at the end gives an interpolating motion that satisfies the kinematic constraints. Now, we present the algorithm:

Algorithm 5.4.1

1. Convert given positions of the coupler link into unit quaternions (points in image space) $\mathbf{Y}_i = (Y_{i1}, Y_{i2}, Y_{i3}, Y_{i4})$ using either Eq. (2.1) or Eq. (3.84).

2. Current list of points to be interpolated = given points $(\mathbf{Y}_i; i = 1 \ldots n)$

3. Construct a C^2 cubic B-spline curve $\mathbf{Y}(u)$ that interpolates current list of points.

4. (a) Evaluate the extrema of $F_1(u)$ and $F_2(u)$ using Eq. (5.16). Say, the

extrema are found at $u = u_1^*$ and at $u = u_2^*$ for $F_1(u)$ and $F_2(u)$, respectively. Designate extreme points on the curve as $\mathbf{Y}_1^*$ and $\mathbf{Y}_2^*$.

(b) Check if $\mathbf{Y}_1^*$ and $\mathbf{Y}_2^*$ satisfy the kinematic constraints (Eqs. (3.87) and (3.88)).

(c) If yes, the curve is constrained; continue to Step 5.

(d) else, find new points $\mathbf{Y}(u_1^*)$ and $\mathbf{Y}(u_2^*)$ (Eqs. (5.24) – (5.33)).

(e) Check if the new points $\mathbf{Y}(u_1^*)$ and $\mathbf{Y}(u_2^*)$ satisfy kinematic constraints (Eqs. (3.87) and (3.88)).

i. If yes, continue to next Sub-step (f)

ii. else, set $\mathbf{Y}_1^* = \mathbf{Y}(u_1^*)$ and $\mathbf{Y}_2^* = \mathbf{Y}(u_2^*)$ and repeat from Sub-step (d).

(f) Add $\mathbf{Y}(u_1^*)$ and $\mathbf{Y}(u_2^*)$ to the current list of points to be interpolated and go to Step 3.

5. The image curve $\mathbf{Y}(u)$ defines a C^2 interpolating piecewise rational motion of degree 6 after the substitution into Eq. (2.5).

We have observed that this algorithm always converges as long as the interpolating points are on the same branch. In this algorithm, the B-spline curve is generated using a global interpolation scheme (Piegl and Tiller [15]). In this scheme, although moving one of the interpolating points changes the curve globally, the change in the curve diminishes away from the modification point. Next we show via an example that the algorithm converges fast as well.

i	$\mathbf{Y}_i = (Y_{i1}, Y_{i2}, Y_{i3}, Y_{i4})$	u_i
1	$(-0.2636, 0, 0, 0.9646)$	0.0
2	$(-0.0039, -0.2949, -0.0814, 0.9520)$	2.0
3	$(-0.0110, -0.5860, 0.0150, 0.8101)$	5.0
4	$(-0.4910, -0.4170, -0.0005, 0.7649)$	7.0
5	$(-0.5620, 0, 0, 0.8271)$	10.0

Table 5.4: **Quaternion coordinates of the given positions of the coupler link of spherical 6R closed chain**($\alpha_1 = \pi/4$, $\beta_1 = \pi/6$, $\alpha_2 = \pi/3$, $\beta_2 = \pi/6$, $\eta = \pi/3$, $\gamma = \pi/2$)

i	Kinematic constraints F_i
1	$0.63 \leq F_1 \leq 0.98$
2	$0.5 \leq F_2 \leq 0.93$

Table 5.5: **Kinematic constraints of given spherical 6R closed chain.**

Table 5.4 gives the quaternion coordinates $(Y_{i1}, Y_{i2}, Y_{i3}, Y_{i4})$ for five positions of the coupler link of a spherical 6R closed chain along with their parameter values. The table also gives the link lengths and the angular distance between moving and the fixed pivots. The range of the inequality in the kinematic constraints given by Eqs. (3.87) and (3.88) are shown in Table 5.5.

To visualize the constraint solids in three-dimensional Euclidean space, we recast the constraint equations (Eqs. (3.87) and (3.88)) in terms of Rodrigues parameters (see Bottema and Roth [8]) given by $(Y_1/Y_4, Y_2/Y_4, Y_3/Y_4)$. Geometrically, this is equivalent to observing the intersection of constraint solids with hyperplane $Y_4 = 1$. Figure 5.4 shows such an intersection of the constraint surfaces of the four dimensional solids described by Eqs. (3.87) and (3.88) with the hyperplane $Y_4 = 1$. Various surfaces are the indicated by the

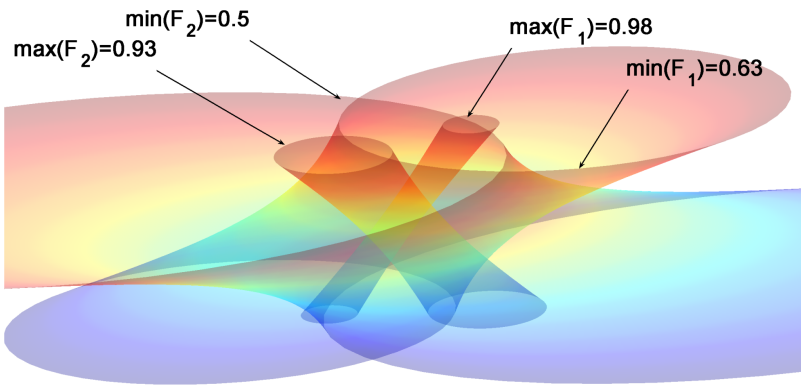

Figure 5.4: Kinematic constraint surfaces obtained by intersection with the hyperplane $Y_4 = 1$

limits of the inequalities (Table 5.5). These surfaces are hyperboloid of one sheet and we note here that contrary to what the reader might expect, the inner hyperboloids indicate the upper and the outer ones indicate the lower range of the kinematic constraints inequality (Eqs. (3.87) and (3.88)). The volume bounded by the surfaces is the valid region for the image curve to remain in for the constraints to be satisfied.

Figure 5.5 shows the kinematic constraint surfaces and the unconstrained interpolation of the given positions from a different angle than shown in Figure 5.4. The extreme points that violate the kinematic constraints are shown by the star points ('$\star$'). Given positions interpolated by the image curve are indicated by '■'. This figure shows that there are two extreme points, each of which violate one of the kinematic constraints. One of the extreme point is seen to be at $u^* = 0.92$, where $F_1 = 1.0$. This point is outside the bounds

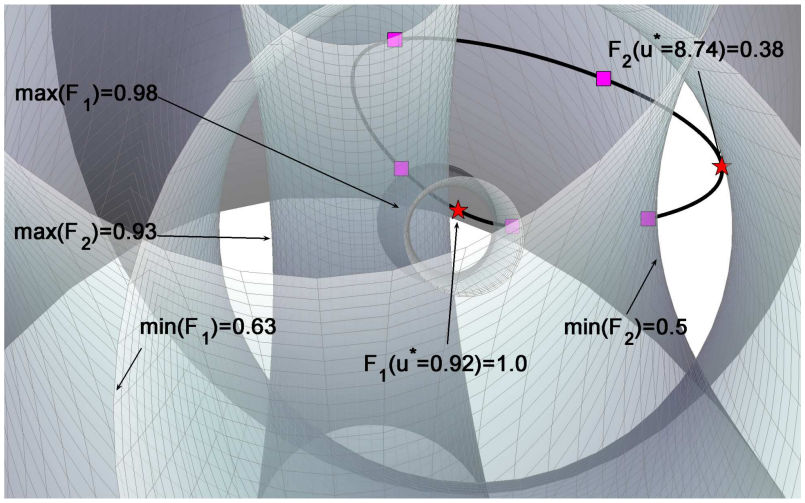

Figure 5.5: Unconstrained interpolation and the kinematic constraint surfaces obtained by intersection with the hyperplane $Y_4 = 1$. The image curve is shown as a continuous curve, the two extreme points that violate kinematic constraints as '$\star$', and the given positions as '$\blacksquare$'.

of $F_1(u)$: $0.63 \leq F_1 \leq 0.98$. It is not difficult to see from the figure that this extreme point does not violate the other kinematic constraint given by $0.5 \leq F_2 \leq 0.93$. This point can be clearly seen between the two kinematic constraint surfaces given by the limits of $F_2(u)$. Similarly, the other extreme point is seen to be at $u^* = 8.74$, where $F_2 = 0.38$. This point violates the lower limit of the inequality $0.5 \leq F_2$. It can also be seen from the figure that this extreme point does not violate the other kinematic constraint.

Figures 5.6 and 5.7 focus on the violation of different constraints and visually demonstrate the application of the algorithm in constraining the image curve. Figure 5.6, which is obtained by rotating and zooming in on the Figure 5.5 focuses on the kinematic constraint $0.5 \leq F_2 \leq 0.93$ and the extreme point that violates this constraint. It shows the unconstrained curve with one

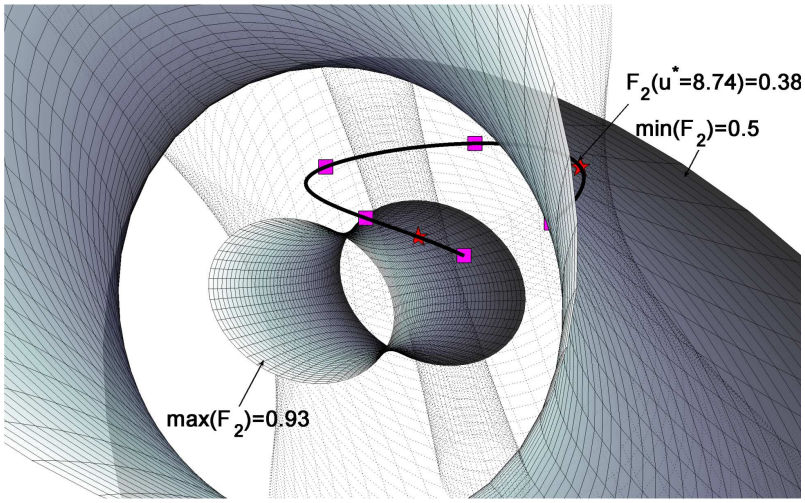

Figure 5.6: A zoomed-in and rotated view of the Figure 5.5: one extreme point with $F_2(8.74) = 0.38$ violates the kinematic constraint: $0.5 \leq F_2$. Surfaces given by the limits of the other constraint are shown in light broken lines.

of the extreme points that violates the constraint $0.5 \leq F_2$. This point is clearly seen to be outside the volume defined by the constraint surfaces. Surfaces given by the limits of the other constraint $F_1(u)$ (Eq. (3.87)) that is not violated by this extreme point are shown in light broken lines. It should also be noted that the other unlabeled extreme point does not violate the constraint $0.5 \leq F_2 \leq 0.93$.

Similarly, Figure 5.7 shows from another angle the kinematic constraint surfaces given by the limits of $F_1(u)$ and the unconstrained curve. This figure shows the other extreme point that violates the upper limit ($F_1 \leq 0.98$) of the kinematic constraints given by the inequality in Eq. (3.87). Also shown are the constraint surfaces given by the limits of $F_2(u)$ in light broken lines.

Application of the algorithm presented earlier is illustrated in the Fig-

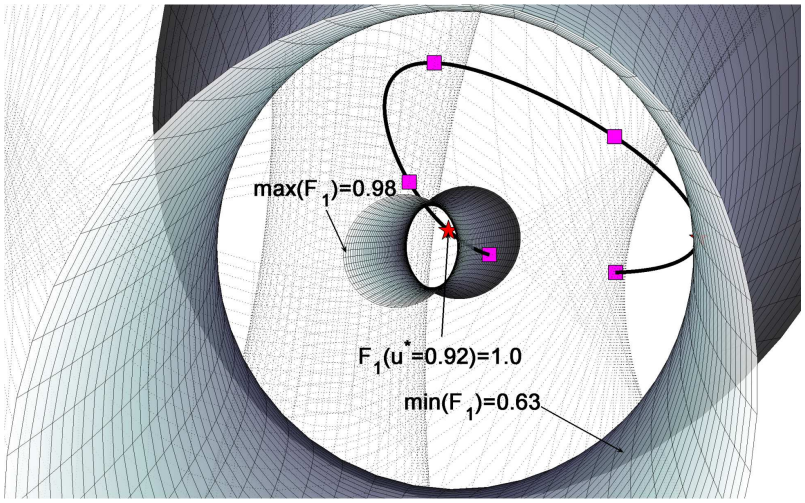

Figure 5.7: A zoomed-in and rotated view of the Figure 5.5: one extreme point with $F_1(0.92) = 1.0$ violates the kinematic constraint: $F_1 \leq 0.98$. Surfaces given by the limits of the other constraint are shown in light broken lines.

ure 5.8 and 5.9, where in two iterations, the algorithm produces a constrained curve. On a Pentium 4 2.0 GHz system with 512 MB of primary memory, the algorithm ran for less than 5 seconds. Two new points that replace the extreme points are added in the process, which are indicated by 'o'. It might seem from Figure 5.8 and 5.9 that the two new points added to satisfy the limits on $F_2(u)$ and $F_1(u)$ are not minimally away from their corresponding extreme point, respectively. However, this is not true. A different view of the locations of the two new points verifies this assertion visually in Figure 5.8 and Figure 5.9, e.g., the new point labeled in Figure 5.8 is seen to be very close to the outer constraint surface shown in light broken lines in the Figure 5.9.

The final interpolating curve satisfies all the constraints, which in the Cartesian space translates into a C^2 continuous B-spline rational motion of

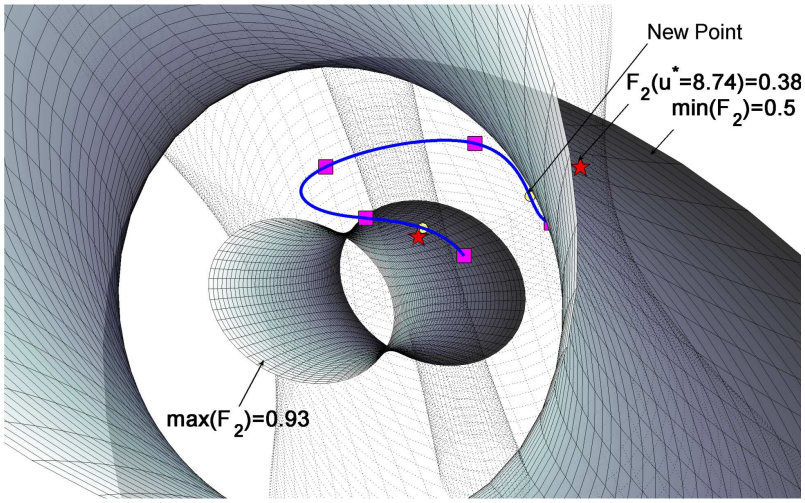

Figure 5.8: Constrained interpolation (compare with Figure 5.6): The labeled new point ('o') is the point that replaces the labeled extreme point.

the spherical 6R closed chain.

5.4.2 Rational Motion Interpolation of Spherical 4R And 5R Closed Chains

Constraint equations of spherical 4R and 5R closed chains can be seen as a special case of the constraint equations of spherical 6R closed chain. By turning the equality equations of 4R and 5R closed chains (Eqs. (3.75), (3.76), and (3.82)) into inequalities, we can apply the same algorithm as presented before to do constrained interpolation for 4R and 5R closed chains as well.

For spherical 4R closed chains, we modify the kinematic constraints (Eqs. (3.75) and (3.76)) as follows:

$$\cos^2(\frac{\alpha_1+\delta_1}{2}) \leq F_1(Y_1,Y_2,Y_3,Y_4) \leq \cos^2(\frac{\alpha_1-\delta_1}{2}), \tag{5.34}$$

$$\cos^2(\frac{\alpha_2+\delta_2}{2}) \leq F_2(Y_1,Y_2,Y_3,Y_4) \leq \cos^2(\frac{\alpha_2-\delta_2}{2}) \tag{5.35}$$

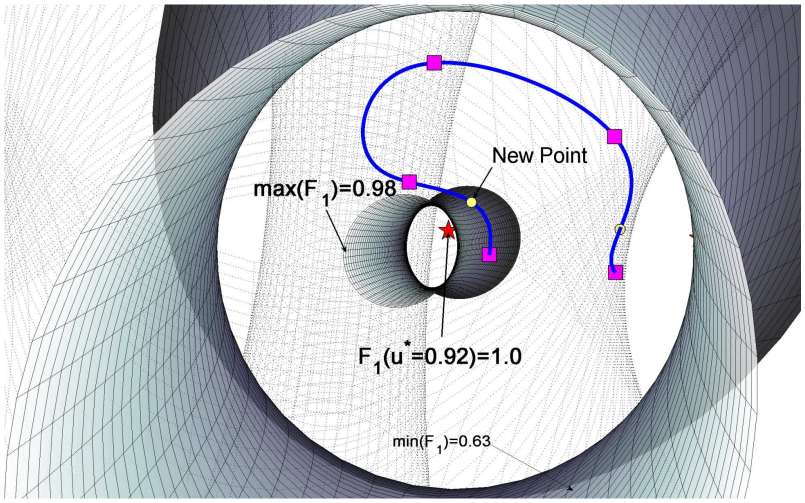

Figure 5.9: Constrained interpolation (compare with Figure 5.7): The labeled new point ('o') is the point that replaces the labeled extreme point.

where δ_1 and δ_2 are user-defined tolerances. Thus, by choosing these values to be as small as possible, the user can use the same algorithm to do constraint interpolation to a desired degree of satisfaction. The addition of new points is handled slightly differently; we use the following equations in place of Eq. (5.19):

$$r_1 = \cos\frac{\alpha_1}{2},\ r_2 = \cos\frac{\alpha_2}{2}. \tag{5.36}$$

In case of spherical 5R closed chain, we modify the kinematic constraint (Eq. (3.82)) to be same as the kinematic constraint Eq. (5.34). This puts the first kinematic constraint in the inequality form, while we use the second constraint equation unchanged, i.e., Eq. (3.83). The new points are added in the same way as illustrated before; however, the value of r_1 in the Eq. (5.19) is replaced with the value given in Eq. (5.36).

5.5 Conclusions

In this chapter, we studied the problem of rational motion interpolation subject to the kinematic constraints of planar and spherical 4R, 5R, and 6R closed chains. We presented algorithms for rational B-spline motion interpolation of given positions of the coupler link of a planar or a spherical 6R closed chain which satisfies the kinematic constraints exactly. We showed these algorithms are also applicable to planar and spherical 4R and 5R closed chains that result in motions that approximate the kinematic constraints.

Chapter 6

Constrained Motion Interpolation for Spatial Mechanisms

6.1 Introduction

The purpose of this chapter is to study the problem of synthesizing rational motions under kinematic constraints of spatial mechanisms, such as spatial SS open chain and ADEPT robot. In this chapter, the spatial displacement is represented by a dual quaternion (see Bottema and Roth [8] and McCarthy [10] for quaternion representation of displacements). In this way, the problem of rational motion interpolation is transformed into that of rational curve interpolation, and the kinematic constraints of spatial mechanisms are transformed into geometric constraints for the rational interpolation. Thus, given a series of positions of end effector in Cartesian space, the problem of synthesizing the smooth interpolating rational motion of a spatial SS open chain and an ADEPT robot is reduced to that of designing a smooth rational curve con-

strained to lie on the constraint manifolds of the corresponding spatial mechanisms. We can apply similar algorithms developed in chapter 4 and chapter 5 to the rational motion interpolation under the kinematic constraints of a spatial SS open chain and an ADEPT robot.

The organization of this chapter is as follows. Section 6.2 presents an algorithm for rational motion interpolation of spatial SS open chain. Section 6.3 deals with the problem of synthesizing rational motions under the kinematic constraints of an ADEPT robot.

6.2 Rational Motion Interpolation of Spatial SS Open Chain

In this section, we present an algorithm for synthesizing piecewise ration motions of spatial SS open chain under the kinematic constraints derived in section 3.4.1.

Given: A set of positions of the end effector of a spatial SS open chain in its workspace as well as the corresponding parameter values $u_i (i = 1 \dots n)$ and the length of the first link a.

Find: A rational motion of the end effector that interpolates the given positions at the parameter values subject to the kinematic constraints of the spatial SS open chain.

The given positions of the end effector of a spatial SS open chain can be specified using either Cartesian based parameters $(s_{xi}, s_{yi}, s_{zi}, \theta_i, d_{1i}, d_{2i}, d_{3i})$ or joint coordinates of the open chain $(\alpha_i, \theta_i, \beta_i, \phi_i, \gamma_i)$. In the case of Cartesian

parameters, we use Eqs. (2.1) and (2.13) to convert them into dual quaternions; in the case of joint parameters, we convert them into dual quaternions by using Eq. (3.91).

Detection of Kinematic Constraints Violation

Finding a motion that satisfies the requirements stated earlier in this section begins with the construction of a cubic B-spline image curve [1] $\hat{\mathbf{q}}(u)$ that interpolates $\hat{\mathbf{q}}_i$ at parameter values u_i. Such a curve does not automatically satisfy the kinematic constraints given by Eqs. (3.92), (2.16), and (2.17). Specifically, kinematic constraints are considered violated at those points of the synthesized image curve $\hat{\mathbf{q}}(u)$, where the following inequalities (rewritten from Eqs. (3.92), (2.16), and (2.17)) do not satisfy:

$$|F_1(u)| = |q_1^2 + q_2^2 + q_3^2 + q_4^2 - 1| \leq \delta_1, \tag{6.1}$$

$$|F_2(u)| = |q_1 q_1^0 + q_2 q_2^0 + q_3 q_3^0 + q_4 q_4^0| \leq \delta_2, \tag{6.2}$$

$$|F_3(u)| = |(q_1^0)^2 + (q_2^0)^2 + (q_3^0)^2 + (q_4^0)^2 - a^2/4| \leq \delta_3, \tag{6.3}$$

where parameters, δ_1, δ_2, and δ_3 are specified tolerances that define how far deviated the points from the kinematic constraints.

To detect violations of the kinematic constraints, extrema of functions $F_1(u)$, $F_2(u)$, and $F_3(u)$ are calculated using the following equations:

$$\frac{dF_1(u)}{du} = 0, \quad \frac{dF_2(u)}{du} = 0, \quad \frac{dF_3(u)}{du} = 0. \tag{6.4}$$

[1] Designing a C^2 B-spline curve is a standard scheme in CAGD (see Farin [11], Hoschek and Lasser [14], and Piegl and Tiller [15]).

If an extremum falls outside the limits in any of the inequalities (Eqs. (6.1),(6.2), and (6.3)), the constraints are considered violated. We call a point on the image curve corresponding to an extremum of $F_1(u)$, $F_2(u)$, or $F_3(u)$ an extreme point. A C^2 cubic B-spline curve can be easily converted into a piecewise Bézier form using standard methods (see Farin [11]), where each Bézier segment has an algebraic form. The piecewise Bézier form allows easy evaluation of Eq. (6.4) for each of the L Bézier segments, $\hat{\mathbf{q}}_i(u)$; $i = 0, 1, \ldots, L-1$, that make up the B-spline curve $\hat{\mathbf{q}}(u)$.

Modifying the Curve to Satisfy Kinematic Constraints

By restricting to operate in the image space, we have transformed the kinematic constraints into geometric constraints. Thus, we can take a purely geometric approach to modify the synthesized image curve in such a way that the kinematic constraints are satisfied. Our approach is to replace the extreme point that violates the kinematic constraints with a new point that satisfies kinematic constraints, thereby dragging the curve to the inside of the constraint manifold. To ensure that the image curve changes minimally, the new point should be closest possible to the extreme point. This approach is realized in a two step iterative process: first, a new point is generated that satisfies the kinematic constraints and is minimally away from the extreme point, then this new point is added (at the parameter value corresponding to the extreme point) to the initial set of given points and a new C^2 cubic B-spline image curve is generated that interpolates the new point as well. If the new image curve detects any further violation of the kinematic constraints as outlined

in the previous subsection, this two-step process is repeated until no extreme points that violate the kinematic constraints are found. This process may yield many new points before the kinematic constraints are satisfied completely.

Generating New Points

Since we require the new points to be minimally away from the extreme points, the issue of finding new points can be seen as a normal distance minimization problem in the image space subject to certain constraints. Ravani and Roth [45] proposed a general algebraic method for approximate normal distance calculation between the image curve and a given position in the image space. Later on, Bodduluri and McCarthy [46] used their method for finite position synthesis of a spherical four-bar motion. In what follows, we outline this method and conform the issue of finding new points to the framework of this method for solution.

Note: *In this subsection we use vector* $\mathbf{q} = (q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8)$ *in* $\mathbb{R}^8$ *to represent the corresponding dual quaternion* $\hat{\mathbf{q}} = \mathbf{q} + \epsilon\mathbf{q}_0$.

In the context of spherical four-bar mechanism synthesis, the method used by Bodduluri and McCarthy [46] solves the design problem of determining an image curve that passes through or near a set of given points. The image curve is algebraically given by intersection of two quadric hypersurfaces (constraint surfaces) and is expressed as functions of the design parameters, such as link lengths and the pivot locations. They seek the values of design parameters which reduce the error between the image curve and the given points. The error at each given position is defined as the normal distance to

the image curve. They define total error as the sum of squares of each position error. The method is approximate in the sense that the constraint surfaces are approximated by their tangent hyperplane in the vicinity of the desired position. We use the same approach to find a new point $\mathbf{q}$ that satisfies the constraints given by Eqs. (6.1),(6.2), and (6.3) and is also at the minimum distance from the extreme point ($\mathbf{q}^*$) that violates the kinematic constraints. In our case, we define $\mathbf{e} = \mathbf{q} - \mathbf{q}^*$ as the normal error vector at the position $\mathbf{q}^*$. Minimization of the square of the magnitude of $\mathbf{e}$ would yield the new point $\mathbf{q} = \mathbf{e}^* + \mathbf{q}^*$, where $\mathbf{e}^*$ is the optimized normal distance. This new point $\mathbf{q}$ may not satisfy the kinematic constraints because the constraint surfaces are only approximated in this approach. In that case, this new point $\mathbf{q}$ is set as the new extreme point $\mathbf{q}^*$ and the process described above is repeated.

The optimization procedure outlined above involves the tangent hyperplane approximation to the hypersurfaces. For that, we first rewrite Eqs. (6.1),(6.2), and (6.3) as follows:

$$H_1(\mathbf{q}) : q_1^2 + q_2^2 + q_3^2 + q_4^2 - 1 = h_1, \tag{6.5}$$

$$H_2(\mathbf{q}) : q_5^2 + q_6^2 + q_7^2 + q_8^2 - a^2/4 = h_2, \tag{6.6}$$

$$H_3(\mathbf{q}) : q_1 q_5 + q_2 q_6 + q_3 q_7 + q_4 q_8 = h_3, \tag{6.7}$$

where, $h_1 = h_2 = h_3 = 0$.

These surfaces can be approximated by the tangent hyperplanes at a point

$\mathbf{q}^*$ by following:

$$\begin{aligned} 0 &= H_1(\mathbf{q}^*) + \sum_{i=1}^{8} \frac{\partial H_1(\mathbf{q}^*)}{\partial q_i} \Delta q_i, \\ 0 &= H_2(\mathbf{q}^*) + \sum_{i=1}^{8} \frac{\partial H_2(\mathbf{q}^*)}{\partial q_i} \Delta q_i, \\ 0 &= H_3(\mathbf{q}^*) + \sum_{i=1}^{8} \frac{\partial H_3(\mathbf{q}^*)}{\partial q_i} \Delta q_i. \end{aligned} \tag{6.8}$$

These equations can be assembled as follows:

$$\begin{bmatrix} 2q_1 & 2q_2 & 2q_3 & 2q_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2q_5 & 2q_6 & 2q_7 & 2q_8 \\ q_5 & q_6 & q_7 & q_8 & q_1 & q_2 & q_3 & q_4 \end{bmatrix} \begin{bmatrix} \Delta q_1 \\ \Delta q_2 \\ \Delta q_3 \\ \Delta q_4 \\ \Delta q_5 \\ \Delta q_6 \\ \Delta q_7 \\ \Delta q_8 \end{bmatrix} = \begin{bmatrix} -H_1 \\ -H_2 \\ -H_3 \end{bmatrix}. \tag{6.9}$$

The above can also be written as

$$[J]\mathbf{e} = \mathbf{v}. \tag{6.10}$$

Bodduluri and McCarthy [46] solve for the normal error vector $\mathbf{e}$ by minimizing the Lagrangian function given as follows:

$$L(\mathbf{e}, \Lambda) = \mathbf{e}^T\mathbf{e} + \Lambda^T([J]\mathbf{e} - \mathbf{v}), \tag{6.11}$$

where $\Lambda = (\Lambda_1, \Lambda_2, \Lambda_3)$ is a vector of Lagrange multipliers, $\mathbf{e} = \mathbf{q} - \mathbf{q}^*$, $\mathbf{v} = (-H_1(\mathbf{q}^*),\ -H_2(\mathbf{q}^*),\ -H_3(\mathbf{q}^*))$. Putting the condition for a minimum as

$$\frac{\partial L}{\partial e_i} = 0, \qquad i = 1, \ldots, 8, \tag{6.12}$$

where e_i are the coordinates of the vector $\mathbf{e}$ and assembling the solution equations, we obtain

$$2\mathbf{e}^T + \Lambda^T[J] = 0. \tag{6.13}$$

Thus, if $\mathbf{e}^*$ designates the solution to the error vector (or, the normal distance) then it should satisfy the equations

$$\begin{aligned} [J]\mathbf{e}^* &= \mathbf{v}, \\ 2\mathbf{e}^* + [J]^T\Lambda &= 0. \end{aligned} \tag{6.14}$$

Equation (6.14) gives an explicit formula for the solution error vector $\mathbf{e}^*$ in terms of variables r_1 and r_2 (see Bodduluri and McCarthy [46]):

$$\mathbf{e}^* = [J]^T([J][J]^T)^{-1}\mathbf{v}. \tag{6.15}$$

With normal error vector $\mathbf{e}^*$ known, the new point is given by

$$\mathbf{q} = \mathbf{e}^* + \mathbf{q}^*. \tag{6.16}$$

Since the constraint surfaces have been approximated by their tangent hyperplanes in the vicinity of the extreme point $\mathbf{q}^*$, this new point $\mathbf{q}$ may or may not lie on the constraint surface given by Eqs. (6.1),(6.2), and (6.3). If the new point does not satisfy the constraints, the newly obtained point $\mathbf{q}$ is set as a new $\mathbf{q}^*$ and the procedure described above is repeated from Eq. (6.8) until a new point $\mathbf{q}$ is obtained that satisfies the kinematic constraints. This new point not only satisfies the kinematic constraints but is also at the shortest distance from the original extreme point $\mathbf{q}^*$. With this new point added to

the set of initial points in the quaternion space, a new C^2 B-spline is interpolated to the points. If the new image curve detects any further violation of the kinematic constraints, the optimization process is repeated until no further extreme points that violate the kinematic constraints are found. This process at the end gives an interpolating motion that satisfies the kinematic constraints. Now, we present the algorithm:

Algorithm 6.2

1. Convert given positions of the coupler link into unit quaternions (points in image space) $\hat{\mathbf{q}}_i$ using either Eqs. (2.1) and (2.13) or Eq. (3.91).

2. Current list of points to be interpolated = given points $(\hat{\mathbf{q}}_i; i = 1 \ldots n)$

3. Construct a C^2 cubic B-spline curve $\hat{\mathbf{q}}(u)$ that interpolates current list of points.

4. (a) Evaluate the extrema of $F_1(u)$, $F_2(u)$, and $F_3(u)$ using Eq. (6.4). Say, the extrema are found at $u = u_1^*$, at $u = u_2^*$, and at $u = u_3^*$ for $F_1(u)$, $F_2(u)$, and $F_3(u)$ respectively. Designate extreme points on the curve as $\hat{\mathbf{q}}_1^*$, $\hat{\mathbf{q}}_2^*$, and $\hat{\mathbf{q}}_3^*$.

   (b) Check if $\hat{\mathbf{q}}_1^*$, $\hat{\mathbf{q}}_2^*$, and $\hat{\mathbf{q}}_3^*$ satisfy the kinematic constraints (Eqs.(6.1), (6.2), and (6.3)).

   (c) If yes, the curve is constrained; continue to Step 5; else, find new points $\hat{\mathbf{q}}(u_1^*)$, $\hat{\mathbf{q}}(u_2^*)$, and $\hat{\mathbf{q}}(u_3^*)$.

(d) Add $\hat{\mathbf{q}}(u_1^*)$, $\hat{\mathbf{q}}(u_2^*)$, and $\hat{\mathbf{q}}(u_3^*)$ to the current list of points to be interpolated and go to Step 3.

5. The image curve $\hat{\mathbf{q}}(u)$ defines a C^2 interpolating piecewise rational motion of degree 6 after the substitution into Eq. (2.11).

We have observed that this algorithm always converges as long as the interpolating points are on the same branch. In this algorithm, the B-spline curve is generated using a global interpolation scheme (Piegl and Tiller [15]). In this scheme, although moving one of the interpolating points changes the curve globally, the change in the curve diminishes away from the modification point. Next we show an example.

Table 6.1 gives joint parameters for the five positions of the end effector of a spatial SS open chain along with their parameter values. The joint parameters are converted into dual quaternions using Eq. (3.91), which are shown in Table 6.2. We use **Algorithm 6.2** to find extreme positions ,generate new positions, and finally generate a C^2 B-spline curve within the kinematic constraints of the spatial SS open chain. The extreme positions, new positions, corresponding kinematic constraint values, and control positions of final C^2 B-spline curve are presented in Table 6.3, 6.4, 6.5, and 6.6, respectively. In this example, we use the deviation values $\delta_1 = 0.02$, $\delta_2 = 0.02$, and $\delta_3 = 0.02$ in Eqs. (6.1), (6.2), and (6.3). The parameter sequence for the final C^2 rational B-spline curve in Table 6.6 is $[0.0, 0.7501, 1.1066, 1.4564, 2.1101, 2.7559, 3.1917, 3.8652, 4.8226, 6.0534, 7.2038, 8.0158, 8.6900, 10.0]$. The algorithm converges at the third iteration in this example.

i	$(\alpha_i, \theta_i, \beta_i, \phi_i, \gamma_i)$ deg	u_i
1	$(10, 45, 20, 30, 30)$	0.0
2	$(-20, 60, 30, 35, 75)$	2.0
3	$(60, 25, -30, 75, 30)$	5.0
4	$(45, 30, 20, 20, 20)$	7.0
5	$(-30, 30, -30, 30, 100)$	10.0

Table 6.1: **Joint coordinates of the given positions of the end effector of spatial SS open chain** $(a = 2)$.

i	$\mathbf{q}_i(q_{i1}, q_{i2}, q_{i3}, q_{i4})$	$\mathbf{q}_i^0(q_{i1}^0, q_{i2}^0, q_{i3}^0, q_{i4}^0)$	u_i
1	$(0.4290, 0.1264, 0.5865, 0.6753)$	$(0.8704, 0.1082, -0.1264, -0.4634)$	0.0
2	$(0.5255, 0.5629, 0.4398, 0.4622)$	$(0.7557, 0.0006, -0.2831, -0.5905)$	2.0
3	$(0.2732, -0.0859, 0.7430, 0.6049)$	$(0.7367, -0.4456, 0.0859, -0.5014)$	5.0
4	$(0.6255, -0.0753, 0.4096, 0.6598)$	$(0.7429, 0.0997, -0.0531, -0.6599)$	7.0
5	$(0.2678, 0.4377, 0.2041, 0.8337)$	$(0.9198, 0.1173, 0.0547, -0.3704)$	10.0

Table 6.2: **Dual quaternions of the given positions of the end effector of spatial SS open chain** $(a = 2)$.

i	$\mathbf{q}_i^*(q_{i1}^*, q_{i2}^*, q_{i3}^*, q_{i4}^*)$	$(\mathbf{q}_i^0)^*((q_{i1}^0)^*, (q_{i2}^0)^*, (q_{i3}^0)^*, (q_{i4}^0)^*)$	u_i^*
1	$(0.4965, 0.2849, 0.5067, 0.6152)$	$(0.8452, 0.1557, -0.2051, -0.5114)$	0.28
2	$(0.5741, 0.5184, 0.4070, 0.5163)$	$(0.7984, 0.1706, -0.3070, -0.5799)$	0.95
3	$(0.5763, 0.5319, 0.4031, 0.5094)$	$(0.7946, 0.1656, -0.3114, -0.5837)$	1.02
4	$(0.5751, 0.5749, 0.3975, 0.4830)$	$(0.7782, 0.1271, -0.3198, -0.5953)$	1.35
5	$(0.3892, 0.3804, 0.5791, 0.4871)$	$(0.7401, -0.2456, -0.1461, -0.5437)$	2.98
6	$(0.3450, 0.3007, 0.6274, 0.5038)$	$(0.7383, -0.3187, -0.0953, -0.5263)$	3.28
7	$(0.3423, 0.2954, 0.6304, 0.5049)$	$(0.7382, -0.3231, -0.0920, -0.5252)$	3.31
8	$(0.4607, -0.1352, 0.5842, 0.6411)$	$(0.7358, -0.1703, 0.0293, -0.5894)$	6.16
9	$(0.7395, 0.1465, 0.1547, 0.7088)$	$(0.7900, 0.4208, -0.1407, -0.6779)$	8.45
10	$(0.7278, 0.1733, 0.1406, 0.7164)$	$(0.7981, 0.4287, -0.1389, -0.6668)$	8.60
11	$(0.6592, 0.2544, 0.1181, 0.7431)$	$(0.8264, 0.4145, -0.1168, -0.6152)$	9.02

Table 6.3: **Extreme positions**.

i	$\mathbf{q}_i(q_{i1}, q_{i2}, q_{i3}, q_{i4})$	$\mathbf{q}_i^0(q_{i1}^0, q_{i2}^0, q_{i3}^0, q_{i4}^0)$	u_i
1	$(0.4874, 0.2872, 0.5217, 0.6393)$	$(0.8177, 0.1462, -0.2126, -0.5155)$	0.28
2	$(0.5255, 0.5047, 0.4205, 0.5438)$	$(0.7364, 0.1357, -0.3176, -0.5862)$	0.95
3	$(0.5262, 0.5168, 0.4156, 0.5356)$	$(0.7329, 0.1301, -0.3218, -0.5896)$	1.02
4	$(0.5268, 0.5573, 0.4046, 0.5013)$	$(0.7264, 0.0960, -0.3284, -0.5992)$	1.35
5	$(0.4990, 0.3915, 0.6185, 0.4785)$	$(0.8210, -0.2218, -0.0970, -0.5264)$	2.98
6	$(0.4604, 0.2978, 0.6826, 0.4998)$	$(0.8162, -0.3040, -0.0330, -0.5011)$	3.28
7	$(0.4574, 0.2915, 0.6863, 0.5012)$	$(0.8153, -0.3090, -0.0293, -0.4996)$	3.31
8	$(0.4642, -0.1363, 0.5893, 0.6471)$	$(0.7681, -0.1777, 0.0302, -0.6159)$	6.16
9	$(0.6715, 0.1193, 0.1564, 0.7182)$	$(0.6696, 0.3695, -0.1341, -0.6444)$	8.45
10	$(0.6572, 0.1429, 0.1438, 0.7301)$	$(0.6736, 0.3743, -0.1326, -0.6386)$	8.60
11	$(0.5895, 0.2209, 0.1239, 0.7715)$	$(0.7072, 0.3600, -0.1145, -0.6118)$	9.02

Table 6.4: **New positions generated for extreme positions**.

i	$(F_{i1}(u_i^*), F_{i2}(u_i^*), F_{i3}(u_i^*))$	u_i^*/u_i	$(F_{i1}(u_i), F_{i2}(u_i), F_{i3}(u_i))$
1	$(-0.0371, 0.0422, 0.0454)$	0.28	$(0.0009, 0.0009, 0.0001)$
2	$(0.0305, 0.0970, 0.1224)$	0.95	$(0.0035, 0.0052, 0.0032)$
3	$(0.0371, 0.0965, 0.1232)$	1.02	$(0.0036, 0.0052, 0.0033)$
4	$(0.0524, 0.0784, 0.1060)$	1.35	$(0.0030, 0.0037, 0.0029)$
5	$(-0.1312, -0.0750, -0.1548)$	2.98	$(0.0138, 0.0098, 0.0109)$
6	$(-0.1432, -0.0674, -0.1660)$	3.28	$(0.0164, 0.0108, 0.0123)$
7	$(-0.1432, -0.0664, -0.1660)$	3.31	$(0.0164, 0.0107, 0.0122)$
8	$(-0.0172, -0.0814, 0.0012)$	6.16	$(0.0001, 0.0018, -0.0000)$
9	$(0.0946, 0.2805, 0.1435)$	8.45	$(0.0055, 0.0183, 0.0099)$
10	$(0.0928, 0.2846, 0.1579)$	8.60	$(0.0061, 0.0193, 0.0108)$
11	$(0.0654, 0.2469, 0.1792)$	9.02	$(0.0068, 0.0172, 0.0102)$

Table 6.5: **Kinematic constraints for extreme and new positions;** $|F_1| \leq 0.02$**,** $|F_2| \leq 0.02$**, and** $|F_3| \leq 0.02$.

i	$\mathbf{D}_i(D_{i1}, D_{i2}, D_{i3}, D_{i4})$	$\mathbf{D}_i^0(D_{i1}^0, D_{i2}^0, D_{i3}^0, D_{i4}^0)$
1	$(0.4290, 0.1264, 0.5865, 0.6753)$	$(0.8704, 0.1082, -0.1264, -0.4634)$
2	$(0.4993, 0.2918, 0.5251, 0.6480)$	$(0.8208, 0.1584, -0.2160, -0.5162)$
3	$(0.5223, 0.4432, 0.4466, 0.5850)$	$(0.7556, 0.1617, -0.2952, -0.5683)$
4	$(0.5275, 0.5358, 0.4068, 0.5240)$	$(0.7261, 0.1246, -0.3298, -0.5954)$
5	$(0.5262, 0.5784, 0.4003, 0.4810)$	$(0.7247, 0.0731, -0.3296, -0.6035)$
6	$(0.5247, 0.5705, 0.4401, 0.4527)$	$(0.7605, -0.0118, -0.2806, -0.5908)$
7	$(0.5272, 0.4792, 0.5504, 0.4590)$	$(0.8182, -0.1400, -0.1643, -0.5524)$
8	$(0.4741, 0.3070, 0.6868, 0.4969)$	$(0.8266, -0.3028, -0.0296, -0.5007)$
9	$(0.3304, 0.0758, 0.7932, 0.5535)$	$(0.7652, -0.4712, 0.0797, -0.4528)$
10	$(0.2083, -0.0968, 0.7649, 0.6018)$	$(0.7132, -0.5032, 0.0963, -0.4741)$
11	$(0.4315, -0.1705, 0.6391, 0.6520)$	$(0.7885, -0.2497, 0.0582, -0.6213)$
12	$(0.6682, -0.0725, 0.3827, 0.6574)$	$(0.7451, 0.1557, -0.0700, -0.6720)$
13	$(0.7055, 0.0462, 0.2068, 0.6824)$	$(0.6682, 0.3395, -0.1331, -0.6595)$
14	$(0.6490, 0.1846, 0.1029, 0.7490)$	$(0.6630, 0.4079, -0.1397, -0.6330)$
15	$(0.4692, 0.3330, 0.1208, 0.8371)$	$(0.7799, 0.3058, -0.0769, -0.5728)$
16	$(0.2678, 0.4377, 0.2041, 0.8337)$	$(0.9198, 0.1173, 0.0547, -0.3704)$

Table 6.6: **Control points of C^2 B-spline curve**.

6.3 Rational Motion Interpolation of ADEPT Robot

In this section, we present an algorithm for synthesizing continuous piecewise ration motions of ADEPT robot under the kinematic constraints derived in section 3.4.2.

Given: A set of positions of the end effector of a ADEPT robot in its workspace, the corresponding parameter values $u_i (i = 1 \dots n)$, the length of the first and the second link a and b, respectively.

Find: A rational motion of the end effector that interpolates the given positions at the parameter values subject to the kinematic constraints of the

ADEPT robot.

The given positions of the end effector of an ADEPT robot can be specified using either Cartesian based parameters $(s_{xi}, s_{yi}, s_{zi}, \theta_i, d_{1i}, d_{2i}, d_{3i})$ or joint coordinates of the open chain $(\theta_i, \phi_i, \psi_i, c_i)$. In the case of Cartesian parameters, we use Eqs. (2.1) and (2.13) to convert them into dual quaternions; in the case of joint parameters, we convert them to dual quaternions by using Eq. (3.95).

The kinematic constraint of an ADEPT robot Eq. (3.96) is the equation of a circle. Thus we can apply **Algorithm 4.2** for C^1 piecewise rational Bézier interpolation on a circle, or we can use **Algorithm 4.3** for smooth (C^2 or higher) rational motions that approximate the circle kinematic constraint. For Eq. (3.97), which is the equation of a circular ring, **Algorithm 4.3** can be used for C^2 or higher rational B-spline interpolation within the the circular ring kinematic constraint. Finally for Eqs. (3.98) and (3.99) we could use one dimensional version of **Algorithm 4.3** to interpolate within the kinematic constraints.

6.4 Conclusions

In this chapter, we studied the problem of rational motion interpolation subject to the kinematic constraints of a spatial SS open chain and an ADEPT robot. We presented an algorithms for rational B-spline motion interpolation of given positions of the end effector of a spatial SS open which approximates the kinematic constraints. We also showed an algorithm for rational B-spline motion interpolation of the end effector of an ADEPT robot which satisfies

the kinematic constraints.

Chapter 7

Matrix Approach to Constrained Motion Interpolation for Planar Open and Closed Chains

7.1 Introduction

The purpose of this chapter is to study the problem of synthesizing rational motions of a rigid body under kinematic constraints that are imposed by planar open and closed kinematic chains. This chapter adopts a different approach for motion interpolation from chapter 4 and chapter 5 by investigating the matrix representation of planar displacements directly. It has been found that the elements of the displacement matrix can be used directly for synthesizing rational motions as well as for characterizing the kinematic constraints of planar kinematic chains. Interestingly, the kinematic constraints for planar 2R and 3R can be reduced to circles and circular rings, essentially the same as those represented using planar quaternions. Thus the algorithms developed

by chapter 4 may be directly applied to the elements of the displacement matrix. Furthermore we can apply a similar algorithm developed in chapter 5 to the problem of synthesizing rational motions of planar 4R, 5R, and 6R closed chains in a parametric space defined by the elements of the displacement matrix. The advantage of this approach is that the handling of the interpolation process as well as the kinematic constraints are more straightforward. In addition, the degree of the resulting rational motion is lower. For example, while the degree of the rational motion resulting from a cubic interpolation of planar quaternions is six, the degree of the rational motion resulting from a cubic interpolation of matrix elements remains to be three.

The organization of this chapter is as follows. Section 7.2 presents kinematics of planar open and closed chains. In this section, the algebraic form of kinematic constraints are derived using the elements of the displacement matrix. Section 7.3 deals with the problem of constrained motion interpolation of a planar 6R closed chain in a parametric space defined by the elements of the displacement matrix. Section 7.4 shows how the algorithm presented in the previous section can be used to do rational motion interpolation of planar 4R and 5R closed chains. Section 7.5 deals with the problem of constrained motion planning of various open chains in a parametric space defined by the elements of the displacement matrix.

7.2 Kinematic Constraints of Planar Open and Closed Chains

This section derives the algebraic relations that characterize the kinematic constraints of various planar open and closed chains in terms of the displacement matrix elements. The goal is to present a representation of rational motion that can easily handle the kinematic constraints of the planar chains. A planar quaternion based formulation of the kinematics of planar open and closed chains is presented in chapter 3.

7.2.1 Planar Displacement Matrix

The components of a planar quaternion are related to the planar displacement matrix by:

$$[M] = \begin{bmatrix} m_1 & -m_2 & m_3 \\ m_2 & m_1 & m_4 \\ 0 & 0 & 1 \end{bmatrix}, \tag{7.1}$$

where

$$\begin{array}{ll} m_1 = Z_4^2 - Z_3^2, & m_2 = 2Z_3Z_4, \\ m_3 = 2(Z_1Z_4 - Z_2Z_3), & m_4 = 2(Z_1Z_3 + Z_2Z_4). \end{array} \tag{7.2}$$

It follows that

$$m_1^2 + m_2^2 = 1. \tag{7.3}$$

This circular constraint (7.3) ensures that the matrix $[M]$ represents a rigid-body transformation.

If $\mathbf{m} = (m_1, m_2, m_3, m_4)$ are rational functions of degree n in parameter u (u being usually associated with time), such that the circular condition (7.3) is satisfied, then the matrix $[M]$ represents a rational motion of degree n.

On the other hand, from Eq. (7.2) it is clear that choosing planar quaternion $\mathbf{Z} = (Z_1, Z_2, Z_3, Z_4)$ to construct a rational motion would produce a rational motion of degree $2n$. In this chapter, we use $\mathbf{m} = (m_1, m_2, m_3, m_4)$ directly to construct a rational motion.

Let $\mathbf{m}_i = (m_{i1}, m_{i2}, m_{i3}, m_{i4});\ i = 0, ..., n$ be $(n+1)$ vectors of displacement matrix parameters, then the following represents a Bézier curve in the parameter space:

$$\mathbf{m}(u) = \sum_{i=0}^{n} B_i^n(u)\mathbf{m}_i. \tag{7.4}$$

where $B_i^n(u)$ are the Bernstein polynomials.

Similarly, a B-Spline curve in the parameter space is given by:

$$\mathbf{m}(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{m}_i. \tag{7.5}$$

where $N_{i,p}(u)$ are pth-degree basis functions.

A representation for the rational Bézier motion and rational B-Spline motion in the Cartesian space is obtained by substituting $\mathbf{m}(u)$ from Eq. (7.4) or Eq. (7.5) into the homogeneous matrix $[M]$. From Eq. (7.1), it can be seen that if the B-Spline curve $\mathbf{m}(u)$ is expressed as a polynomial function of degree p, then the matrix $[M]$ represents a rational B-Spline motion of degree p.

Since m_1 and m_2 must satisfy the circular constraint in order to meet the requirement of rigid transformation, the goal of the remainder of the section is to derive constraints on m_3 and m_4 that characterize the kinematic constraints associated with various planar open and closed chains.

7.2.2 Planar 2R Open Chain

The planar quaternion for the second link of the planar 2R open chain (see Figure 3.1) is defined by Eq. (3.2).

The substitution of Eq.(3.2) into Eq. (7.2) leads to

$$\begin{array}{ll} m_1 = \cos(\theta + \phi), & m_2 = \sin(\theta + \phi), \\ m_3 = a\cos(\theta), & m_4 = a\sin(\theta). \end{array} \tag{7.6}$$

Thus we have

$$m_3^2 + m_4^2 = a^2. \tag{7.7}$$

The above equation characterizes the kinematic constraint of a planar 2R open chain. Thus, the problem of interpolating planar displacements for a planar 2R open chain can be reduced to that of circular interpolation in two separate planes, one in the $m_1 m_2$ plane to meet the general planar rigid-body requirement (7.3) and the other in the $m_3 m_4$ plane to meet the specific requirement (7.7) for the planar 2R chain.

7.2.3 Planar PR Open Chain

The planar quaternion for the second link of the planar PR open chain (see Figure 3.2) is defined by Eq. (3.5).

The substitution of Eq.(3.5) into Eq. (7.2) leads to

$$\begin{array}{ll} m_1 = \cos(\theta), & m_2 = \sin(\theta), \\ m_3 = b, & m_4 = 0. \end{array} \tag{7.8}$$

Let b_1, b_2 denote the lower and upper limits for the range of travel of the

prismatic joint, then we have the kinematic constraints for planar PR arm:

$$b_1 \leq \quad m_3 \quad \leq b_2, \tag{7.9}$$

$$m_4 \quad = \quad 0. \tag{7.10}$$

7.2.4 Planar RP Open Chain

Similar to a planar RP open chain (see Figure (3.3)), the planar quaternion for the end link of RP open chain is given by Eq. (3.7).

The substitution of Eq.(3.7) into Eq. (7.2) leads to

$$\begin{array}{ll} m_1 = \cos(\theta), & m_2 = \sin(\theta), \\ m_3 = b\cos(\theta), & m_4 = b\sin(\theta). \end{array} \tag{7.11}$$

In addition to the common circle constraint (7.3), we have

$$\begin{array}{l} m_3 = bm_1, \\ m_4 = bm_2, \\ b_1 \leq b \leq b_2. \end{array} \tag{7.12}$$

7.2.5 Planar 3R Open Chain

The planar quaternion for the end link of the planar 3R open chain (see Figure 3.4) is defined by Eq. (3.10).

The substitution of Eq.(3.10) into Eq. (7.2) leads to

$$m_1 = \cos(\theta + \phi + \psi), \quad m_2 = \sin(\theta + \phi + \psi), \tag{7.13}$$

$$m_3 = a\cos(\theta) + b\cos(\theta + \phi),$$

$$m_4 = a\sin(\theta) + b\sin(\theta + \phi).$$

It is not difficult to show that

$$m_3^2 + m_4^2 = a^2 + b^2 + 2ab\cos(\phi). \tag{7.14}$$

Since the range of $\cos(\phi)$ is [-1 1], it follows that

$$(a-b)^2 \leq m_3^2 + m_4^2 \leq (a+b)^2. \tag{7.15}$$

Eq. (7.15) characterizes the kinematic constraint of the planar 3R open chain. It represents a circular ring in the $m_3 m_4$ plane.

7.2.6 Planar RRP Open Chain

The planar quaternion for the end link of the planar 3R open chain (see Figure 3.5) is defined by Eq. (3.15).

The substitution of Eq.(3.15) into Eq. (7.2) leads to

$$\begin{array}{l} m_1 = \cos(\theta+\phi), \quad m_2 = \sin(\theta+\phi), \\ m_3 = b\cos(\theta+\phi) + a\cos(\theta), \\ m_4 = b\sin(\theta+\phi) + a\sin(\theta). \end{array} \tag{7.16}$$

It is clear that

$$(m_3 - bm_1)^2 + (m_4 - bm_2)^2 = a^2. \tag{7.17}$$

Eq.(7.17) and the range of b ($b_1 \leq b \leq b_2$) guarantee the motion is within workspace.

7.2.7 Planar RPR Open Chain

The planar quaternion for the end link of the planar 3R open chain (see Figure 3.6) is defined by Eq. (3.18).

The substitution of Eq.(3.18) into Eq. (7.2) leads to

$$\begin{array}{ll} m_1 = \cos(\theta+\phi), & m_2 = \sin(\theta+\phi), \\ m_3 = b\cos(\theta), & m_4 = b\sin(\theta). \end{array} \tag{7.18}$$

It follows that

$$b_1^2 \leq m_3^2 + m_4^2 = b^2 \leq b_2^2, \tag{7.19}$$

where we assume that $b_1 \leq b \leq b_2$.

7.2.8 Planar PRR Open Chain

The planar quaternion for the end link of the planar 3R open chain (see Figure 3.7) is defined by Eq. (3.21).

The substitution of Eq.(3.21) into Eq. (7.2) leads to

$$\begin{array}{ll} m_1 = \cos(\theta + \phi), & m_2 = \sin(\theta + \phi), \\ m_3 = b + a\cos(\theta), & m_4 = a\sin(\theta). \end{array} \tag{7.20}$$

Thus the kinematic constraint associated with the PRR chain is given by

$$\begin{array}{c} (m_3 - b)^2 + m_4^2 = a^2, \\ b_1 \leq b \leq b_2. \end{array} \tag{7.21}$$

7.2.9 Planar PRP Open Chain

The planar quaternion for the end link of the planar 3R open chain (see Figure 3.8) is defined by Eq. (3.24).

The substitution of Eq.(3.24) into Eq. (7.2) leads to

$$\begin{array}{ll} m_1 = \cos(\theta), & m_2 = \sin(\theta), \\ m_3 = b + c\cos(\theta), & m_4 = c\sin(\theta). \end{array} \tag{7.22}$$

The kinematic constraint for the chain is given by

$$\begin{array}{c} m_3 = b + cm_1, \\ m_4 = cm_2, \\ b_1 \leq b \leq b_2, \\ c_1 \leq c \leq c_2. \end{array} \tag{7.23}$$

7.2.10 Planar 4R Closed Chain

The planar 4R closed chain (see Figure 3.9)has one degree of freedom so the constraint manifold is the intersection of two constraint surfaces governed by two planar 2R open chains (see Jin and Ge [1, 3]).

The planar quaternion representing one planar 2R open chain is given by:

$$\begin{aligned} Y_1 &= (a_1/2)\cos((\theta_1 - \phi_1)/2) - \tau\cos((\theta_1 + \phi_1)/2), \\ Y_2 &= (a_1/2)\sin((\theta_1 - \phi_1)/2) + \sigma\sin((\theta_1 + \phi_1)/2), \\ Y_3 &= \sin((\theta_1 + \phi_1)/2), \\ Y_4 &= \cos((\theta_1 + \phi_1)/2). \end{aligned} \tag{7.24}$$

Substituting $Y_i; i = 1 \ldots 4$ from Eq. (7.24) into Eq. (7.2), we obtain elements of the matrix $[M]$, given by $\mathbf{m}$:

$$\begin{aligned} m_1 &= \cos(\theta_1 + \phi_1), \\ m_2 &= \sin(\theta_1 + \phi_1), \\ m_3 &= a_1\cos\theta_1 + (\sigma - \tau)\cos(\theta_1 + \phi_1) - (\sigma + \tau), \\ m_4 &= a_1\sin\theta_1 + (\sigma - \tau)\sin(\theta_1 + \phi_1). \end{aligned} \tag{7.25}$$

Similarly the planar quaternion representing the other planar 2R open chain is given by:

$$\begin{aligned} Y_1 &= (a_2/2)\cos((\theta_2 - \phi_2)/2) + \tau\cos((\theta_2 + \phi_2)/2), \\ Y_2 &= (a_2/2)\sin((\theta_2 - \phi_2)/2) - \sigma\sin((\theta_2 + \phi_2)/2), \\ Y_3 &= \sin((\theta_2 + \phi_2)/2), \\ Y_4 &= \cos((\theta_2 + \phi_2)/2). \end{aligned} \tag{7.26}$$

Substituting $Y_i; i = 1 \ldots 4$ from Eq. (7.26) into Eq. (7.2), we obtain elements of the matrix $[M]$, given by $\mathbf{m}_2$:

$$\begin{aligned} m_1 &= \cos(\theta_2 + \phi_2), \\ m_2 &= \sin(\theta_2 + \phi_2), \\ m_3 &= a_2\cos\theta_2 + (\tau - \sigma)\cos(\theta_1 + \phi_1) + (\sigma + \tau), \\ m_4 &= a_2\sin\theta_2 + (\tau - \sigma)\sin(\theta_1 + \phi_1). \end{aligned} \tag{7.27}$$

The constraint curve for the planar 4R closed chain is the intersection of the constraint surfaces given by (7.25) and (7.27), that is $\mathbf{m}_1 = \mathbf{m}_2 = \mathbf{m} = (m_1, m_2, m_3, m_4)$. The algebraic equations for the kinematic constraints are obtained by eliminating $\theta_i, \phi_i; i = 1, 2$ from Eqs. (7.25) and (7.27):

$$\begin{array}{c} m_1^2 + m_2^2 = 1, \\ (m_3 - (\sigma - \tau)m_1 + (\sigma + \tau))^2 + (m_4 - (\sigma - \tau)m_2)^2 = a_1^2, \\ (m_3 + (\sigma - \tau)m_1 - (\sigma + \tau))^2 + (m_4 + (\sigma - \tau)m_2)^2 = a_2^2. \end{array} \tag{7.28}$$

In this chapter, the constraint equations of the form (7.28) will be used for rational motion synthesis.

7.2.11 Planar 5R Closed Chain

Consider a planar 5R closed chain, Figure 3.10. The constraint manifold for this planar 5R closed chain is a portion of the constraint surface of a planar 2R open chain cut by the inner and outer boundaries of the constraint manifold of a planar 3R open chain.

The planar quaternion representing one planar 2R open chain is given by Eq. (7.24).

The constraint manifold of a planar 3R open chain is given by (see Jin and Ge [53]):

$$\mathbf{Y}_2(\theta_2, \phi_2, \psi_2) \;\; = \;\; [C_2]\mathbf{Z}(\theta_2, \phi_2, \psi_2), \tag{7.29}$$

where $[C_2]$ is given by Eq. (3.30) and $\mathbf{Z}(\theta_2, \phi_2, \psi_2)$ is given by Eq. (3.10).

Substituting $\mathbf{Y}_2$ from Eq. (7.29) into Eq. (7.2), we obtain elements of the

homogeneous matrix $[M]$:

$$\begin{aligned} m_1 &= \cos(\theta_2 + \phi_2 + \psi_2), \\ m_2 &= \sin(\theta_2 + \phi_2 + \psi_2), \\ m_3 &= a_2 \cos\theta_2 + b_2 \cos(\theta_2 + \phi_2) - (\sigma - \tau)\cos(\theta_2 + \phi_2 + \psi_2) \\ &\quad + (\sigma + \tau), \\ m_4 &= a_2 \sin\theta_2 + b_2 \sin(\theta_2 + \phi_2) - (\sigma - \tau)\cos(\theta_2 + \phi_2 + \psi_2). \end{aligned} \tag{7.30}$$

Once again, the kinematic constraint equations are obtained by eliminating θ_2, ϕ_2, ψ_2 from Eq. (7.30). Assembling all the constraint equations:

$$\begin{gathered} m_1^2 + m_2^2 = 1, \\ (m_3 - (\sigma - \tau)m_1 + (\sigma + \tau))^2 + (m_4 - (\sigma - \tau)m_2)^2 = a_1^2, \\ (m_3 + (\sigma - \tau)m_1 - (\sigma + \tau))^2 + (m_4 + (\sigma - \tau)m_2)^2 = R_2^2(\phi_2), \end{gathered} \tag{7.31}$$

where

$$\begin{gathered} R_2^2(\phi_2) = a_2^2 + b_2^2 + 2a_2 b_2 \cos(\phi_2), \\ |a_2 - b_2| \leq R_2(\phi_2) \leq (a_2 + b_2). \end{gathered} \tag{7.32}$$

7.2.12 Planar 6R Closed Chain

Consider a planar 6R closed chain, see Figure 3.11. The constraint manifold for this planar 6R closed chain is the intersection of constraint manifolds of two planar 3R open chains. Similar to the case of a planar 5R closed chain, the two constraint manifolds for the 6R chain are given by

$$\begin{aligned} \mathbf{Y}_1(\theta_1, \phi_1, \psi_1) &= [C_1]\mathbf{Z}(\theta_1, \phi_1, \psi_1), \\ \mathbf{Y}_2(\theta_2, \phi_2, \psi_2) &= [C_2]\mathbf{Z}(\theta_2, \phi_2, \psi_2). \end{aligned} \tag{7.33}$$

Following the same procedure as outlined in the previous subsections, the kinematic constraints in terms of the elements of the homogeneous matrix $[M]$

for a planar 6R closed chain are obtained and given by:

$$\begin{gathered} m_1^2 + m_2^2 = 1, \\ (m_3 - (\sigma - \tau)m_1 + (\sigma + \tau))^2 + (m_4 - (\sigma - \tau)m_2)^2 = R_1^2(\phi_1), \\ (m_3 + (\sigma - \tau)m_1 - (\sigma + \tau))^2 + (m_4 + (\sigma - \tau)m_2)^2 = R_2^2(\phi_2), \end{gathered} \tag{7.34}$$

where

$$\begin{gathered} R_1^2(\phi_1) = a_1^2 + b_1^2 + 2a_1 b_1 \cos(\phi_1), \\ |a_1 - b_1| \leq R_1(\phi_1) \leq (a_1 + b_1), \end{gathered} \tag{7.35}$$

and $R_2(\phi_2)$ is given by Eq. (7.32).

7.3 Rational Motions of Planar 6R Closed Chain

In this section, we present an algorithm adopted from Purwar et al. [5] for synthesizing C^2 continuous piecewise rational motions of planar 6R closed chain under the kinematic constraints derived in the previous section.

7.3.1 C^2 Interpolating Rational Motion for Planar 6R Closed Chain

This subsection presents a matrix approach for the following constrained motion interpolation problem:

Given: A set of positions of the coupler link of a planar 6R closed chain in its workspace, the corresponding parameter values $u_i (i = 1 \ldots n)$, lengths of the links given by $a_i, b_i (i = 1, 2)$, and the distance between the two moving pivots and between the two fixed pivots given by h and g, respectively.

Find: A smooth (C^2 or higher) rational motion of the coupler link that interpolates the given positions at the respective parameter values subject to the kinematic constraints of the planar 6R closed chain.

The given positions of the coupler link of a planar 6R closed chain can be given using either Cartesian based parameters $(d_{1i}, d_{2i}, \alpha_i)$ or the joint coordinates of the robot arm. In the case of Cartesian parameters, we use Eq. (2.6) and (7.2) to obtain the elements of the matrix $[M]$; in the case of joint parameters, we directly substitute joint parameters into homogeneous matrix form of the planar robot arm. If the given positions of the coupler are not connectable with a continuous trajectory, then the mechanism has to be disassembled and re-assembled to interpolate the positions. This is commonly known as "assembly mode defect". Recently, Schröcker and Husty [62] presented fast numerical algorithms for determining if two task positions of planar four-bar mechanisms suffer from this defect. The algorithm presented below assumes that the given positions are on the same branch.

In what follows, we present a sketch of the algorithm:

1. Given positions of the coupler are converted to the elements of the displacement matrix; this gives us a set of element vectors, $\mathbf{m}_i; i = 0 \ldots n$. An initial interpolating C^2 B-spline curve $\mathbf{m}(u)$ is constructed[1] in the space of the elements of the displacement matrix using Eq. (7.5).

2. The curve $\mathbf{m}(u)$ in the parameter space should satisfy following geometric constraints (rewritten from Eq. (7.34) and slightly modified from Eq. (7.3)):

[1]Designing a C^2 B-spline curve is a standard scheme in CAGD (see Farin [11], Hoschek and Lasser [14], and Piegl and Tiller [15]).

$$(a_1 - b_1)^2 \leq F_1(u) \leq (a_1 + b_1)^2, \tag{7.36}$$

$$(a_2 - b_2)^2 \leq F_2(u) \leq (a_2 + b_2)^2, \tag{7.37}$$

$$(1.0 - \delta)^2 \leq F_3(u) \leq (1.0 + \delta)^2, \tag{7.38}$$

where

$$F_1(u) = (m_3(u) - (\sigma - \tau)m_1(u) + (\sigma + \tau))^2 + (m_4(u) - (\sigma - \tau)m_2(u))^2,$$
$$F_2(u) = (m_3(u) + (\sigma - \tau)m_1(u) - (\sigma + \tau))^2 + (m_4(u) + (\sigma - \tau)m_2(u))^2,$$

and $F_3(u) = m_1^2(u) + m_2^2(u)$. In Eq. (7.38), δ is a user-defined value that can be chosen as small as desired to approximate the circular constraints. Here, we have modified the form of (7.3) to an inequality constraint for numerical computation.

Since a C^2 B-spline curve, such as $\mathbf{m}(u)$ has a piecewise cubic Bézier representation, it is easy to evaluate the first order derivative of functions $F_i(u); i = 1, 2, 3$ and verify if the geometric constraints given above are satisfied. The solution of following equations yields the extrema of functions $F_i(u); i = 1, 2, 3$:

$$\frac{dF_1(u)}{du} = 0, \quad \frac{dF_2(u)}{du} = 0, \quad \frac{dF_3(u)}{du} = 0. \tag{7.39}$$

If an extremum of any of the functions $F_i(u)$ is outside the corresponding inequality given in Eqs. (7.36), (7.37), or (7.38), the constraints are considered violated. Such a constraint violating extremum (say, at $u = u^*$) is called an extreme point $\mathbf{m}^*$ on the parameter space curve $\mathbf{m}(u)$.

3. If an extreme point is found at $u = u^*$, this point is replaced with a new point $\mathbf{m}(u^*)$ that satisfies the geometric constraints (Eqs. (7.36), (7.37), and (7.38)) and added to the initial set of positions given to be interpolated. A new C^2 B-spline curve is generated that interpolates this new point as well. We also require this new point to be minimally away from the extreme point so as to allow the least amount of change in the shape of the previously generated curve. If the new curve satisfies all the constraints, we stop otherwise we repeat the procedure outlined above.

4. The issue of finding a new point that is minimally away from an extreme point can be turned into a normal distance minimization problem in the parameter space. This problem has been effectively solved by Ravani and Roth [45], who gave a general algebraic method for approximate normal distance calculation between the image curve and a given position in the image space. Bodduluri and McCarthy [46] used their method for finite position synthesis of a spherical four-bar motion. However, our operating space in this chapter being a parameter space rather than an image space, a proper metric for this space has to be defined. We show via a simple derivation (see appendix B) that our choice of a metric for

the space of the elements of the displacement matrix is directly related to the metric used by Ravani and Roth [45] for planar displacements. We note here that the choice of a metric for planar or spatial displacements is a continuing topic of research (see Angeles [64] for the latest on this topic) due to the problem associated with combining translation and rotation in a meaningful way. Our choice of metric in the space of the elements of displacement matrices combines translation with the rotation in such a way that the "distance" between two planar displacements is same as the metric used by Ravani and Roth [45]. In the context of four-bar mechanism synthesis, the method used by Ravani and Roth [45] solves the design problem of determining an image curve that passes through or near a set of given points. The image curve is algebraically given by intersection of two quadric hypersurfaces (constraint surfaces) and is expressed as functions of the design parameters, such as link lengths and the pivot locations. They seek the values of design parameters which reduce the error between the image curve and the given points. The error at each given position is defined as the normal distance to the image curve. They define total error as the sum of squares of each position error. The method is approximate in the sense that the constraint surfaces are approximated by their tangent hyperplane in the vicinity of the desired position. We now outline and conform Ravani and Roth [45]'s method for calculation of a new point in our problem:

Assuming that there exists an extreme point $\mathbf{m}^*$ at $u = u^*$ on the curve

and a new point $\mathbf{m}$ is desired to be inserted to the initial set of given positions at the same parameter $u = u^*$, we define a normal error vector $\mathbf{e} = \mathbf{m} - \mathbf{m}^*$. The new point $\mathbf{m} = (m_1, m_2, m_3, m_4)$ should satisfy the geometric constraints given by Eqs. (7.36), (7.37), and (7.38), rewritten as follows.

$$H_1(\mathbf{m}) : (m_3-(\sigma-\tau)m_1+(\sigma+\tau))^2+(m_4-(\sigma-\tau)m_2)^2-r_1^2 = 0, \quad (7.40)$$

$$H_2(\mathbf{m}) : (m_3+(\sigma-\tau)m_1-(\sigma+\tau))^2+(m_4+(\sigma-\tau)m_2)^2-r_2^2 = 0, \quad (7.41)$$

$$H_3(\mathbf{m}) : m_1^2 + m_2^2 - r_3^2 = 0, \quad (7.42)$$

where we have introduced new variables r_1, r_2, and r_3 that should satisfy following inequalities:

$$\begin{array}{l} |a_1 - b_1| \leq r_1 \leq (a_1 + b_1), \\ |a_2 - b_2| \leq r_2 \leq (a_2 + b_2), \\ |1.0 - \delta| \leq r_3 \leq (1.0 + \delta). \end{array} \quad (7.43)$$

Equations (7.40), (7.41), and (7.42) describe three different quadric hypershells in the parameter space. We seek to minimize the square of the l^2 norm of the error vector $\mathbf{e}$ subject to the condition that $\mathbf{m}$ satisfies these inequalities. This optimization procedure yields the optimal values of variables r_1, r_2 and r_3, which in turn give the new point $\mathbf{m} = \mathbf{e}^* + \mathbf{m}^*$, where $\mathbf{e}^*$ is the optimized normal error vector. This new point $\mathbf{m}$ may not satisfy the kinematic constraints because the constraint surfaces are only approximated in this approach. In that case, this new point $\mathbf{m}$

is set as the new extreme point $\mathbf{m}^*$ and the process described above is repeated.

For faster computation of optimized normal error vector, Ravani and Roth [45] suggest an approximate method by using Taylor series expansion of the hypersurfaces (Eqs. (7.40) (7.41), and (7.42)) in the vicinity of the extreme point $\mathbf{m}^*$:

$$\begin{aligned} 0 &= H_1(\mathbf{m}^*) + \sum_{i=1}^{4} \frac{\partial H_1(\mathbf{m}^*)}{\partial m_i} \Delta m_i, \\ 0 &= H_2(\mathbf{m}^*) + \sum_{i=1}^{4} \frac{\partial H_2(\mathbf{m}^*)}{\partial m_i} \Delta m_i, \\ 0 &= H_3(\mathbf{m}^*) + \sum_{i=1}^{2} 2m_i^* \Delta m_i. \end{aligned} \tag{7.44}$$

These equations can be assembled as follows:

$$\begin{bmatrix} \frac{\partial H_1}{\partial m_1} & \frac{\partial H_1}{\partial m_2} & \frac{\partial H_1}{\partial m_3} & \frac{\partial H_1}{\partial m_4} \\ \frac{\partial H_2}{\partial m_1} & \frac{\partial H_2}{\partial m_2} & \frac{\partial H_2}{\partial m_3} & \frac{\partial H_2}{\partial m_4} \\ 2m_1 & 2m_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta m_1 \\ \Delta m_2 \\ \Delta m_3 \\ \Delta m_4 \end{bmatrix} = \begin{bmatrix} -H_1 \\ -H_2 \\ -H_3 \end{bmatrix}. \tag{7.45}$$

The above can also be written as

$$[J]\mathbf{e} = \mathbf{v}. \tag{7.46}$$

We solve for the normal error vector $\mathbf{e}$ by minimizing the Lagrangian function given as follows:

$$L(\mathbf{e}, \mathbf{a}) = \mathbf{e}^T[Q]\mathbf{e} + \mathbf{a}^T([J]\mathbf{e} - \mathbf{v}), \tag{7.47}$$

where $\mathbf{a} = (a_1, a_2, a_3)$ is a vector of Lagrange multipliers, $\mathbf{e} = (m_1 - m_1^*, m_2 - m_2^*, m_3 - m_3^*, m_4 - m_4^*)$, $\mathbf{v} = (-H_1(\mathbf{m}^*), -H_2(\mathbf{m}^*), -H_3(\mathbf{m}^*))$, and $[Q] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/4 \end{bmatrix}$.
Matrix $[Q]$ comes about from using a proper metric for the parameter space of the elements of the displacement matrix (see appendix B). Putting the condition for a minimum as

$$\frac{\partial L}{\partial e_i} = 0, i = 1, 2, 3, 4 \tag{7.48}$$

where (e_1, e_2, e_3, e_4) are the coordinates of the vector $\mathbf{e}$ and assembling the solution equations, we obtain

$$2\mathbf{g} + [J]^T\mathbf{a} = 0, \tag{7.49}$$

where $\mathbf{g} = (2e_1, 2e_2, e_3/2, e_4/2)$. If we transform the matrix $[J]$ to $[J']$ such that

$$[J'] = \begin{bmatrix} \dfrac{\partial H_1}{\partial m_1} & \dfrac{\partial H_1}{\partial m_2} & 4\dfrac{\partial H_1}{\partial m_3} & 4\dfrac{\partial H_1}{\partial m_4} \\ \dfrac{\partial H_2}{\partial m_1} & \dfrac{\partial H_2}{\partial m_2} & 4\dfrac{\partial H_2}{\partial m_3} & 4\dfrac{\partial H_2}{\partial m_4} \\ 2m_1 & 2m_2 & 0 & 0 \end{bmatrix}, \tag{7.50}$$

then Eq. (7.49) changes to

$$2\mathbf{e} + [J']^T\mathbf{a} = 0. \tag{7.51}$$

Thus, if $\mathbf{e}^*$ designates the solution to the error vector (or, the normal

distance) then it should satisfy the equations

$$\begin{aligned} [J]\mathbf{e}^* &= \mathbf{v}, \\ 2\mathbf{e}^* + [J']^T\mathbf{a} &= 0. \end{aligned} \tag{7.52}$$

Equation (7.52) gives an explicit formula for the solution error vector $\mathbf{e}^*$ as:

$$\mathbf{e}^* = [J']^T([J][J']^T)^{-1}\mathbf{v}. \tag{7.53}$$

Thus, we can determine the variables r_1, r_2, and r_3 by optimizing the function

$$E(\mathbf{m}, r_1, r_2, r_3) = (\mathbf{e}^*)^T\mathbf{e}^*, \tag{7.54}$$

subject to constraints given by Eq. (7.43).

With normal error vector $\mathbf{e}^*$ known, the new point is given by

$$\mathbf{m} = \mathbf{e}^* + \mathbf{m}^*. \tag{7.55}$$

Since the constraint surfaces have been approximated by their tangent hyperplanes in the vicinity of the extreme point $\mathbf{m}^*$, this new point $\mathbf{m}$ may not lie inside the constraint solids given by Eqs. (7.40), (7.41), and (7.42). If the new point does not satisfy the constraints, the newly obtained point $\mathbf{m}$ is set as an extreme point $\mathbf{m}^*$ and the procedure described above is repeated from Eq. (7.44) until a new point $\mathbf{m}$ is obtained that satisfies the kinematic constraints. With this new point added to the set of initial positions, a new C^2 B-spline is generated that interpolates the points. If the new curve detects any further violation of the kinematic

constraints, the optimization process is repeated until no further extreme points are found. This process at the end gives an interpolating motion that satisfies the kinematic constraints. Now, we present the algorithm:

Algorithm 7.3.1

1. Convert given positions of the coupler link into matrix elements $\mathbf{m}_i = (m_{i1}, m_{i2}, m_{i3}, m_{i4})$ using Eqs. (2.6) and (7.2) or, if given in terms of joint angles, by direct substitution into the homogeneous form of the displacement matrix.

2. Current list of points to be interpolated = given points $(\mathbf{m}_i; i = 1 \ldots n)$

3. Construct a C^2 cubic B-spline curve $\mathbf{m}(u)$ that interpolates $\mathbf{m}_i$ at parameter values u_i

4. Evaluate the extrema of $F_i(u); i = 1, 2, 3$ using Eq. (7.39). Repeat Steps (a) to (f) for all extrema.

   (a) Say, an extreme point is found at $u = u^*$.

   (b) Check if $\mathbf{m}(u^*)$ satisfies the kinematic constraints (Eqs. (7.40), (7.41), and (7.42)).

   (c) If yes, the curve is constrained; continue to Step 5. If no, designate $\mathbf{m}(u^*)$ as an extreme point $\mathbf{m}^*$ and continue.

   (d) Find a new point $\mathbf{m}$ (Eqs. (7.44) – (7.55)).

   (e) Check if the new point $\mathbf{m}$ satisfies kinematic constraints (Eqs. (7.40), (7.41), and (7.42)).

i. If yes, continue to next Sub-step (f)

ii. else, set $\mathbf{m}^* = \mathbf{m}$ and repeat from Sub-step (d).

(f) Add $\mathbf{m}$ at $u = u^*$ to the current list of points to be interpolated and go to Step 3.

5. The parameter space curve $\mathbf{m}(u)$ defines a C^2 interpolating piecewise rational motion of degree 3 after substitution into Eq. (7.1).

We have observed that this algorithm always converges as long as the interpolating points are on the same branch. In this algorithm, the B-spline curve is generated using a global interpolation scheme (Piegl and Tiller [15]). In this scheme, although moving one of the interpolating points changes the curve globally, the change in the curve diminishes away from the modification point.

7.3.2 Example

In this section, we present an example to demonstrate the algorithm presented earlier. Table 7.1 gives the elements of the displacement matrix $(m_{i1}, m_{i2}, m_{i3}, m_{i4})$ for five positions of the coupler link of a planar 6R closed chain along with their parameter values. The table also gives the link lengths and the distances between moving and the fixed pivots. The range of the inequality for the kinematic constraints given by Eqs. (7.36), (7.37), (7.38) are shown in Table 7.2.

For the input data given in the Table 7.1, our algorithm takes two iterations to produce a C^2 B-spline motion that satisfies all the kinematic constraints. In the first iteration, four extreme points are detected; out of which all the four

i	$\mathbf{m}_i = (m_{i1}, m_{i2}, m_{i3}, m_{i4})$	u_i
1	$(1.0000, 0, 2.0449, -0.1941)$	0.0
2	$(0.8660, 0.5000, 1.9067, 1.5028)$	2.0
3	$(0.9659, -0.2588, -0.8893, 3.4851)$	5.0
4	$(0.9659, 0.2588, -0.7850, 3.2651)$	7.0
5	$(0.8568, 0.5157, -2.3005, 3.1447)$	10.0

Table 7.1: **Elements of the displacement matrix of the given positions of the coupler link of a planar 6R closed chain**($a_1 = 1.0, b_1 = 3.0, a_2 = 4.0, b_2 = 3.2, g = 6.0, h = 3.6$)

i	Kinematic constraints (F_i)
1	$4.00 \leq F_1 \leq 16.00$
2	$0.64 \leq F_2 \leq 51.84$
3	$0.95 \leq F_3 \leq 1.05$

Table 7.2: **Kinematic constraints of given planar 6R closed chain.** $\delta = 0.05$

violate the approximate rigid body constraint given by Eq. (7.38), and two of them also violate the constraint given by Eq. (7.36). In the second iteration, one extreme point is detected. This is shown partially in Figure 7.1, where a part of the constraint surface parameterized by coordinates m_1, m_2, m_3 is shown. The figure prominently shows the hypersurface given by Eq. (**??**), which is a cylinder perpendicular to $m_1 m_2$ plane. The initial unconstrained B-spline curve is shown by broken line and the initial positions to be interpolated are shown by round filled circles ('$\bullet$') The extreme points detected in the first iteration are shown by the 'star' ('$\star$') symbol, while the lone extreme point detected in the second iteration is shown by a 'delta' ('$\triangle$') symbol. The figure also indicates the parameter and the value of the $F_i(u); i = 1, 3$ func-

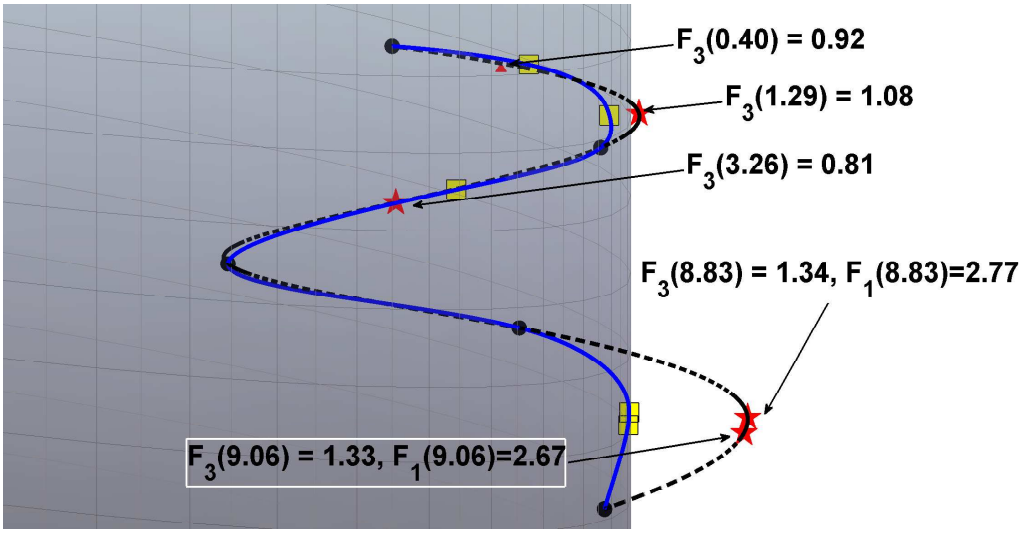

Figure 7.1: Unconstrained and constrained interpolation; kinematic constraint surface in a three-dimensional space parameterized by m_1, m_2, m_3 coordinates: all five extreme points ($\star$) violate circular constraint; algorithm adds five new points ($\square$).

tions ((7.36), and (7.38)) at the extreme points. It can be clearly seen that none of the extreme points are on the hypercylinder, indicating a violation of rigid body constraint, however in this figure it is difficult to see the violation of the constraint given by Eq. (7.36). The algorithm adds five new points (indicated by '$\square$') corresponding to the five extreme points. The new curve lies on or very near (due to a choice of $\delta = 0.05$) to the hypercylinder and satisfies all the constraints. To visualize the violation of the other constraint, we show the intersection of the corresponding four dimensional constraint shells (Eqs.(7.36), (7.37)) with $m_1 = 1$ hyperplane in Figure 7.2. In $m_1 = 1$ hyperplane, Eqs.(7.36) and (7.37) describe two elliptic cylindrical shells. The curve is constrained to lie inside the volume between the boundary surfaces. The fig-

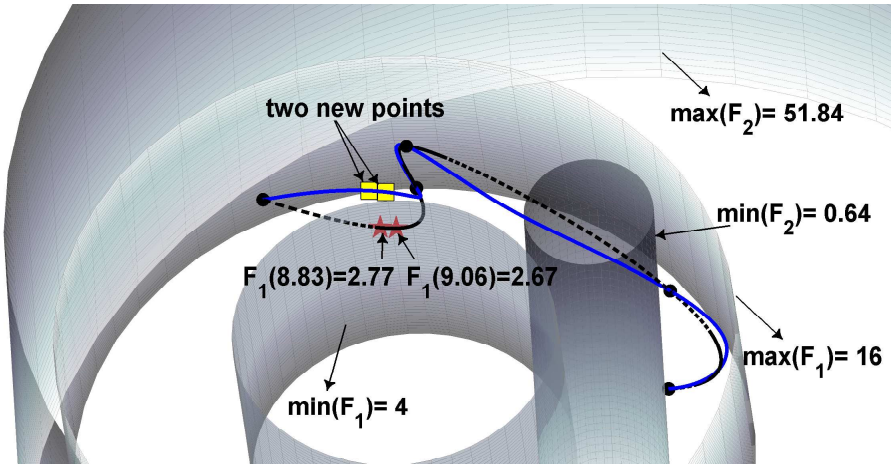

Figure 7.2: Intersection of two constraint shells ($4.00 \leq F_1 \leq 16.00$ and $0.64 \leq F_2 \leq 51.84$) with $m_1 = 1$ hyperplane and the unconstrained and constrained curve: two extreme points with $F_1(8.83) = 2.77$ and $F_1(9.06) = 2.67$ violate the kinematic constraint: $4.00 \leq F_1$

ure shows two extreme points that violate the F_1 constraint (Eq.(7.36)). These two points are clearly seen to be outside the constraint shell bounded by surfaces marked as $\min(F_1)$ and $\max(F_1)$. These are the same two points, which violate the F_1 constraint in first iteration (see Figure 7.1). Also shown are two new points ('□') added by the algorithm and the constrained curve. The final interpolating curve satisfies all the constraints, which in the Cartesian space translates into a C^2 continuous B-spline rational motion of the planar 6R closed chain.

7.4 Matrix Approach to Rational Motions of Planar 4R and 5R Closed Chains

Constraint equations of planar 4R and 5R closed chains can be seen as a special case of the constraint equations of planar 6R closed chain. By transforming the equality relations of 4R and 5R closed chains from Eqs. (7.28) and (7.31) into inequalities, we can apply the same algorithm as presented before to do constrained interpolation for 4R and 5R closed chains as well. We have already seen how to transform the circular constraint into an inequality (Eq. (7.36)). Here we would focus on the other equality constraints.

7.4.1 Planar 4R Closed Chain

We modify the equality constraints from Eq. (7.28) of planar 4R closed chains as follows:

$$(a_1 - \delta_1)^2 \leq F_1(u) \leq (a_1 + \delta_1)^2, \tag{7.56}$$

and

$$(a_2 - \delta_2)^2 \leq F_2(u) \leq (a_2 + \delta_2)^2, \tag{7.57}$$

where δ_1 and δ_2 are user-defined tolerances, and $F_1(u)$ and $F_2(u)$ have been previously defined. Thus, by choosing these tolerances to be as small as possible, the user can use the same algorithm to do constrained interpolation to a desired degree of satisfaction. The new points are added in the same way as described before.

7.4.2 Planar 5R Closed Chain

In case of planar 5R closed chain, we modify the only non-circular equality constraint in Eq. (7.31) to be same as the modified constraint given by Eq. (7.56).

7.5 Matrix Approach to Rational Motion Interpolation of Planar Open Chains

This section presents matrix approach for the following constrained motion interpolation problem:

Given: A set of the positions of the end link of planar 2R, PR, RP, 3R, RRP, RPR, PRR, PRP open chains in their workspace as well as the parameter values $u_i,\ i = 1..n$.

Find: Rational motions of the end link that interpolate through the given positions at the parameter values and satisfy the constraints of given planar open chains.

The given positions of the end link can be specified using either Cartesian based parameters $(d_{1i}, d_{2i}, \alpha_i)$ or the joint coordinates of the robot arm. In the case of Cartesian parameters, we use Eq.(2.6) to convert them into planar quaternions; in the case of joint parameters, we directly substitute joint parameters into homogeneous matrix form of the planar open chain.

We can see from the constraint equations of planar open chains mentioned in previous section 7.2 that they are combinations of the geometric constraints of circle, circular ring, and band. We can apply **Algorithm 4.2** for C^1 piece-

	C^1 Interpolation		C^2 Interpolation	
	(m_1, m_2)	(m_3, m_4)	(m_1, m_2)	(m_3, m_4)
2R	**Alg. 4.2**	**Alg. 4.2**	**Alg. 4.3**	**Alg. 4.3**
PR	**Alg. 4.2**	**Alg. 4.3**	**Alg. 4.3**	**Alg. 4.3**
RP	**Alg. 4.2**	**Alg. 4.3**	**Alg. 4.3**	**Alg. 4.3**
3R	**Alg. 4.2**	**Alg. 4.3**	**Alg. 4.3**	**Alg. 4.3**
RRP	**Alg. 4.2**	**Alg. 4.2, 4.3**	**Alg. 4.3**	**Alg. 4.3**
RPR	**Alg. 4.2**	**Alg. 4.3**	**Alg. 4.3**	**Alg. 4.3**
PRR	**Alg. 4.2**	**Alg. 4.2, 4.3**	**Alg. 4.3**	**Alg. 4.3**
PRP	**Alg. 4.2**	**Alg. 4.3**	**Alg. 4.3**	**Alg. 4.3**

Table 7.3: C^1 **and** C^2 **Interpolation of planar open chains** (For PR robot arm $m_4 = 0$.

wise rational Bézier interpolation on a circle and use **Algorithm 4.3** for C^2 B-spline interpolation within a circular ring and C^2 B-spline interpolation inside a band.

Table 7.3 summarizes algorithm combinations for rational motion interpolation problem under kinematic constrains of planar open chains described in the beginning of this section.

7.6 Conclusions

In this chapter, we have studied the problem of constrained motion interpolation directly in a parametric space defined by the elements of the planar displacement matrix. Kinematic constraints of various planar open and closed chains have been characterized by the algebraic relations among the elements of the displacement matrix. This facilitates the development of the algorithms for generating C^1 and C^2 piecewise rational interpolating motions in a para-

metric space defined by these elements. Several examples have been presented to demonstrate the effectiveness of the algorithms. Future research is to explore the feasibility of this approach for spherical and spatial kinematic chains.

Chapter 8

Conclusions

This dissertation focussed on the constrained Cartesian motion interpolation for planar, spherical, and spatial kinematic chains. It was shown that by quaternion based representation of displacements, the problem of Cartesian motion interpolation could be reduced to that of curve interpolation in quaternion space subject to geometric constraints which are transformed from kinematic constraints of kinematic chains. This dissertation proposed several iterative algorithms for aforementioned constrained Cartesian motion interpolation. It was shown that the algorithm for piecewise rational Bézier interpolation on a circle and the algorithm for Rational B-spline interpolation inside an n-spherical shell could be brought together to do constrained motion interpolation of planar and spherical open chains and ADEPT robot. An algorithm for B-spline interpolation inside intersection of two hyperboloidal shells was developed for constrained motion interpolation of planar and spherical 4R, 5R, and 6R closed chains. For spatial SS open chain, this dissertation showed a numerical algorithm which is similar to that for planar and spherical closed

chains.

In this dissertation, it was also shown that a matrix approach could be used for Cartesian constrained motion interpolation of planar chains. In a parametric space defined by the elements of planar displacement matrix, the problem of Cartesian motion interpolation could be transformed into that of curve interpolation under geometric constraints in terms of the elements of planar displacement matrix. Thus, the numerical algorithms developed previously could be applied. Advantages of directly using the elements of displacement matrix for motion interpolation are that the interpolation process is straightforward and the resulting motion is of lower degree.

Overall, this dissertation has resolved the problem of constrained Cartesian rational motion interpolation for various planar, spherical, and spatial open and closed kinematic chains.

Bibliography

[1] Zhe, J., and Ge, Q. J., 2007. "Computer Aided Synthesis of Piecewise Rational Motions for Planar 2R and 3R Robot Arms". *ASME Journal of Mechanical Design,* **129**(4), pp. 1031–1036.

[2] Jin, Z., and Ge, Q. J., 2007. "Constrained Motion Interpolation for Planar Open Kinematic Chains". *Mechanism and Machine Theory*(accepted for publication).

[3] Jin, Z., and Ge, Q. J., 2007. "Rational Motion Interpolation under Kinematic Constraints of Planar 6R Closed Chain". In ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Paper No. DETC2006-99650, in press.

[4] Purwar, A., Zhe, J., and Ge, Q. J., 2006. "Computer Aided Synthesis of Piecewise Rational Motions for Spherical 2r and 3r Robot Arms". In Proceedings of IDETC/CIE 2006 ASME 2006 International Design Engineering Technical Conferences, Paper No. DETC2006-99650.

[5] Purwar, A., Jin, Z., and Ge, Q. J., 2007. "Rational Motion Interpolation under Kinematic Constraints of Spherical 6R Closed Chains". In ASME

2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Paper No. DETC2007-35727, in press.

[6] Reuleaux, F., 1875. *Theoretical Kinematics: Outline of a Theory of Machines.*

[7] Hunt, K., 1978. *Kinematic Geometry of Mechanisms.* Oxford University Press.

[8] Bottema, O., and Roth, B., 1979. *Theoretical Kinematics.* North Holland, Amsterdam.

[9] Angeles, J., 1988. *Rational Kinematics.* Springer-Verlag.

[10] McCarthy, J. M., 1990. *Introduction to Theoretical Kinematics.* MIT.

[11] Farin, G., 1996. *Curves And Surfaces for Computer-Aided Geometric Design: A Practical Guide*, 4th ed. Academic Press, New York.

[12] Farin, G., 1999. *NURBS: From Projective Geometry to Practical Use.* A. K. Peters, Ltd, Natick, MA, USA.

[13] Farin, G., Hoschek, J., and Kim, M.-S., eds., 2002. *Handbook of Computer Aided Geometric Design.* Elsevier Science, North Holland.

[14] Hoschek, J., and Lasser, D., 1993. *Fundamentals of Computer Aided Geometric Design.* A K Peters.

[15] Piegl, L., and Tiller, W., 1995. *The NURBS Book*. Springer, Berlin.

[16] Gallier, J., 2000. *Curves and surfaces in geometric modeling: theory and algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[17] Shoemake, K., 1985. "Animating rotation with quaternion curves". In Proceedings of the 12th annual conference on Computer graphics and interactive techniques, ACM Press, pp. 245–254.

[18] Pletinckx, D., 1989. "Quaternion calculus as a basic tool in computer graphics". *The Visual Computer,* **5**, pp. 2–13.

[19] Dam, E. B., Koch, M., and Lillholm, M., 1998. Quaternions, interpolation and animation. Technical Report DIKU-TR-98/5, University of Copenhagen.

[20] Duff, T., 1985. "Quaternion splines for animating orientation". In USENIX Association Second Computer Graphics Workshop, pp. 54–62.

[21] Kim, M.-J., Kim, M.-S., and Shin, S. Y., 1995. "A C^2 continuous B-spline quaternion curve interpolating a given sequence of solid orientations". In Proceedings of the Computer Animation, Proceedings of the Computer Animation, IEEE Computer Society, pp. 19–21.

[22] Kim, M. S., and Nam, K. W., 1995. "Interpolating solid orientations with circular blending quaternion curves". *Computer-Aided Design,* **27**(5), pp. 385–398.

[23] Nielson, G., 1993. "Smooth interpolation of orientations". In Computer Animation, Models and Techniques in Computer Animation, Springer, pp. 75–93.

[24] Nielson, G. M., 2004. "Nu-quaternion splines for the smooth interpolation of orientations". *IEEE Transactions on Visualization and Computer Graphics,* **10**(2), pp. 224–229.

[25] Wang, W., and Joe, B., 1993. "Orientation Interpolation in Quaternion Space Using Spherical Biarcs". In Graphics Interface '93, Canadian Information Processing Society, pp. 24–32.

[26] Barr, A. H., Currin, B., Gabriel, S., and Hughes, J. F., 1992. "Smooth interpolation of orientations with angular velocity constraints using quaternions". *Computer Graphics,* **26**(2), pp. 313–320.

[27] Ge, Q. J., and Ravani, B., 1994. "Computer-aided geometric design of motion interpolants". *ASME Journal of Mechanical Design,* **116**(3), pp. 756–762.

[28] Ge, Q. J., and Ravani, B., 1994. "Geometric construction of Bezier motions". *ASME Journal of Mechanical Design,* **116**(3), pp. 749–755.

[29] Juttler, B., and Wagner, M. G., 1996. "Computer-aided design with spatial rational B-spline motions". *ASME Journal of Mechanical Design,* **118**(2), pp. 193–201.

[30] Wagner, M. G., 1994. "A Geometric Approach to Motion Design". Ph.d. dissertation, Technische Universitt Wien.

[31] Purwar, A., and Ge, Q. J., 2005. "On the effect of dual weights in computer aided design of rational motions". *ASME Journal of Mechanical Design,* **127**(5), pp. 967–972.

[32] Röschel, O., 1998. "Rational motion design - a survey". *Computer-Aided Design,* **30**(3), pp. 169–178.

[33] Purwar, A., 2005. "Design and Fine Tuning of One- and Two-Parameter Rational Motion". Ph.d. dissertation, Stony Brook University.

[34] Horsch, T., and Juttler, B., 1998. "Cartesian spline interpolation for industrial robots". *Computer-Aided Design,* **30**(3), pp. 217–224.

[35] Wagner, M., and Ravani, B., 1996. "Computer Aided Design of Robot Trajectories Using Rational Motions". In *Recent Advances in Robot Kinematics*, J. Lenarcic and V. Parenti-Castelli, eds. Kluwer Academic Publishers, pp. 151–158.

[36] Ge, Q. J., and Larochelle, P. M., 1999. "Algebraic motion approximation with nurbs motions and its application to spherical mechanism synthesis". *ASME Journal of Mechanical Design,* **121**(4), pp. 529–532.

[37] Hofer, M., Pottmann, H., and Ravani, B., 2002. "Subdivision algorithms for motion design based on homologous points". In *Advances in Robot*

Kinematics–Theory and Applications, J. Lenarcic and F. Thomas, eds. Kluwer Academic Publishers, pp. 235–244.

[38] Hofer, M., Pottmann, H., and Ravani, B., 2004. "From curve design algorithms to the design of rigid body motions". *Visual Computer,* **20**(5), pp. 279–297.

[39] Hofer, M., and Pottmann, H., 2004. "Energy-minimizing splines in manifolds". *ACM Transactions on Graphics,* **23**(3), pp. 284–293.

[40] Pottmann, H., Hofer, M., and Ravani, B., 2004. "Variational motion design". In *On Advances in Robot Kinematics*, J. Lenarcic and C. Galletti, eds. Kluwer, pp. 361–370.

[41] Pobegailo, A. P., 2003. "Design of motion along parameterized curves using B-splines". *Computer-Aided Design,* **35**(11), pp. 1041–1046.

[42] Meek, D. S., Ong, B. H., and Walton, D. J., 2003. "Constrained interpolation with rational cubics". *Computer Aided Geometric Design,* **20**(5), pp. 253–275.

[43] Gfrerrer, A., 1999. "Rational interpolation on a hypersphere". *Computer Aided Geometric Design,* **16**(1), pp. 21–37.

[44] Wang, W., and Joe, B., 1997. "Interpolation on Quadric Surfaces with Rational Quadratic Spline Curves". *Computer Aided Geometric Design,* **14**(3), pp. 207–230.

[45] Ravani, B., and Roth, B., 1983. "Motion Synthesis Using Kinematic Mappings". *Journal of Mechanisms Transmissions and Automation in Design-Transactions of the Asme,* **105**(3), pp. 460–467.

[46] Bodduluri, R. M. C., and McCarthy, J. M., 1992. "Finite position synthesis using image curve of a spherical four-bar motion". *ASME J. of Mechanical Design,* **114**(1).

[47] Larochelle, P., and McCarthy, J. M., 1994. "Design of spatial 4C mechanisms for rigid body guidance". In Proc. 1994 ASME Mechanisms Conference, pp. 135–142.

[48] Murray, A. P., Pierrot, F., Dauchez, P., and McCarthy, J. M., 1997. "A planar quaternion approach to the kinematic synthesis of a parallel manipulator". *Robotica,* **15**, pp. 361–365.

[49] Perez, A., and McCarthy, J. M., 2005. "Clifford Algebra Exponentials and Planar Linkage Synthesis Equations". *ASME Journal of Mechanical Design,* **127**(5), pp. 931–940.

[50] Fillmore, J. P., 1984. "A note on rotation matrices". *IEEE Computer Graphics & Application,* **4**(2), pp. 30–33.

[51] Ge, Q. J., and Sirchia, M., 1999. "Computer aided geometric design of two-parameter freeform motions". *ASME Journal of Mechanical Design,* **121**(4), pp. 502–506.

[52] Ge, Q. J., 1990. "Kinematics Constraints as Algebraic Manifolds in the Clifford Algebra of Projective Three Space". Ph.d. dissertation, University of California, Irvine.

[53] Jin, Z., and Ge, Q. J., 2006. "Computer Aided Synthesis of Piecewise Rational Motions for Planar 2R and 3R Robot Arms". In Proc. of IDETC/CIE 2006, Paper No. DETC 2006-99649.

[54] Jin, Z., and Ge, Q. J., 2007. "Constrained Motion Interpolation for Planar Open Kinematic Chains". In Proc. of IFToMM World Congress 2007, Paper No. CK 969.

[55] Müller, H. R., 1962. *Sphärische Kinematic*. Deutscher Ver-lag der Wissenschaften, Berlin.

[56] Ge, Q. J., 1990. "Kinematic Constraints as Algebraic Manifolds in the Clifford Algebra of Projective Three Space". Ph.d. dissertation, University of California Irvine.

[57] Forrest, A. R., 1968. "Curves and Surfaces for Computer-Aided Design". Ph.d. dissertation, University of Cambridge.

[58] Piegl, L., 1986. "A geometric investigation of the rational Bezier scheme of computer-aided-design". *Computers in Industry,* **7**(5), pp. 401–410.

[59] Piegl, L., 1989. "Key developments in computer-aided geometric design". *Computer-Aided Design,* **21**(5), pp. 262–274.

[60] Wagner, M. G., 1995. "Planar rational B-spline motions". *Computer-Aided Design,* **27**(2), pp. 129–137.

[61] Ravani, B., and Roth, B., 1984. "Mappings of spatial kinematics". *Journal of Mechanisms Transmissions and Automation in Design-Transactions of the ASME,* **106**(3), pp. 341–347.

[62] Schrcker, H.-P., Husty, M. L., and McCarthy, J. M., 2007. "Kinematic Mapping Based Assembly Mode Evaluation of Planar Four-Bar Mechanisms". *ASME Journal of Mechanical Design,* **129**(9), pp. 924–929.

[63] Schrcker, H.-P., and Husty, M. L., 2007. "Kinematic Mapping Based Assembly Mode Evaluation of Spherical Four-Bar Mechanisms". In IFToMM 2007, The 12th World Congress in Mechanism and Machine Science, J.-P. Merlet and M. Dahan, eds., Vol. 129, pp. 924–929.

[64] Angeles, J., 2006. "Is There a Characteristic Length of a Rigid-Body Displacement?". *Mechanism and Machine Theory,* **41**(8), pp. 884–896.

[65] Hamilton, W. R., 1853. *Lectures on Quaternions.* Hodges Smith & Co., Dublin.

[66] Dai, J., Y. S.-T., and Gu, X.-F., 2007. "Geometric accuracy analysis for discrete surface approximation". *Computer Aided Geometirc Design,* **24**(6), pp. 323–338.

Appendices

Appendix A

Quaternions

Quaternions A quaternion $\mathbf{Q}$ as introduced by W. R. Hamilton [65] is defined as a hypercomplex number depending on four units 1, $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$:

$$\mathbf{Q} = Q_1\mathbf{i} + Q_2\mathbf{j} + Q_3\mathbf{k} + Q_4, \tag{A.1}$$

where Q_l $(l = 1, 2, 3, 4)$ are real numbers, called the components of $\mathbf{Q}$.

The addition of two quaternions is given by

$$\mathbf{G} + \mathbf{Q} = (G_1 + Q_1)\mathbf{i} + (G_2 + Q_2)\mathbf{j} + (G_3 + Q_3)\mathbf{k} + (G_4 + Q_4) \tag{A.2}$$

The multiplication of two quaternions is distributive with respect to summation and is defined by the following rules for the multiplication of the units:

$$\begin{array}{lll} \mathbf{i}^2 = -1, & \mathbf{j}^2 = -1, & \mathbf{k}^2 = -1, \\ \mathbf{ij} = \mathbf{k}, & \mathbf{jk} = \mathbf{i}, & \mathbf{ki} = \mathbf{j}, \\ \mathbf{ik} = -\mathbf{j}, & \mathbf{kj} = -\mathbf{i}, & \mathbf{ji} = -\mathbf{k}. \end{array} \tag{A.3}$$

Hence

$$\mathbf{GQ} = [G^+]\mathbf{Q} = [Q^-]\mathbf{G}, \tag{A.4}$$

where

$$[G^+] = \begin{bmatrix} G_4 & -G_3 & G_2 & G_1 \\ G_3 & G_4 & -G_1 & G_2 \\ -G_2 & G_1 & G_4 & G_3 \\ -G_1 & -G_2 & -G_3 & G_4 \end{bmatrix}, \quad [Q^-] = \begin{bmatrix} Q_4 & Q_3 & -Q_2 & Q_1 \\ -Q_3 & Q_4 & Q_1 & Q_2 \\ Q_2 & -Q_1 & Q_4 & Q_3 \\ -Q_1 & -Q_2 & -Q_3 & Q_4 \end{bmatrix}. \tag{A.5}$$

Quaternion multiplication is associative but not commutative. If $\mathbf{Q} = Q_1\mathbf{i} + Q_2\mathbf{j} + Q_3\mathbf{k} + Q_4$ is a quaternion, the conjugate quaternion $\mathbf{Q}^*$ is defined by $\mathbf{Q}^* = -Q_1\mathbf{i} - Q_2\mathbf{j} - Q_3\mathbf{k} + Q_4$. From quaternion multiplication, it follows that $\mathbf{Q}\mathbf{Q}^* = \mathbf{Q}^*\mathbf{Q} = Q_1^2 + Q_2^2 + Q_3^2 + Q_4^2$, a non-negative number called the norm $N(\mathbf{Q})$ of $\mathbf{Q}$. If $N(\mathbf{Q}) = 1$ then $\mathbf{Q}$ is called a *unit quaternion*. A quaternion with $Q_4 = 0$ is called a *vector quaternion*.

Dual Quaternions A dual quaternion $\hat{\mathbf{Q}}$ is defined as a quaternion whose components are dual numbers:

$$\hat{\mathbf{Q}} = \mathbf{Q} + \epsilon\mathbf{Q}^0 = \hat{Q}_1\mathbf{i} + \hat{Q}_2\mathbf{j} + \hat{Q}_3\mathbf{k} + \hat{Q}_4, \tag{A.6}$$

where $\hat{Q}_l = Q_l + \epsilon Q_l^0$, $l = 1, 2, 3, 4$, Q_l and Q_l^0 being real numbers.

The addition of two dual quaternions is given by

$$\hat{\mathbf{G}} + \hat{\mathbf{Q}} = (\hat{G}_1 + \hat{Q}_1)\mathbf{i} + (\hat{G}_2 + \hat{Q}_2)\mathbf{j} + (\hat{G}_3 + \hat{Q}_3)\mathbf{k} + (\hat{G}_4 + \hat{Q}_4) \tag{A.7}$$

The multiplication of two dual quaternions $\hat{\mathbf{G}}$ and $\hat{\mathbf{Q}}$ goes as follows:

$$\begin{aligned} \hat{\mathbf{G}}\hat{\mathbf{Q}} &= \mathbf{G}\mathbf{Q} + \epsilon(\mathbf{G}\mathbf{Q}^0 + \mathbf{Q}\mathbf{G}^0) \\ &= [\hat{\mathbf{G}}^+]\hat{\mathbf{Q}} = [\hat{\mathbf{Q}}^-]\hat{\mathbf{G}}, \end{aligned} \tag{A.8}$$

where $[\hat{\mathbf{G}}^+] = [\mathbf{G}^+] + \epsilon[\mathbf{G}^{0+}]$ and $[\hat{\mathbf{Q}}^-] = [\mathbf{Z}^-] + \epsilon[\mathbf{Q}^{0-}]$.

Planar Quaternions A planar quaternion $\mathbf{Z}$ is defined as a quaternion whose components are dual numbers:

$$\mathbf{Z} = Z_1\mathbf{i}\epsilon + Z_2\mathbf{j}\epsilon + Z_3\mathbf{k} + Z_4, \tag{A.9}$$

where Z_l, $l = 1, 2, 3, 4$ are real numbers.

The addition of two planar quaternions is given by

$$\mathbf{Y} + \mathbf{Z} = (Y_1 + Z_1)\mathbf{i}\epsilon + (Y_2 + Z_2)\mathbf{j}\epsilon + (Y_3 + Z_3)\mathbf{k} + (Y_4 + Z_4) \tag{A.10}$$

The multiplication of two planar quaternions $\mathbf{Y}$ and $\mathbf{Z}$ goes as follows:

$$\mathbf{YZ} = [Y^+]\mathbf{Z} = [Z^-]\mathbf{Y}, \tag{A.11}$$

where

$$[Y^+] = \begin{bmatrix} Y_4 & -Y_3 & Y_2 & Y_1 \\ Y_3 & Y_4 & -Y_1 & Y_2 \\ 0 & 0 & Y_4 & Y_3 \\ 0 & 0 & -Y_3 & Y_4 \end{bmatrix}, \quad [Z^-] = \begin{bmatrix} Z_4 & Z_3 & -Z_2 & Z_1 \\ -Z_3 & Z_4 & Z_1 & Z_2 \\ 0 & 0 & Z_4 & Z_3 \\ 0 & 0 & -Z_3 & Z_4 \end{bmatrix}. \tag{A.12}$$

Appendix B

Derivation of a Simple Metric in the Parameter Space of the Elements of Displacement Matrix

Normal distance in the image space of planar displacements is given by $\mathbf{Y}^T\mathbf{Y} = Y_1^2 + Y_2^2 + Y_3^2 + Y_4^2$, where $\mathbf{Y} = (Y_1, Y_2, Y_3, Y_4)$ is a planar quaternion (see McCarthy [10]). Using Eqs. (2.7) and (7.2), we obtain:

$$m_1^2 + m_2^2 + \frac{1}{4}m_3^2 + \frac{1}{4}m_4^2 = Y_1^2 + Y_2^2 + Y_3^2 + Y_4^2 \tag{B.1}$$

or, $\mathbf{m}^T[Q]\mathbf{m} = \mathbf{Y}^T\mathbf{Y}$, where $[Q] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/4 \end{pmatrix}$ Thus, to calculate normal distance between two points in the parameter space, expression $\mathbf{m}^T[Q]\mathbf{m}$ should be used.

Appendix C

Frame Independent Kinematic Constraint for Spherical 3R Open Chain

In this appendix, we show that kinematic constraint for spherical 3R robot arm derived earlier are frame independent. From spherical trigonometry we have the following equation (see Figure C.1):

$$\cos(\gamma) = \cos(\beta)\cos(\alpha) - \sin(\beta)\sin(\alpha)\cos(\phi) \tag{C.1}$$

Since the range of $\cos(\phi)$ is $[-1, 1]$, Eq. (C.1) reduces to:

$$\cos(\beta + \alpha) \leq \cos(\gamma) \leq \cos(\beta - \alpha). \tag{C.2}$$

After using double-angle formulae in trigonometry and performing algebra manipulation, inequality (C.2) changes to:

$$\tan^2((\beta - \alpha)/2) \leq \tan^2(\gamma/2) \leq \tan^2((\beta + \alpha)/2). \tag{C.3}$$

Similarly, using double-angle formulae Eq. (C.1) can be changed to:

$$\sin^2(\gamma/2) = -\frac{1}{2}\cos(\beta)\cos(\alpha) + \frac{1}{2}\sin(\beta)\sin(\alpha)\cos(\phi) + \frac{1}{2}, \tag{C.4}$$

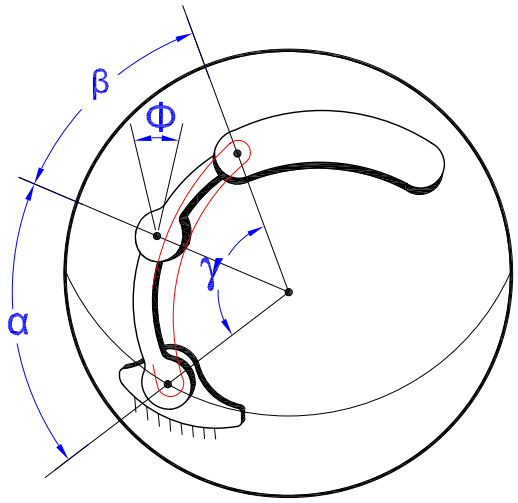

Figure C.1: A Spherical 3R robot arm.

and

$$\cos^2(\gamma/2) = \frac{1}{2}\cos(\beta)\cos(\alpha) - \frac{1}{2}\sin(\beta)\sin(\alpha)\cos(\phi) + \frac{1}{2}. \tag{C.5}$$

We can expand the first equation of (3.60) to:

$$q_1^2 + q_2^2 = \cos(\phi)\sin(\beta)\sin(\alpha)/2 - \cos(\beta)\cos(\alpha)/2 + 1/2. \tag{C.6}$$

Similarly, we can expand the second equation of (3.60) to:

$$q_3^2 + q_4^2 = \cos(\beta)\cos(\alpha)/2 - \cos(\phi)\sin(\beta)\sin(\alpha)/2 + 1/2. \tag{C.7}$$

From Eqs. (C.4), (C.5), (C.6) and (C.7), we find the following relationship:

$$\tan^2(\gamma/2) = \frac{\sin^2(\gamma/2)}{\cos^2(\gamma/2)} = \frac{q_1^2 + q_2^2}{q_3^2 + q_4^2}. \tag{C.8}$$

Eq. (C.3) and Eq. (C.8) together constitute the kinematic constraint of a spherical 3R robot arm.

Appendix D

Convergence Issue in Iterative Algorithm for Motion Interpolation

In this appendix, we discuss convergence issue of iterative algorithms for constrained motion interpolation. Using quaternion based representation of displacements the problem of constrained motion interpolation in Cartesian space is transformed into that of curve interpolation under geometric constraints in quaternion space. Quaternion and planar quaternion space is a linear four dimensional space ($\mathbb{R}^4$) and dual quaternion space is a six dimensional space ($\mathbb{R}^6$). It was shown in this dissertation that the geometric constraints could be a curve, circular ring, n-spherical and hyperboloidal shells. It was also shown that the basic strategy of the iterative algorithms for constrained motion interpolation is to detect extreme points, which violate the geometric constraints most, pull back the extreme points into the geometric constraints, interpolate given points as well as modified points. This procedure will be repeated until there are no points on the interpolating curve that violate the geometric

constraints.

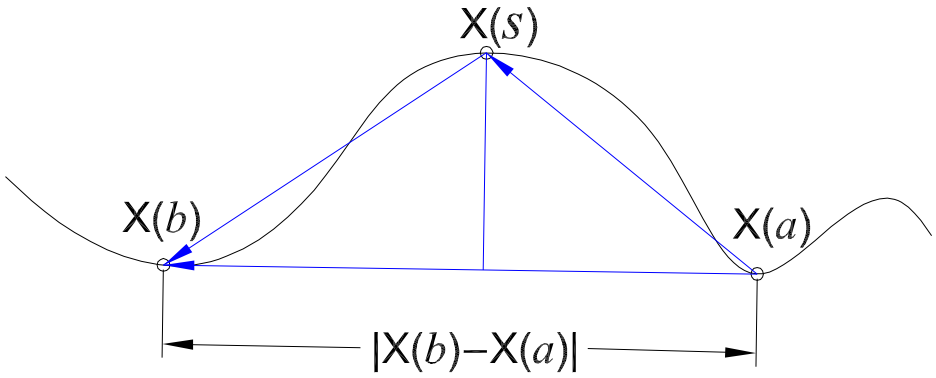

Figure D.1: A parametric curve in $\mathbb{R}^n$.

Dai, Yau, and Gu [66] looked into the properties of an arc length parameterized three dimensional curve. They showed that if arc length parameters satisfy certain conditions and $\mathbf{X}(a)$, $\mathbf{X}(s)$, and $\mathbf{X}(b)$ are three consecutive points on the curve, then the angle $\angle\mathbf{X}(a)\mathbf{X}(s)\mathbf{X}(b)$ is greater than $\pi/2$ (see Figure D.1). In this appendix, we extend their result to $\mathbb{R}^n$ so that it can be applied to the convergence issue of iterative algorithms. *Note that the result is only applicable when the geometric constraint is a curve in quaternion space.*

D.1 Property of Arc Length Parameterized Curve in $\mathbb{R}^n$

Definition D.1.1. Let $\mathbf{X}$ and $\mathbf{Y}$ be two non-zero vectors in $\mathbb{R}^n$, then the angle between $\mathbf{X}$ and $\mathbf{Y}$ is defined by

$$\angle(\mathbf{X}, \mathbf{Y}) = \arccos \frac{\langle \mathbf{X},\ \mathbf{Y} \rangle}{|\mathbf{X}| \cdot |\mathbf{Y}|}, \tag{D.1}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors; $|\cdot|$ denotes the magnitude of a vector and $0 \leq \angle(\mathbf{X}, \mathbf{Y}) \leq \pi$.

Definition D.1.2. Let $\mathbf{e}_i,\ i = 1, 2, \ldots, n$ be the basis vectors in $\mathbb{R}^n$, then the wedge product of two basis vectors are given by

$$\begin{aligned} &\mathbf{e}_i \wedge \mathbf{e}_j = -\mathbf{e}_j \wedge \mathbf{e}_i \quad \text{for } i \neq j, \\ &\mathbf{e}_i \wedge \mathbf{e}_i = 0. \end{aligned} \tag{D.2}$$

Definition D.1.3. Let $\mathbf{e}_i,\ i = 1, 2, \ldots, n$ be the basis vectors of an n-dimensional space, let $\mathbf{X}_j,\ j = 1, 2, \ldots, p$ be vectors in $\mathbb{R}^n$, then a general p-vector is represented by

$$\mathbf{X}_1 \wedge \ldots \wedge \mathbf{X}_p = \sum_{i_1 < \cdots < i_p} (-1)^N M^{i_1 i_2 \cdots i_p} \mathbf{e}_{i_1} \wedge \cdots \wedge \mathbf{e}_{i_p}, \tag{D.3}$$

where $M^{i_1 i_2 \cdots i_p}$ is a $p \times p$ minor of the $p \times n$ matrix formed from the components of $\mathbf{X}_j,\ j = 1, \ldots, p$, and N is given by

$$N = (i_1 - 1) + (i_2 - 2) + \cdots + (i_p - p). \tag{D.4}$$

Remark D.1.1. Let $\mathbf{x}$ and $\mathbf{y}$ be two unit vectors in $\mathbb{R}^n$, then the following equation holds

$$\langle \mathbf{x},\ \mathbf{y}\rangle^2 + |\mathbf{x} \wedge \mathbf{y}|^2 = 1. \tag{D.5}$$

Remark D.1.2. Let $\mathbf{X}$ and $\mathbf{Y}$ be two non-zero vectors in $\mathbb{R}^n$, then the following equation holds:

$$\sin \angle(\mathbf{X}, \mathbf{Y}) = \frac{|\mathbf{X} \wedge \mathbf{Y}|}{|\mathbf{X}| \cdot |\mathbf{Y}|}. \tag{D.6}$$

Definition D.1.4. Let $\mathbf{X}(t)$ be a general parameterized continuous curve in $\mathbb{R}^n$, then the arc length parameter s is defined to be

$$s(t) = \int_0^t \langle d\mathbf{X}/dt, d\mathbf{X}/dt\rangle^{1/2} dt. \tag{D.7}$$

Definition D.1.5. Let $\mathbf{X}(s)$ be an arc length parameterized smooth curve $\mathbb{R}^n$, then we define the tangent vector $\mathbf{T}$ and the total curvature κ as

$$\mathbf{T} = d\mathbf{X}/ds = \mathbf{X}', \tag{D.8}$$

and

$$\kappa = |d^2\mathbf{X}/ds^2| = \mathbf{X}''. \tag{D.9}$$

Remark D.1.3. The magnitude of $\mathbf{T}$ is seen to be 1 by the computation

$$\langle d\mathbf{X}, d\mathbf{X}\rangle = ds^2 \Longrightarrow \langle \mathbf{T}, \mathbf{T}\rangle = \langle \frac{d\mathbf{X}}{ds}, \frac{d\mathbf{X}}{ds}\rangle = 1. \tag{D.10}$$

Remark D.1.4. $\mathbf{X}'(s)$ and $\mathbf{X}''(s)$ are orthogonal.

$$\langle \mathbf{T}, \mathbf{T}\rangle = 1 \Longrightarrow \langle \mathbf{T}, \mathbf{T}'\rangle = 0. \tag{D.11}$$

Lemma D.1.1. *Let $\boldsymbol{X}(s)$ be an arc length parameterized smooth spatial curve in $\mathbb{R}^n$ with curvature bound κ_m, and a, s, b satisfy $0 \le a \le s \le b \le \pi/\kappa_m$, then the following estimate holds*

$$\angle(\boldsymbol{X}(b) - \boldsymbol{X}(s), \boldsymbol{X}(s) - \boldsymbol{X}(a)) \le \pi/2. \tag{D.12}$$

Proof. Consider function $f(s) = \langle \mathbf{X}'(s), \mathbf{X}'(0) \rangle$, then since $\mathbf{X}'(s)$ and $\mathbf{X}''(s)$ are orthogonal, the inner product $\langle \mathbf{X}''(s), \langle \mathbf{X}'(0), \mathbf{X}'(s) \rangle \mathbf{X}'(s) \rangle$ is 0.

$$f'(s) = \langle \mathbf{X}''(s), \mathbf{X}'(0) \rangle = \langle \mathbf{X}''(s), \mathbf{X}'(0) - \langle \mathbf{X}'(0), \mathbf{X}'(s) \rangle \mathbf{X}'(s) \rangle. \tag{D.13}$$

Since $|\mathbf{X}'(s)| = 1$ and the inner product is linear, we obtain the magnitude of the second term of above equation to be

$$\begin{aligned} & |\mathbf{X}'(0) - \langle \mathbf{X}'(0), \mathbf{X}'(s) \rangle \mathbf{X}'(s)| \\ = \; & \sqrt{\langle \mathbf{X}'(0) - \langle \mathbf{X}'(0), \mathbf{X}'(s) \rangle \mathbf{X}'(s), \mathbf{X}'(0) - \langle \mathbf{X}'(0), \mathbf{X}'(s) \rangle \mathbf{X}'(s) \rangle} \\ = \; & \sqrt{\langle \mathbf{X}'(0), \mathbf{X}'(0) \rangle - 2\langle \mathbf{X}'(0), \mathbf{X}'(s) \rangle^2 + \langle \mathbf{X}'(0), \mathbf{X}'(s) \rangle \langle \mathbf{X}'(s), \mathbf{X}'(s) \rangle} \\ = \; & \sqrt{1 - \langle \mathbf{X}'(0), \mathbf{X}'(s) \rangle^2} \\ = \; & \sqrt{1 - f^2(s)} \end{aligned} \tag{D.14}$$

we take into account the property $|\mathbf{X}''(s)| \le \kappa_m$, then Eq. (D.13) satisfies $|f'(s)| \le \kappa_m \sqrt{1 - f^2(s)}$, therefore $f'(s) \ge -\kappa_m \sqrt{1 - f^2(s)}$, furthermore,

$$\begin{gathered} -\frac{f'(s)}{\sqrt{1 - f^2(s)}} \le \kappa_m, \\ \frac{\partial}{\partial s}(\arccos f(s)) \le \kappa_m, \\ f(s) \ge \cos(\kappa_m s), \quad s \in [0, \pi/\kappa_m]. \end{gathered} \tag{D.15}$$

$$
\begin{aligned}
& \langle \mathbf{X}(b) - \mathbf{X}(s), \mathbf{X}(s) - \mathbf{X}(a) \rangle \qquad \text{(D.16)} \\
= \; & \langle \int_s^b \mathbf{X}'(u)du, \; \int_a^s \mathbf{X}'(v)dv \rangle \\
= \; & \int_a^s \int_s^b \langle \mathbf{X}'(u), \; \mathbf{X}'(v) \rangle dudv \\
\geq \; & \int_a^s \int_s^b \cos \kappa_m (u - v) dudv \\
= \; & \cos \kappa_m (b - s) + \cos \kappa_m (s - a) - \cos \kappa_m (b - a) - 1 \\
= \; & 2 \cos \frac{\kappa_m (b - a)}{2} (\cos \frac{\kappa_m (b + a - 2s)}{2} - \cos \frac{\kappa_m (b - a)}{2}) \\
\geq \; & 0.
\end{aligned}
$$

Consider the definition of the inner product along with the result of Eqs. (D.16) we can prove the **Lemma D.1.1**.

□

Theorem D.1.2. *Let $\boldsymbol{X}(s)$ be an arc length parameterized smooth spatial curve in $\mathbb{R}^n$ with curvature bound κ_m, and a, s, b satisfy $0 \leq a \leq s \leq b \leq \pi/\kappa_m$, then the orthogonal distance from the point on the curve $\boldsymbol{X}(s)$ to the line passing through $\boldsymbol{X}(a)$ and $\boldsymbol{X}(b)$ is less than the magnitude of $\boldsymbol{X}(b) - \boldsymbol{X}(a)$.*

Proof. Orthogonal distance from the point $\mathbf{X}(s)$ to the line passing through

$\mathbf{X}(a)$ and $\mathbf{X}(b)$ is less than $|\mathbf{X}(s) - \mathbf{X}(a)|$ and $|\mathbf{X}(b) - \mathbf{X}(s)|$.

$$\begin{aligned} & \langle \mathbf{X}(b) - \mathbf{X}(a), \mathbf{X}(b) - \mathbf{X}(a) \rangle \qquad \text{(D.17)} \\ = \; & \langle (\mathbf{X}(b) - \mathbf{X}(s)) + (\mathbf{X}(s) - \mathbf{X}(a)), (\mathbf{X}(b) - \mathbf{X}(s)) + (\mathbf{X}(s) - \mathbf{X}(a)) \rangle \\ = \; & |\mathbf{X}(b) - \mathbf{X}(s)|^2 + |\mathbf{X}(s) - \mathbf{X}(a)|^2 + 2\langle \mathbf{X}(b) - \mathbf{X}(s), \mathbf{X}(s) - \mathbf{X}(a) \rangle \\ \geq \; & |\mathbf{X}(b) - \mathbf{X}(s)|^2, \; |\mathbf{X}(s) - \mathbf{X}(a)|^2, \end{aligned}$$

we use the result of **Lemma D.1.1**, which is $\langle \mathbf{X}(b) - \mathbf{X}(s), \mathbf{X}(s) - \mathbf{X}(a) \rangle \geq 0$, in above derivation.

Consider the result of Eq. (D.17) we can conclude that the orthogonal distance from the point on the curve $\mathbf{X}(s)$ to the line passing through $\mathbf{X}(a)$ and $\mathbf{X}(b)$ is less than the magnitude of $\mathbf{X}(b) - \mathbf{X}(a)$. □

D.2 Convergence Issue in Iterative Algorithm for Motion Interpolation

According to the **Theorem D.1.2** we can conclude that the maximum of $|\mathbf{X}(b) - \mathbf{X}(a)|$ is the upper-bound of the deviation from a parameterized curve $\mathbf{X}(s)$ to the linear interpolation of points on that curve. Thus, if the upper-bound is within the tolerance δ, we can assume the linear interpolation approximates the curve $\mathbf{X}(s)$.

$$|\mathbf{X}(b) - \mathbf{X}(a)|_{max} \leq \delta. \qquad \text{(D.18)}$$

Furthermore, due to the standard approximation theory, B-spline interpolation converges to linear interpolation, so we can conclude that B-spline interpola-

tion converges to the parameterized curve $\mathbf{X}(s)$ as long as the upper-bound condition Eq.(D.18) is satisfied.