Stony Brook University

Task Driven Design of Mechanisms and Robotic Systems using Kinematic Mapping and Fourier Schemes

A Dissertation Presented
by

Xiangyun Li

to

The Graduate School
in Partial Fulfillment of the
Requirements
for the Degree of

Doctor of Philosophy
in
Mechanical Engineering

Stony Brook University
May 2014

Stony Brook University
The Graduate School
Xiangyun Li
We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree,
hereby recommend acceptance of this dissertation.

Dr. Q. Jeffrey Ge, Advisor
Professor, Mechanical Engineering Department

Dr. Anurag Purwar, Co-advisor,
Research Associate Professor, Mechanical Engineering Department

Dr. Lei Zuo, Chair of Thesis Committee,
Associate Professor, Mechanical Engineering Department

Dr. Hai-Jun Su, Outside Member,
Assistant Professor, Mechanical and Aerospace Engineering Department,
Ohio State University

This dissertation is accepted by the Graduate School.

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Task Driven Design of Mechanisms and Robotic Systems using Kinematic Mapping and Fourier Schemes

by

Xiangyun Li

Doctor of Philosophy

in

Mechanical Engineering

Stony Brook University

2014

This dissertation deals with the problem of task driven design of mechanisms and robotic systems via Fourier and Kinematic Mapping schemes. Task driven design requires that the synthesis process be initiated from the task motion itself rather than from a specific mechanism or robotic system. In other words, extrinsic or intrinsic characteristics of task path or motion is considered as the driving force to the synthesis of mechanisms and robotics systems.

Kinematic Mapping approach is applied in the framework of task driven design to synthesize multi-degrees-of-freedom planar and spatial manipulators in a unified way. We present novel, unified, and simultaneous type and dimensional synthesis approaches respectively to planar and spatial parallel manipulator synthesis by using kinematic mapping, surface fitting, and least squares techniques. Novelty of our approach lies in linearization of a highly non-linear problem and the fact that the nature of the given motion or displacement drives the synthesis process without assuming leg topology or their geometry.

For analysis and simulation of four-bar motion, a unified and efficient algorithm is given through finding the parametrization of the planar four-bar motion in image space represented as intersection curves of two constraint manifolds.

In the field of Computational Shape Analysis it is routine to process and simplify shapes before comparisons are made. The simplified representation of shapes is called shape descriptor, which is the intrinsic characteristic or the signature of the shape. In this dissertation, Fourier transform is employed to analyze the task path or motion and to obtain their signatures, termed as Fourier descriptors, in frequency domain. Therefore, we use Fourier descriptors to address the problem of path and motion synthesis of planar mechanisms. For motion synthesis problem, a given motion is represented by two finite harmonic series, one for translational component of the motion and the other for rotational component. It is shown that there is a simple linear relationship between harmonic content of the rotational motion and that of the translational motion for a planar four-bar linkage. For path synthesis problem, we present an algorithm resolving parameterization issue that has been often ignored in the past research. This approach has the advantage of unifying a variety of parameterizations into a unique one based upon the inherent property of the path.

To my dear parents, Hongying Li and Gang Li, for their patience and dedication in my raising and educating of the past 26 years

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGEMENTS

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

I would like to thank my committee members Professor Anurag Purwar, Professor Lei Zuo and Professor Hai-Jun Su for taking their precious time to attend my presentation as well as review this manuscript.

My thanks also go to my lab mates Jun Wu, Ping Zhao, Po-Yu Chuang and Xin Ge for their suggestions and friendship; and my department staff Mayra Santiago and Melissa Castelbuono for their valuable assistance.

Also, I gratefully acknowledge the financial support of this work by the National Science Foundation under grant number CMMI-0856594 as well as by National Natural Science Foundation of China (Oversea Scholar Research Collaboration Grant No.50728503).

In the end, I would like to especially thank my family, for more than 20 years of raising and education from my parents.

Chapter 1

Introduction and Background

In the dissertation, the synthesis and analysis of planar and spatial mechanisms using Kinematic Mapping and Fourier based design approaches is investigated. For robotics system of multiple degrees-of-freedom, a geometric constraint based synthesis theory using Kinematic Mapping is advocated to bridge the gap between type and dimensional synthesis; Owing to its wide applications in shape analysis, Fourier approach is naturally applied to solving the path and motion synthesis problem of one degree-of-freedom planar mechanism. In what follows, a general overview on background and the existing work is presented followed by main composition of this dissertation.

Kinematics is the study of classical mechanics which describes the motion of points, bodies (objects) and systems of bodies (groups of objects) without consideration of the causes of motion. There exists sizeable amount of literature (Reuleaux [1], Hunt [2], Phillips [3], Bottema and Roth [4], Erdman and Sandor [5], McCarthy [6, 7],Erdman [8]) that have treated kinematics as a fundamental science in its own right. Following previous seminal work in

development of Kinematics theory and technique, much of the current work of kinematics deal with synthesis of mechanisms, with focus on the determination of mechanism types (type synthesis) and/or their link dimensions (dimensional synthesis). Type synthesis is also known as number synthesis, structural synthesis, systematics, classification and enumeration, and census of linkages. The goal is to come up with the most appropriate mechanism type for the specified motion requirement. Once a mechanism type is determined, the next step is to determine the dimensions of the mechanism such as the lengths of the links so that the output motion of the mechanism best matches the specified motion. Dimensional synthesis approaches the problem of determination of kinematic dimensions (link lengths, offsets, etc.) of the mechanism to satisfy the required motion characteristics. The choice of the method depends largely on the type of problem to be solved. The problems can be classified as motion generation, path generation and function generation.

The dimensional synthesis of linkages is an extensively researched subject with many textbooks. The synthesis equations can be systematically derived with various mathematical formulations such as vector loop closure equations [5], quaternion/dual quaternion [9], homogeneous matrices [10]. Recently Su and McCarthy [11] proposed an algebraic curve/surface formulation for spatial open chain synthesis. By exploiting the intrinsic geometry of a specific mechanism type, this formulation leads to a polynomial system with relatively low complexity compared with general homogeneous matrix and dual quaternion formulation. The solution to such a polynomial system in the context of mech-

anism design is a well studied topic [12, 13] and the solution techniques include iterative optimization [14], exact analytical methods [11, 15] and continuation (homotopy) method [10].

Kinematic Mapping is among the popular approaches to mechanism synthesis, which was proposed by Blaschke [16] and Grunwald [17] almost a century ago, and was dug into by Ravani and Roth [18, 19] for motion approximation. More modern application can be found in the formative texts of Bottema and Roth [4] and McCarthy [20]. In the kinematic mapping approach to kinematic synthesis, both planar and spherical displacements in Cartesian Space can be mapped into points in a three-dimensional projective space (called Image Space of Planar or Spherical Kinematics), while workspace constraints of a mechanism map into algebraic manifolds in the same space. In this way, a single degree of freedom motion of a planar or spherical mechanism is represented by the intersection curve of two algebraic manifolds. The problem of motion approximation is transformed into a algebraic curve fitting problem in the image space, where various methods in approximation theory may be applied. This includes the definition of the approximation error (called structural error) in the image space, formulation of a least squares problem and application of appropriate numerical methods to find values of the design variables for minimization of the error. Following Ravani and Roth's kinematic mapping approach for mechanism synthesis, further research has been done by Bodduluri and McCarthy [21], Bodduluri [22], Larochelle [23, 24], Ge and Larochelle [25], Husty et al. [26], and more recently by Wu et al. [27]. Further-

more, kinematic mapping approach has also been widely used for synthesis and analysis of parallel robots. Hayes and colleagues [28, 29] presented a unified treatment for developing kinematic constraint equations of three legged planar platforms possessing three degrees of freedom. Murray et al. [30] presented a technique for designing planar manipulators with platform capable of reaching any number of desired poses, which is connected to ground by three RPR chains. Brunnthaler et al. [31] used kinematic mapping to solve the problem of designing a spherical four-bar mechanism that interpolates a coupler through five given orientations. Venkataramanujam and Larochelle [32] employed the approach of a parameterized constraint manifold and nonlinear optimization to synthesize spherical open and closed chain for approximating task motion. Those aforementioned methods involve highly nonlinear optimization or leg topologies to be specified beforehand. In addition, dimensions of legs are not fully considered resulting in manipulator taking up a large amount of space. Later, Dou [33] presented an algorithm to conduct optimal deign of 3-RRR PPMs and Purwar et al. [34] have recently presented an intuitive, visual synthesis approach for PPMs.

In the context of our task driven design, the emphasis is on extracting from the task motion itself the useful information from which the design process can be unfolded, different from the traditional mechanism-dominated design philosophy. With Kinematic Mapping, the task motion is converted to task curve and workspace of different mechanisms to constraint manifolds in quaternion space. Depending on the shape of task curve, we seek to find a best mechanis-

m type with its associated constraint manifolds fitting the task curve as close as possible, and further determine its dimensional parameters. In this dissertation, we first studies the rigid body guidance problem for 3-DOF planar parallel manipulators (PPM) with three-triad assembly. We present a novel, unified, and simultaneous type and dimensional synthesis approach to planar parallel manipulator synthesis by using kinematic mapping, surface fitting, and least squares techniques. Novelty of our approach lies in linearization of a highly non-linear problem and the fact that the nature of the given motion or displacement drives the synthesis process without assuming triad topology or their geometry. The kinematic constraints associated with planar RR-, PR- and RP-dyads correspond to a single quadric in Image Space, while that of each of the six planar triads (RRR, RPR, PRR, PPR, RRP and RPP) map to a pair of quadrics and the space between them. Moreover, the quadrics associated with RRR- and RPR-triads are of the same type as that of RR dyads, of PRR- and PPR-triads as that of PR-, and RRP- and RPP-triads as that of RP-dyad. This simplification nicely extends a dyad synthesis problem to a triad synthesis one. The problem is formulated as the least-squares error minimization problem to find a family of quadrics that best fit the image points of task displacements. The fitting error corresponding to each single quadric of the family is regarded as variation (thickness) of that quadric, which turns that quadric into a pair of quadrics. Hence, three dyads with minimal surface fitting errors can be converted to three triads in the Cartesian Space. Building upon our work of planar parallel manipulator, we further investigate the

problem of designing spatial mechanisms by studying the constrained motion of a rigid body such that one or more of its points stays on a sphere or a plane. By extending Innocenti's spatial Burmester problem [35] from purely spherical constraints to include any combination of spherical and planar constraints for seven position synthesis of a spatial rigid body. The dissertation provides a unified formulation using homogeneous coordinates and dual quaternions for simultaneous dimensional and type synthesis of spatial platform linkages and manipulators involving spherical and planar constraints.

In addition to the aforementioned research efforts on synthesis theories and algorithms for mechanisms and robotics systems, Many softwares have also been developed to implement those theories and algorithms, including LINCAGES [36, 37], Sphinx [38] and SPADES[39], as well as Synthetica [40], which firstly aims at synthesis of spatial linkages. All of these mechanism design software focus on the dimensional synthesis with specified mechanism topologies. Upon synthesis, motion animation is implemented to analyze the motion generated by the synthesized mechanism. Traditional way of animating the coupler motion of a four-bar mechanism is based on loop-closure equation method (see Norton [41]), which is used to find the coupler point trajectory and coupler angle when the input link rotates. However, this method has disadvantage of writing different versions of code for different types of four-bar mechanisms. In this dissertation, we present a unified and efficient algorithm to simulate the coupler motion of four-bar mechanisms of all six types constructed by R- and P- joints.The algorithm again utilizes Kinematic Mapping to find

the parametrization of the intersection curves of algebraic manifolds of dyads in Image Space, which is converted to the parametrization of the planar motion in cartesian space.

The other approach that has received increasing attention lately is the use of Fourier Descriptors for linkage synthesis. This idea was first explored by Freudenstein [42] in the context of function generation. The research was followed by Funabashi [43], Farhang et al. [44, 45], Chu and Cao [46], and McGarva [47, 48], and Ullah and Kota [49], Nie and Krovi [50], and Wu et al. [51]. Lately, Chu and Sun [52, 53, 54] have extended Fourier Descriptor based method to the synthesis of spherical and spatial linkages. While the path of a coupler point depends on the choice of the coupler point, one may extract a subset of Fourier descriptors of the path in such a way that they depend only on the linkage but not on the choice of the coupler point. This means that for each four bar linkage, one only need to store one set of Fourier descriptors for all coupler curves. Chu and Cao [46] and Chu and Wang [55] made this key observation and achieved significant reduction in the size of the database for numerical atlas using only four design parameters associated with a four-bar linkage. Instead of searching a matching linkage directly from a database, Ullah and Kota [49] sought to find the solution by numerical search method. They also observed that the Fourier formulation allows the decoupling of the nine design variables involved in path generation and this has led to the reduction of the dimension of the search space from nine to five. Recently, Wu et al. [51] further reduced the search dimension from five

to four. Xie and Chen [56] were the first to extend Fourier Descriptor method to the image space of kinematic mapping to solve the whole cycle motion generation problem in four bar linkage synthesis. In their work, image curve of a desired motion was indexed with Fourier Descriptors which were used to be matched with those of four-bar coupler motion. Neural networks were used to establish the relationship between Fourier Descriptors and dimensions of a four-bar linkage.

While Kinematic Mapping method enables us to accomplish task driven design by directly matching the task motion with the workspace of candidate mechanisms, Fourier approach works in an indirect way that captures the characteristic of task path or motion and that of workspace of mechanisms, and then carries out the matching process. As for motion synthesis, this dissertation tackles the classical problem of dimensional synthesis of planar four-bar linkages for motion generation. Using Fourier Descriptors, a given motion is represented by two finite harmonic series, one for translational component of the motion and the other for rotational component. It is shown that there is a simple linear relationship between harmonic content of the rotational motion and that of the translational motion for a planar four-bar linkage. Furthermore, it is shown that the rotational component can be used to identify the initial angle and the link ratios of a four-bar linkage. The rest of the design parameters of a four-bar linkage such as locations of the fixed and moving pivots can be obtained from the translational component of the given motion. This leads naturally to a decomposed design space for four-bar motion synthesis for

approximate motion generation.

Much as the Fourier-based approach has been extensively used to synthesize mechanism for path and motion generation, it has a limitation in such application: dependency on parametrization (timing) of path or motion. Under different forms of parametrization, Fourier Descriptors of the same path or motion have different values. In addition, parametrization of task path or motion is not necessarily consistent with that of mechanism path or motion. Different forms of parametrization corresponding to the same task path or motion would lead to generated mechanisms of different dimensions. As a consequence, it is highly possible that a best solution of mechanism is missed. Nie and Krovi [50] noticed this problem and skillfully utilized it to render smallest number of harmonic components for synthesizing coupled multi-link serial open chain with smallest number of links. Vasiliu and Yannou [57] also mentioned parametrization issue in their paper but in effect handled the problem of different parameter distributions corresponding to different sets of sampling points under a given parametrization, which is a sampling-independent method. In this dissertation, we use arc-length parametrization to identify both the task path and four-bar path . Arc-length parametrization is based on the inherent property of curve: arc length, thus allowing us to compare the Fourier Descriptors of task path and four-bar path under the same parametrization.

The rest of the dissertation is organized as follows. Chapter 2 deals with simultaneous type synthesis and dimensional optimization of planar parallel manipulator using algebraic fitting of a family of quadrics. Spatial platform

linkage is studied in Chapter 3 from the viewpoint of extraction of spherical or planar constraint from a given set of spatial displacements. Chapter 4 introduces a unified algorithm for analysis and simulation of planar four-bar motions defined with r- and p- joints. Chapter 5 presents a Fourier Descriptor based approach to design space decomposition for planar motion approximation. Chapter 6 studies the problem of path synthesis of Mechanism using parametrization-independent non-uniform Fourier approach. Finally, We draw conclusion remarks in chapter 7.

Chapter 2

A Task Driven Approach to Simultaneous Type Synthesis and Dimensional Optimization of Planar Parallel Manipulator Using Algebraic Fitting of a family of Quadrics

A parallel manipulator is a relatively high performance, accurate and stiff mechanism that consists of a moving platform connected to a base platform by several legs. Merlet [58] defines a parallel manipulator as a closed-loop mechanism in which the end-effector (mobile platform) is connected to the base by at least two independent kinematic chains. Early in the history of parallel mechanism designs, Gough [59] invented the first variable-length octahedral hexapod and Stewart [60] presented a design of a flight simulator based upon a 6-DOF parallel platform. Their work was followed by a great deal of subsequent developments in the field of parallel mechanisms.

In this chapter, we extend our recent work in motion approximation by a planar 4R linkage [61]. In a companion paper (Ge et al. [62]), we presented a

unified approach to type and dimension synthesis of planar four bars consisting of any type of possible dyads. This approach is realized by formulating a least squares problem using SVD (singular value decomposition) to extract both the dimensions and dyad topologies of a four-bar linkage. The general idea is to first fit the given motion task to a pencil of intersecting quadrics in the image space, and then impose the kinematic constraints associated with planar dyads to extract the dyad topology and the dimensions. As for parallel manipulator consisting of triads, we find that there are simple relationships between geometric constraints of dyads and triads – dyads of RR-type correspond to triads of RRR- and RPR-type, of PR-type to PRR- and PPR-type, and of RP-type to RRP- and RPP-type. This observation leads to simplification in extending single constraint surface of dyads to its thin-shell version, namely, a pair of constraint surfaces for triads. Furthermore, we choose three pairs of constraint surfaces with minimal fitting errors, each representing a type of triad, to construct a 3-DOF manipulator, leading to optimum dimension for each triad leg, which are assembled together to form a PPM.

Rest of this chapter is organized as follows: in section 2.1, we present qualitative arguments on the geometric constraints of the dyads and triads. In section 2.2, we discuss constraint manifolds of dyads and triads. Section 2.3 presents a unifying representation for dyad and triad constraints and give rise to a generalized manifold (called, G-manifold). In section 2.4, we show how given image points can be fitted to a pencil of quadrics in a least-squares sense via SVD and then by solving a Lagrange multiplier minimization problem,

we can select best fit triads. Finally, we present inverse computation for the optimal dimensions of the triads in section 2.5 and an example and discussion in section 2.6.

2.1 Geometric Constraints of Dyads and Triads

In this section, we discuss simple geometric constraints that can be realized by three kinds of dyads: RR, PR and RP. Then, these dyad geometric constraints are extended to those defining triads, thus reflecting the similarity between corresponding dyads and triads.

We discuss these constraints in the context of task-driven synthesis of mechanisms (Wu et al. [63]). The basic idea is to acquire geometric constraints from the motion itself and then match it with a mechanical system. For example, some of the constraints can be that a certain point of a moving object can trace a circle, a circular arc, an ellipse or a straight line. Each constraint can be realized by a certain mechanism, such as circle by crank and circular arc by rocker in crank-rocker mechanisms, or straight line by slider in crank-slider mechanisms. Starting from the motion and obtaining those constraints hidden inside can simultaneously give us the type of mechanisms that can be used to best approximate the motion as well as its dimensions. In that regards, we actually *compute* both the type and the dimensions.

Geometric constraint of RR, PR and RP Dyads Fig. 2.1 shows the kinematic diagram for three dyads that are commonly found in four-bar link-

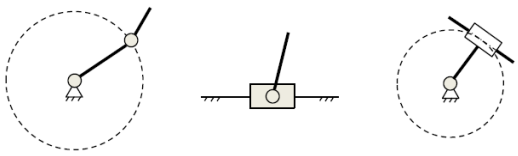

Figure 2.1: RR, PR and RP dyads and their constraints

ages. From left to right, they are RR, PR and RP. For RR, there is one point on the end-effector that lies on a circle; for PR, there is one point on the end-effector that lies on a straight line; for RP, there is a line on the end-effector that is always tangent to a circle. These three geometric constraints can be written in an algebraic form, which serve as constraints for the three dyads. For RP dyad, there could be an infinity of lines (tangent to concentric circles of different radii) in the moving frame attached to the end effector. We can even have a line passing through the center of those circles, which gives commonly known swinging block configuration of the dyad. Normally, we would choose the effector so as to minimize the length of first link, i.e., radius of the circle.

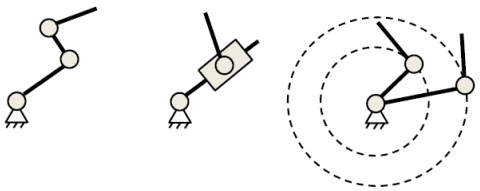

Figure 2.2: Kinematic diagram of RRR-, RPR-triads and their comparison with RR dyad

Geometric constraint of RRR, RPR, PRR, PPR, RRP and RPP triads Fig. 2.2 shows the kinematic diagram for RRR- and RPR-triads. It

is clear that a point on the end-effector of RRR and RPR stays in the region of a circular ring, just like the length of first link of RR dyad can change in an interval. Therefore, the two triads give us the same kind of geometric constraint and can be treated as same type of open chain geometrically.

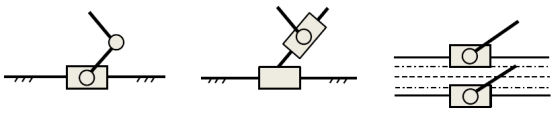

Figure 2.3: Kinematic diagram of PRR-, PPR-triads and their comparison with PR dyad

Fig. 2.3 shows the kinematic diagram for PRR- and PPR-triads. A point on the end-effector of PRR and PPR stays in the striped region delimited by a pair of parallel lines, which could be regarded as the offset of the prismatic joint of PR dyad, along the direction perpendicular to itself.

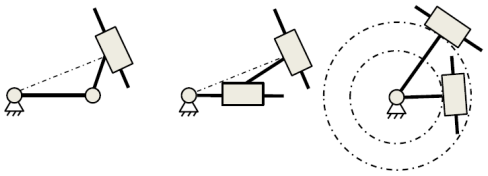

Figure 2.4: Kinematic diagram of RRP, RPP-triads and their comparisons with RP dyad

Fig. 2.4 shows the kinematic diagram for triads of RRP and RPP. It is clear that a line on the end-effector of RRP and RPP is tangent to a circle of varying radius, just like the length of first link of RP dyad can change in an interval. As mentioned earlier, we can pick numerous line-circle pairs to de-

scribe the tangency relationship in the moving frame. We call this as coupling effect between circle in fixed frame and line on the end-effector. Therefore, another way to understand this constraint is that when we move the RRP or RPP triads, there are different lines on the end-effector that will be tangent to a circle in fixed frame with fixed radius. For the sake of clarity, we prefer using the former interpretation. The coupling effect will also show up in mathematical derivations of the constraints later on.

2.2 Constraint Manifolds of Dyads and Triads

In this section, we review the concept of planar quaternions as far as necessary for the development of this chapter and discuss constraint manifolds of dyads and triads.

Planar Quaternion Consider a planar displacement in X-Y plane shown in Fig. 2.5. 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 orientation of $\mathbf{M}$ relative to $\mathbf{F}$. Then, a planar displacement can be represented by a planar quaternion, $\mathbf{Z} = (Z_1, Z_2, Z_3, Z_4)$ (see McCarthy [20]) via following kinematic mapping:

$$\begin{aligned} Z_1 &= (d_1/2)\cos(\alpha/2) - (d_2/2)\sin(\alpha/2) \\ Z_2 &= (d_1/2)\sin(\alpha/2) + (d_2/2)\cos(\alpha/2) \\ Z_3 &= \sin(\alpha/2) \\ Z_4 &= \cos(\alpha/2) \end{aligned} \tag{2.1}$$

Planar quaternion coordinates can be considered as a set of homogeneous coordinates that define the image space of planar displacements (Ravani and Roth [19]). The point **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 displacement and is denoted as Σ. In view of Eq. (2.1), the coordinates of image point must satisfy the following equation, which actually represents a rigid body constraint:

$$Z_3^2 + Z_4^2 = 1 \tag{2.2}$$

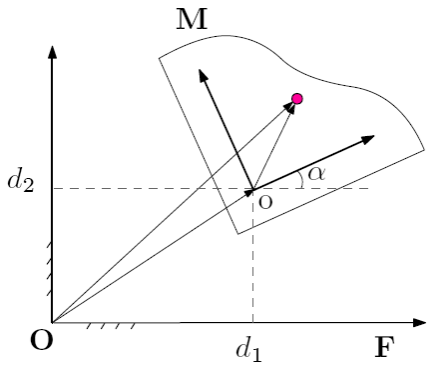

Figure 2.5: A planar displacement.

Displacement of the end-effector of dyads and triads have additional workspace related constraints that limits their position and orientation in the Cartesian space. By using quaternion representation, these constraints can be represented as quadratic surfaces (quadrics) in the image space, called constraint manifolds. For visualization, we project these four-dimensional quadrics onto a 3-D subspace (we choose $Z_4 = 1$). Now, we present a qualitative discussion on the constraint manifolds of aforementioned dyads and triads before we

present a mathematical derivation of their algebraic forms in the next section.

Constraint Manifold of RRR-, RPR-triads and RR-dyads Fig. 2.6 shows the constraint manifolds of triads RRR and RPR. The constraint manifold is a pair of hyperboloids of one sheet and its geometric parameters, such as location, orientation, and size are a function of triad geometry. Displacements that can be realized by RRR- and RPR-triads must have their image points in between the two hyperboloids shown in the figure. The constraint manifold of RR-dyads can be viewed as a single hyperboloid in between these two boundary hyperboloids. Alternatively, we can treat the single hyperboloid of RR-dyad as having thickness to become a pair of hyperboloids and space in between, just like the circle constraint of a RR-dyad becomes circular ring constraint of an RRR.

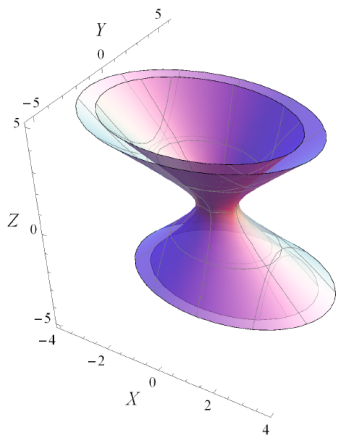

Figure 2.6: A pair of hyperboloids of one sheet associated with RRR- or RPR-triads

Constraint Manifold of PRR-, PPR-triads and PR-dyads Fig. 2.7 shows the constraint manifolds of PRR- and PPR-triads. The constraint manifold is a pair of hyperbolic paraboloids opening up along negative X-axis. Image points of displacements that can be realized by PRR- and PPR-triads must be between two surfaces. The constraint manifold of PR-dyads can be viewed as a single hyperbolic paraboloid in between these two boundary surfaces. Alternatively, we can treat the single hyperbolic paraboloid of PR-dyads as having thickness to become a pair of hyperbolic paraboloids and space in between, just as the straight line constraint of PR-dyad becomes a striped region of line constrain

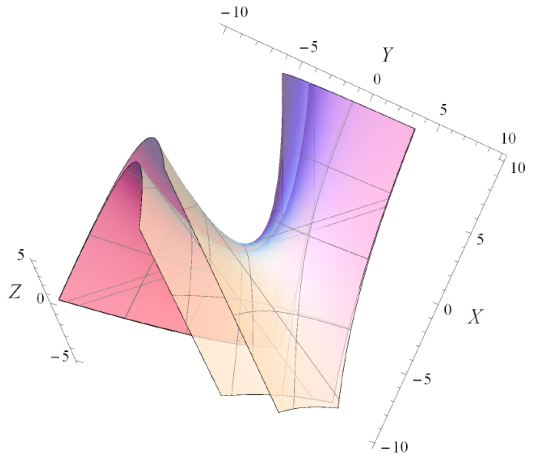

Figure 2.7: A pair of hyperbolic paraboloids opening up along negative X-axis associated with PRR- or PPR-triads

Constraint Manifold of RRP-, RPP-triads and RP-dyads Fig. 2.8 shows the constraint manifolds of RRP- and RPP-triads. Their constraint manifold is a pair of hyperbolic paraboloid opening up along positive X-axis and the space between them, while that of an RP-dyad is a single hyperbolic

parabolic. Once again, as this hyperbolic paraboloid grows in thickness, we obtain the constraint manifold of the corresponding triads. Note that both PR- and RP-dyads have hyperbolic paraboloid (HP) as their constraint manifold. In section 4.4.2, we can differentiate these two quadrics by values of coefficients of their algebraic equations. For clarity hereafter, HP of PR-dyads are being called as hyperbolic paraboloid I and that of RP-dyads as hyperbolic paraboloid II.

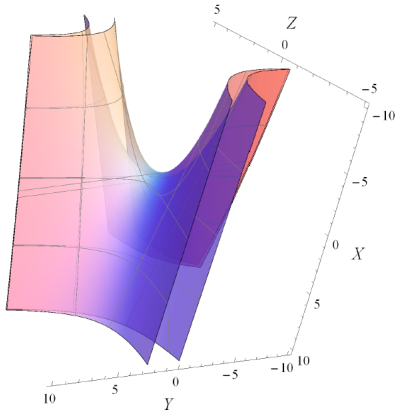

Figure 2.8: A pair of hyperbolic paraboloids opening up along positive X-axis associated with RRP- and RPP-triads

Discussion: Purwar and Gupta [64] have shown that it is intuitive to increase the gap between the pair of constraint surfaces so as to contain the image points visually. However, they do not attempt to optimize the size of the constraint surfaces – increasing the gap arbitrarily to contain the image points may lead to large manipulators. Our goal is to minimize the gap between these pair of surfaces while containing the image points.

2.3 A Unifying representation for Geometric Constraints of RR-, PR- and RP- dyads

In this section, we will first express the geometric constraints of the dyads by algebraic equations, then a unifying quadratic equation representing constraints of RR, PR and RP would be given.

2.3.1 Representation of Geometric Constraints

Let $\mathbf{X} = (X_1, X_2, X_3)$ (where $X_3 \neq 0$) denote the homogeneous coordinates of a point in the fixed frame F and $\mathbf{L} = (L_1, L_2, L_3)$ be the homogenous coordinates of a line in F, where $L_1^2 + L_2^2 = 1$ and the absolute value of L_3 is the perpendicular distance to the line from the origin of F.

Circle Constraint If a point travels around a circle in the fixed frame, its homogeneous coordinates should satisfy the following equation:

$$a_0 X_1^2 + a_0 X_2^2 - 2a_1 X_1 X_3 - 2a_2 X_2 X_3 - a_3 X_3^2 = 0 \tag{2.3}$$

Rearranging Eq. (2.3), we get:

$$2a_1 X_1 + 2a_2 X_2 + a_3 X_3 = a_0 (\frac{X_1^2 + X_2^2}{X_3}), \tag{2.4}$$

where the center of the circle is given by homogeneous coordinates $\mathbf{a} = (a_1, a_2, a_0)$ and radius r satisfies relation

$$a_0^2 r^2 - a_0 a_3 = a_1^2 + a_2^2. \tag{2.5}$$

Straight-line Constraint If a point moves along a straight line in the fixed frame, its homogeneous coordinates should satisfy the following equation

$$2a_1X_1 + 2a_2X_2 + a_3X_3 = 0, \tag{2.6}$$

where $\mathbf{a} = (a_1, a_2, a_3)$ represents the homogeneous coordinates of the straight line. Equation (2.6) can be seen as a special case of Eq. (2.4), when the center point of the circle moves to infinity ($a_0 = 0$).

Line-tangent-to-circle Constraint If a line remains tangent to a circle in the fixed frame, the constraint equation is obtained by computing the distance from the center of the circle to the line, which is equal to the radius r of the circle. If (a_1, a_2, a_0) are homogeneous coordinates of center of the circle while (L_1, L_2, L_3) are those of the moving line, then from the projective geometry of conics (Sommerville [65]), we obtain following constrain equation:

$$a_1L_1 + a_2L_2 + a_0L_3 = \pm a_0 r. \tag{2.7}$$

Similarity between Eqns. (2.4), (2.6), and (2.7) lays the foundation for unifying the constraint manifold equations of RR-, PR- and RP- dyads.

2.3.2 A Unifying Representation for Constraint Manifolds

Let $\mathbf{x} = (x_1, x_2, x_3)$ and $\mathbf{X} = (X_1, X_2, X_3)$ denote the homogeneous coordinates of a point on end-effector in the moving frame M and fixed frame F, respectively. Let $\mathbf{l} = (l_1, l_2, l_3)$ and $\mathbf{L} = (L_1, L_2, L_3)$ be the homogeneous coordinates of a line of end-effector in M and F, respectively. The relationship

between $\mathbf{x}$ and $\mathbf{X}$ for a planar displacement parameterized by (d_1, d_2, α) is given as follows:

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & d_1 \\ \sin\alpha & \cos\alpha & d_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{2.8}$$

We denote the transformation matrix in above as $[H]$ and use planar quaternion coordinates in Eq. (2.1) to write $[H]$ as

$$[H] = \begin{bmatrix} Z_4^2 - Z_3^2 & -2Z_3Z_4 & 2(Z_1Z_3 + Z_2Z_4) \\ 2Z_3Z_4 & Z_4^2 - Z_3^2 & 2(Z_2Z_3 - Z_1Z_4) \\ 0 & 0 & Z_3^2 + Z_4^2 \end{bmatrix} \tag{2.9}$$

Using the duality principle between lines and points in a projective plane, we know that the line coordinates transform as $\mathbf{L} = [\overline{H}]\mathbf{l}$, where matrix $[\overline{H}]$ is given by the transpose of the inverse of $[H]$, i.e., $[\overline{H}] = ([H]^{-1})^T$ and is given by

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

Constraint Manifold of RR Dyad We substitute fixed frame coordinates from Eq. (2.8) into Eq. (2.4), and collect like terms to get

$$\begin{aligned} &-2a_0x_3(Z_1^2 + Z_2^2) + 2a_0x_1(Z_1Z_3 - Z_2Z_4) + 2a_0x_2(Z_2Z_3 + Z_1Z_4) + \\ &2a_1x_3(Z_1Z_3 + Z_2Z_4) + 2a_2x_3(Z_2Z_3 - Z_1Z_4) + 2(a_2x_1 - a_1x_2)Z_3Z_4 - \\ &(a_1x_1 + a_2x_2)(Z_3^2 - Z_4^2) + \frac{1}{2x_3}(a_3x_3^2 - a_0x_1^2 - a_0x_2^2)(Z_3^2 + Z_4^2) = 0. \end{aligned} \tag{2.11}$$

We can rewrite Eq. (2.11) in terms of the following coefficients q_i:

$$\begin{aligned} q_1 = -2a_0x_3, \quad q_2 = 2a_0x_1, \quad q_3 = 2a_0x_2, \\ q_4 = 2a_1x_3, \quad q_5 = 2a_2x_3, \quad q_6 = 2(a_2x_1 - a_1x_2), \\ q_7 = -(a_1x_1 + a_2x_2), \\ q_8 = (a_3x_3^2 - a_0x_1^2 - a_0x_2^2)/(2x_3). \end{aligned} \tag{2.12}$$

The inverse relation between q_i and mechanism parameters (a_i, x_i) is given as follows:

$$\begin{aligned} a_0 : a_1 : a_2 : a_3 &= -q_1 : q_4 : q_5 : (4q_8 - \frac{q_1(q_6^2 + 4q_7^2)}{q_4^2 + q_5^2}) \\ x_1 : x_2 : x_3 &= (q_6q_5 - 2q_7q_4) : -(q_6q_4 + 2q_7q_5) : (q_4^2 + q_5^2) \end{aligned} \tag{2.13}$$

Besides, the coefficient q_i must satisfy the following two relations:

$$\begin{aligned} q_1q_6 + q_2q_5 - q_3q_4 &= 0 \\ 2q_1q_7 - q_2q_4 - q_3q_5 &= 0. \end{aligned} \tag{2.14}$$

Equation (2.11) represents a hyperboloid of one sheet in the image space.

Constraint Manifold of PR-Dyad The formulation for PR-dyad is much the same as that of RR except that we have $a_0 = 0$. Moreover, q_1, q_2 and q_3 are zeros because $a_0 = 0$, which means that the constraints in (2.14) are automatically satisfied. In this case, Eq. (2.11) represents a hyperbolic paraboloid oriented along negative X-axis in 3-D space parameterized by $(Z_1/Z_4, Z_2/Z_4, Z_3/Z_4)$. We refer to this quadric as a hyperbolic paraboloid I.

Constraint Manifold of RP-Dyad To obtain the algebraic form of the constraint manifold of an RP-dyad, we substitute fixed frame line coordinates from $\mathbf{L} = [\overline{H}]\mathbf{l}$ into Eq. (2.7) and obtain an equation of the same form as Eq. (2.11). The difference lies in the values of q_i and it turns the constraint manifold of hyperboloid of one sheet into a hyperbolic paraboloid oriented along positive X-axis in 3-D space, which we refer to as a hyperbolic paraboloid II. The coefficients q_i are given by

$$\begin{aligned} q_1 = 0, \quad q_2 = 2a_0 l_1, \quad q_3 = 2a_0 l_2, \\ q_4 = 0, \quad q_5 = 0, \quad q_6 = 2(a_2 l_1 - a_1 l_2), \\ q_7 = -(a_1 l_1 + a_2 l_2), \\ q_8 = a_0(l_3 \pm r). \end{aligned} \tag{2.15}$$

The two relations in Eq. (2.14) are still satisfied by these q_i and the inverse computation is given as follows:

$$\begin{aligned} l_1 : l_2 : (l_3 \pm r) &= q_2 : q_3 : 2q_8 \\ a_0 : a_1 : a_2 &= (q_2^2 + q_3^2) : -(q_3 q_6 + 2q_2 q_7) : 2(q_2 q_6 - 2q_3 q_7) \end{aligned} \tag{2.16}$$

The aforementioned coupling effect shows in the inverse computation for l_3 and r. We can fix value of one variable and allow other to change. Normally for RP dyad, r is set to zero, however, for RPP- and RRP-triads, we fix l_3 while allowing r to change in accordance with the first interpretation discussed in section 2.1.

Based on the above derivations, we now have a common representation for

constraint manifolds of RR-, PR- and RP- dyads:

$$\begin{aligned} & q_1(Z_1^2 + Z_2^2) + q_2(Z_1Z_3 - Z_2Z_4) + q_3(Z_2Z_3 + Z_1Z_4) \\ & \quad + q_4(Z_1Z_3 + Z_2Z_4) + q_5(Z_2Z_3 - Z_1Z_4) + q_6Z_3Z_4 \\ & + q_7(Z_3^2 - Z_4^2) + q_8(Z_3^2 + Z_4^2) = 0 \end{aligned} \tag{2.17}$$

This defines a quadric surface in the image space with eight homogeneous coefficients. Here, we call this quadric a *generalized constraint manifold*, or *G-manifold*. Constraint manifolds of RR-, PR- and RP-dyads are in the subset of this *G-manifold* space, which must satisfy Eq. (2.14).

2.4 Algebraic Fitting of G-Manifolds

Now, let us consider the problem of fitting the G-manifolds to a set of N image points representing task displacements in the Cartesian space. By substituting for the given values of image points in Eq. (2.17), we obtain an over-constrained system of linear equations given by $[A]\mathbf{q} = 0$, where $\mathbf{q}$ is the column vector of homogeneous coefficients $q_i (i = 1 \ldots 8)$. The coefficient matrix $[A]$ of size $N \times 8$ is given by:

$$[A] = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} & A_{16} & A_{17} & A_{18} \\ \vdots & & & & & & & \vdots \\ \vdots & & & \ddots & & & & \vdots \\ \vdots & & & & & & & \vdots \\ A_{N1} & A_{N2} & A_{N3} & A_{N4} & A_{N5} & A_{N6} & A_{N7} & A_{N8} \end{bmatrix} \tag{2.18}$$

where for the ith image point, we have

$$\begin{aligned} A_{i1} &= Z_{i1}^2 + Z_{i2}^2, & A_{i2} &= Z_{i1}Z_{i3} - Z_{i2}Z_{i4} \\ A_{i3} &= Z_{i2}Z_{i3} + Z_{i1}Z_{i4}, & A_{i4} &= Z_{i1}Z_{i3} + Z_{i2}Z_{i4} \\ A_{i5} &= Z_{i2}Z_{i3} - Z_{i1}Z_{i4}, & A_{i6} &= Z_{i3}Z_{i4} \\ A_{i7} &= Z_{i3}^2 - Z_{i4}^2, & A_{i8} &= Z_{i3}^2 + Z_{i4}^2 \end{aligned} \tag{2.19}$$

Singular Value Decomposition In linear algebra, the Singular Value Decomposition (SVD) of an $N \times 8$ matrix $[A]$ is a factorization of the form:

$$[A] = [U][S][V]^T \tag{2.20}$$

where $[U]$ is an $N \times N$ orthonormal matrix, whose N columns being called the left singular vectors of $[A]$ are the eigenvectors of $[A][A]^T$; $[S]$ is an $N \times 8$ rectangular diagonal matrix with 8 non-negative singular values of $[A]$ on the diagonal, which are also the square roots of the eigenvalues of $[A]^T[A]$ or $[A][A]^T$; $[V]$ is an 8×8 orthonormal matrix, whose 8 columns being called the right singular vectors, are the eigenvectors of $[A]^T[A]$ and span the solution of the linear system $[A]\mathbf{q} = 0$.

The over-constrained system of linear equation $[A]\mathbf{q} = 0$ can be solved as a total least squares minimization problem, i.e., minimizing the fitting error $\varepsilon = ([A]\mathbf{q})^T([A]\mathbf{q})$ with the constraint $\mathbf{q}^T\mathbf{q} = 1$. SVD solves this exact problem with the solution embedded in the column vectors of the matrix $[V]$ corresponding to zero singular values of the matrix $[S]$. The error ε can be

written as

$$\varepsilon = a_1^2\lambda_1 + a_2^2\lambda_2 + \cdots + a_8^2\lambda_8 = \sum_{i=1}^{8} a_i^2\lambda_i \tag{2.21}$$

where λ_i ($\lambda_1 < \lambda_2 < \cdots < \lambda_8$) are eigenvalues of $[A]^T[A]$ or $[A][A]^T$ and a_i will be explained in the following proof for Eq. (2.21).

Proof. Since $\mathbf{q}$ is a unit vector in an eight dimensional space, we can express $\mathbf{q}$ in terms of a set of orthonormal basis. Column vectors of the matrix $[V]$, being called $\mathbf{v}_i (i = 1 \ldots 8)$, are null space solutions of $[A]\mathbf{q} = 0$ and are orthonormal vectors, and therefore serve as a suitable basis for a general solution of coefficient vector $\mathbf{q}$:

$$\mathbf{q} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \cdots + a_8\mathbf{v}_8 = \sum_{i=1}^{8} a_i\mathbf{v}_i, \tag{2.22}$$

where a_i are coefficients of linear combination and $\sqrt{\sum_{i=1}^{8} a_i^2} = 1$. Then, we have

$$[A]\mathbf{q} = [U][S][V]^T[a_1\mathbf{v}_1 + \ldots + a_8\mathbf{v}_8], \tag{2.23}$$

where $[V] = [\mathbf{v}_1, \ldots, \mathbf{v}_8]$ and above equation simplifies to

$$[A]\mathbf{q} = [U][S][a_1, \ldots, a_8]^T. \tag{2.24}$$

Further simplification of the above gives

$$[A]\mathbf{q} = [U][\underbrace{\sqrt{\lambda_1}a_1, \ldots, \sqrt{\lambda_8}a_8}_{8}, \underbrace{0, \ldots, 0}_{n-8}]^T = [U]\mathbf{w} \tag{2.25}$$

For ε, we have the final expression:

$$\varepsilon = ([U]\mathbf{w})^T([U]\mathbf{w}] = \mathbf{w}^T\mathbf{w} = \sum_{i=1}^{8} a_i^2\lambda_i \tag{2.26}$$

Based on the characteristic of image curve of task motion, three scenarios would arise:

Task image points lie on the intersection of a pencil of *G-manifolds*
In this situation, $[A]\mathbf{q} = 0$ (ignoring floating-point error in real computation). In other words, $\mathbf{q}$ is in the null-space of $[A]^T[A]$. Now that we have a pencil of G-manifolds that can perfectly fit the task image points, there are at least two eigenvalues that have to be zero, namely, $\lambda_1 = \lambda_2 = 0$ while $\mathbf{v}_1$ and $\mathbf{v}_2$ span the null-space because the rank loss of $[A]^T[A]$ is at least two. For motion generated by four-bar linkage having one RR-dyad with another RR-, PR- or RP-dyad, the rank of loss is two. Hence, different linear combinations of $\mathbf{v}_1$ and $\mathbf{v}_2$ can render different G-manifolds that perfectly fit the image curve. For motion generated by two PR- or two RP-dyads, the rank loss of $[A]^T[A]$ is three (three coefficient values q_i are zero). Hence, λ_3 will be zero and $\mathbf{v}_3$ enters into the null-space basis. Likewise, various linear combinations of $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ can construct various G-manifolds that intersecting in a common image curve. In summary, given that task image points are on the intersection of a pair of *G-manifolds*, there has to be more than one zero eigenvalues.

However, only a subset of *G-manifolds* can be used to construct RR-, PR- and RP-dyads from the pencil of G-manifolds because they have to satisfy the constraint Eq. (2.14). We can compute the coefficients of linear combinations of $\mathbf{v}_1$ and $\mathbf{v}_2$, or $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ by imposing this constraint.

Task image points lie on a single *G-manifold* In this case, we still have $[A]\mathbf{q} = 0$ and $\mathbf{q}$ is in the null-space of $[A]^T[A]$. However, there is only one eigenvalue of $[A]^T[A]$ that is zero, i.e., $\lambda_1 = 0$ corresponding to $\mathbf{v}_1$. If components of $\mathbf{v}_1$ can satisfy Eq. (2.14), there would be one RR- or PR- or RP-dyad that can go through the given tasks. However, this is unlikely to happen in kinematics as the given task usually is a motion or a set of displacements from which a motion can be constructed (see Purwar and Ge [66] for motion design), which maps to a curve in the image space. A curve can be obtained as an intersection of quadrics, which means that we would find two zero-eigenvalues. On the other hand, single zero-eigenvalue solutions are common in CAD and reverse engineering, where a point cloud data needs to be fitted with a surface.

Task image points do not lie on any of *G-manifolds* For this scenario, $[A]\mathbf{q} \neq 0$ and the error ε is not zero. The algebraic fitting error (least square error) ε can be used to form thin-shell *G-manifold* (called *TG-manifold*). As there is no single *G-manifold* can perfectly fit the image curve, our goal is to find the TG-manifolds that can fit the given image points as closely as possible. Thickness of the thin shell is directly related to the error ε – larger the least square error, thicker the *TG-manifolds* will be. Moreover, the thin-shell *G-manifold* falls right in the category of constraint manifold of six triads mentioned in section 3. For using *TG-manifold* to construct those triads, Eq. (2.14) still has to be satisfied.

For finding the minimum ε, we must use all eight vectors $\mathbf{v}_i$ to form linear combinations, each combination standing for one *G-manifold*. Combination with minimum ε under certain task motion is $\mathbf{q} = \mathbf{v}_1$ and ϵ is λ_1, while the combination with maximum ε is $\mathbf{q} = \mathbf{v}_8$ and ϵ is λ_8. By imposing Eq. (2.14), we expect to get a set of solutions for a_i, and then we can search for three sets which render least errors to construct three triads for parallel manipulator.

As Eq. (2.14) are homogeneous second-order equations, we should have at least three a_i substituted in the constraint equations if more than two solutions (real and imaginary solutions both being counted) are required. If we have only two a_i, there would be no solution or two solutions. Therefore, we choose a_1, a_2 and a_3 together with vectors $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$ corresponding to the least singular values to construct $\mathbf{q}$. Therefore, we try to find the best solution in the space spanned by $\mathbf{v}_i (i = 1, 2, 3)$ (subspace of original space spanned $\mathbf{v}_i (i = 1, \ldots, 8)$. Then, the error $\varepsilon = a_1^2\lambda_1 + a_2^2\lambda_2 + a_3^2\lambda_3$.

It is possible that in the subspace spanned by $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$, we may not find real solutions that satisfy Eq. (2.14), which means that there may not be a thin-shellthin shell manifold of hyperboloid or hyperbolic paraboloid I or hyperbolic paraboloid II that can contain the given image points. In this case, we can incrementally expand our *TG-manifold* space to $\mathbf{v}_i (i = 1, \ldots, 4)$. In this case, we have one free variable a_i to be assigned value arbitrarily, say, a_4; for each value of a_4, we can compute $a_i, (i = 1, 2, 3)$ by Eq. (2.14). If this does not yield a real solution, then we enlarge the solution space by including a_5 until a_8. In brief, the needed number of basis eigenvectors increases as

complexity of task displacements goes up.

Assuming that there exist real solutions in the space spanned by $\mathbf{v}_i (i = 1, \ldots, 4)$, $a_i, (i = 1, 2, 3, 4)$ will be real numbers. Since a_4 can be chosen arbitrarily, there would be more than three real sets of $a_i, (i = 1, 2, 3, 4)$ and then the next step is to locate the three sets that will give us three minimum values of error ε. Instead of varying a_4 and imposing Eq. (2.14) to find corresponding $a_i, (i = 1, 2, 3)$, we reformulate it as a constrained optimization problem using Lagrange Multiplier approach.

Consider a vector $\mathbf{q} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + a_3\mathbf{v}_3 + a_4\mathbf{v}_4$. It has to satisfy two constraints in Eq. (2.14), denoted as $h_1(a_1, a_2, a_3, a_4) = 0$ and $h_2(a_1, a_2, a_3, a_4) = 0$, and sum of square of its coefficients a_i is required to be 1, denoted as $g(a_1, a_2, a_3, a_4) = 1$ or $h_3(a_1, a_2, a_3, a_4) = 0$. In view of $\varepsilon = a_1^2\lambda_1 + a_2^2\lambda_2 + a_3^2\lambda_3 + a_4^2\lambda_4$ in this subspace, the constrained optimization problem is formed as follows:

$$\text{MINIMIZE} \quad \varepsilon = f(a_1, a_2, a_3, a_4) \tag{2.27}$$

$$\text{SUBJECT TO} \quad h_1(a_1, a_2, a_3, a_4) = 0 \tag{2.28}$$

$$h_2(a_1, a_2, a_3, a_4) = 0 \tag{2.29}$$

$$h_3(a_1, a_2, a_3, a_4) = 0 \tag{2.30}$$

The method of Lagrange multipliers converts this problem to the following unconstrained optimization problem:

$$\text{MINIMIZE} \quad L(a_1, a_2, a_3, a_4, \gamma_1, \gamma_2, \gamma_3) = f + \gamma_1 h_1 + \gamma_2 h_2 + \gamma_3 h_3 \tag{2.31}$$

Therefore, we have to solve following seven equations and our goal is to locate

three smallest local minimums out of those solutions.:

$$\frac{\partial L}{\partial a_i} = 0 \quad \frac{\partial L}{\partial \gamma_1} = 0 \quad \frac{\partial L}{\partial \gamma_2} = 0 \quad \frac{\partial L}{\partial \gamma_3} = 0 \tag{2.32}$$

where $i = 1, 2, 3, 4$. For the case of $\mathbf{q} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + a_3\mathbf{v}_3$, the number of feasible solutions (real and complex) constrained by Eqs. (2.28), (2.29) and (2.30) is four and above optimization problem degenerates to solving these three constraint equation. This case can still be handled by the above formulation and therefore contains six equations to solve.

If the subspace spanned by $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3, \mathbf{v}_4$ still cannot yield real solutions, then we expand the subspace to successively include $\mathbf{v}_5$, ..., $\mathbf{v}_8$. Thus, the number of equations would be increased to eleven at most. For the case of the six or seven equations, it can be solved by MATHEMATICA's Nsolve function; for cases of more than seven, it is recommended to apply polynomial homotopy algorithms because of its accuracy and computational efficiency. Homotopy algorithms are globally convergent methods for finding all of the isolated solutions to systems of polynomial equations. Here, we use the homotopy package PHCpack developed by Jan Verschelde [67, 68]. In addition to PHCpack, there are other freely available homotopy packages, including POLSYSGLP [69], Bertini [70] and HOM4PS2 [71]. Bertini besides supporitng a user-defined homotopy, provides the regeneration technique for finding isolated and positive dimensional solutions and is capable of dealing with non-square systems of polynomials. HOM4PS2 is a polyhedral homotopy solver which, in general, is a better choice for solving sparse systems.

2.5 Inverse Computation for Variation of Geometric Constraints

In section 2.3.1 and 2.3.2, we have already presented the inverse computation formula for determining dyad geometry parameters from the given values of q_i. Here, we compute geometric constraint variations in order for RR to become RRR or RPR, PR to become PRR or PPR and RP to become RRP or RPP, namely, inner and outer radius for circular ring constraint, the distance between the two parallel boundary line in the line stripe constraint and inner and outer radius for line-tangent-to-circle constraint. We show that q_8 appears in the variation computation and thickness information (least squares error) is merely related to q_8, which will be justified below.

From last section, we found three constraint manifolds of $\mathbf{q}_i (i = 1, 2, 3)$ with the smallest least squares errors. We now need to find the thickness of each constraint manifold. For each $\mathbf{q}_i$, we have $[A]\mathbf{q}_i = \mathbf{\Delta}_i$. In view of Eq. (2.17) and $Z_3^2 + Z_4^2 = 1$, we have:

$$\begin{aligned} q_1(Z_1^2 + Z_2^2) + q_2(Z_1Z_3 - Z_2Z_4) + q_3(Z_2Z_3 + Z_1Z_4) \\ +q_4(Z_1Z_3 + Z_2Z_4) + q_5(Z_2Z_3 - Z_1Z_4) + q_6Z_3Z_4 \\ +q_7(Z_3^2 - Z_4^2) + q_8 = 0 \end{aligned} \tag{2.33}$$

Denoting the maximum and minimum values of $\mathbf{\Delta}$ as δ_{max} and δ_{min} respectively, and Eq. (2.33) as $f + q_8$, two boundary constraint surfaces can be written as as follows:

$$f + q_8 - \delta_{min} = 0 \quad f + q_8 - \delta_{max} = 0 \tag{2.34}$$

The thickness of constraint manifold is given as $|\delta_{max} - \delta_{min}|$. Geometric constraint variation for RR-, PR- and RP-dyads are presented below.

Variation for RR-dyad In view of Eq. (2.5), r is represented as

$$r = \sqrt{K - \frac{4q_8}{q_1}}, \tag{2.35}$$

where $K = \frac{q_4^2+q_5^2}{q_1^2} + \frac{q_6^2+4q_7^2}{q_4^2+q_5^2}$. Since r is a monotonic square-root function with respect to q_8 and the fitting residue δ shows up in q_8, we can easily determine the range of the varying parameter r:

$$\sqrt{K - \frac{4(q_8 - \delta_{min})}{q_1}} \leq r \leq \sqrt{K - \frac{4(q_8 - \delta_{max})}{q_1}} \tag{2.36}$$

Variation for PR-dyad From Eq. (2.13), $\frac{a_3}{a_1^2+a_2^2}$ represents the distance D between the fixed frame origin and line. And with $a_0 = 0(q_1 = 0)$, we can write D as:

$$D = \frac{2q_8}{\sqrt{q_4^2 + q_5^2}} \tag{2.37}$$

Therefore, it is obvious that the range of the varying distance D is:

$$\frac{2(q_8 - \delta_{max})}{\sqrt{q_4^2 + q_5^2}} \leq D \leq \frac{2(q_8 - \delta_{min})}{\sqrt{q_4^2 + q_5^2}} \tag{2.38}$$

Variation for RP-dyad From Eq. (2.16), we have $l_3 \pm r = \frac{2q_8}{\sqrt{q_2^2+q_3^2}}$. As mentioned earlier if l_3 is fixed, while r being changed, we can also write the equation as:

$$|l_3 - \frac{2q_8}{\sqrt{q_2^2 + q_3^2}}| = r \tag{2.39}$$

If we can let r to be smaller than zero, there is no need to attach the absolute value sign; given $q_8 - \delta_{max} \leq q_8 - \delta_{min}$, the range of r is:

$$l_3 - \frac{2q_8 - \delta_{min}}{\sqrt{q_2^2 + q_3^2}} \leq r \leq l_3 - \frac{2q_8 - \delta_{max}}{\sqrt{q_2^2 + q_3^2}} \tag{2.40}$$

In practice, l_3 can be assigned with an arbitrary value by the user. The lower limit for r sometimes can be negative. The physical meaning is that when r changes from upper limit to lower one, the tangent line is getting closer to center of circle as radius of the circle approaches to zero; when r is zero, the tangent line passes through the center; when r starts from zero to lower limit, the tangent line moves far away from center in the same direction as it goes closer to center.

2.6 Example and Discussions

In this section, we present an example to show the effectiveness of our theory and algorithm. The task motion is arbitrarily given as 16 positions and listed in Table 2.1.

The size for matrix $[A]$ in Eq. (2.18) now becomes 16×8. The next step is to find the three smallest singular values and their corresponding right singular vectors of $[A]$ by SVD. The singular values are listed in Table 2.2. Three smallest singular values are $\sigma_1 = 0.0463$, $\sigma_2 = 0.0987$ and $\sigma_3 = 0.3229$. Their associated right singular vectors are listed in Table 2.3. Based on $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$, we form $\mathbf{q} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + a_3\mathbf{v}_3$, substitute a_i $(i = 1, 2, 3)$ into Eq. (2.31) and solve them together with γ_i $(i = 1, 2, 3)$ using Eq. (2.32). Note that here

Translation	Rotation(rad)
(-0.2213, -0.9272)	-2.9798
(0.5012, -0.0981)	-2.6180
(1.2235, 0.6378)	-2.3562
(1.9995, 1.3301)	-2.0944
(4.2513, 2.8383)	-1.1791
(4.5981, 2.8301)	-1.0472
(4.7678, 2.6291)	-0.9154
(4.8978, 2.0705)	-0.2618
(4.2713, 0.9772)	0.4118
(3.5008, 0.0981)	0.5236
(2.6265, -0.7877)	0.6354
(2.0011, -1.3301)	1.0472
(-0.0213, -2.6083)	1.9326
(-0.5981, -2.8301)	2.0944
(-0.8978, -2.7591)	2.3562
(-0.8952, -2.0706)	2.8798

Table 2.1: Task motion defined by 16 positions

0.0463	0.0987	0.3229	2.1115	2.6228	4.2249	5.6068	17.8257

Table 2.2: The singular values of $\mathbf{A}$

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
$\mathbf{v}_1$	-0.1155	-0.0149	-0.1717	-0.1203	0.4055	0.79645	-0.1444	0.3505
$\mathbf{v}_2$	-0.1319	-0.0193	-0.1902	0.6657	-0.2392	0.4018	0.3526	-0.3999
$\mathbf{v}_3$	0.3211	0.0722	0.4799	-0.3186	0.2838	0.2053	0.5774	-0.3223

Table 2.3: The right singular vectors of $\mathbf{A}$

we only have six equations to solve instead of seven. If we cannot find real solutions for a_i, then a_4 will need to be brought in to expand the subspace.

By applying PHCpack to solve those six equations, we get eight real so-

	a_1	a_2	a_3	γ_1	γ_2	γ_3	ε
$Sol1$	-0.7088	0.7048	0.0291	0.5652	1.0294	-0.0060	6.0106×10^{-3}
$Sol2$	0.6113	0.7554	-0.2359	-0.1292	-0.1965	-0.0122	1.2167×10^{-2}
$Sol3$	0.8579	0.3006	0.4167	-0.0114	2.9800	-0.0206	2.0563×10^{-2}
$Sol4$	0.3414	0.8117	0.4738	-0.4592	-0.2759	-0.0300	3.0082×10^{-2}

Table 2.4: The four solutions of a_i and γ_i (i=1,2,3)

lutions and the running time is 384ms on a 64 bit Windows 7 machine with Intel Core i5 CPU running at 2.40 GHz with 4 GB of RAM. Four solutions are just sign-different from the other four respectively and thus, there are only four valid real solutions. These four solutions are listed in Table 2.4, where the last column ε indicates the surface fitting error. Now that we have four real solutions, the three $\mathbf{q}_{min1}$, $\mathbf{q}_{min2}$ and $\mathbf{q}_{min3}$ with smallest fitting errors would be picked to construct our parallel manipulator.

With the knowledge of a_i, the values of $\mathbf{q}_{min1}$, $\mathbf{q}_{min2}$ and $\mathbf{q}_{min3}$ are shown in Table 2.5. Next, we need to determine the type of surfaces that these vectors represent. First, it is obvious that $\mathbf{q}_{min2}$ defines a hyperboloid; second, the first three components of $\mathbf{q}_{min1}$ and $\mathbf{q}_{min3}$ are quite close to zeros compared to other four components (not considering q_8 as it does not affect the type of the surface but only the size). That means these two vector correspond to hyperbolic paraboloid I surfaces instead of hyperboloids. To verify this further, we treat these two vectors as hyperboloid and compute the variation of radius r of circle constraint. The circular ring constraint for $\mathbf{q}_{min1}$ is shown in Fig. 2.9. It is clear that the circular ring constraint is very similar to a line-pair constraint. Therefore, we can assign zeros to q_i $(i = 1, 2, 3)$ of $\mathbf{q}_{min1}$. The

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
$\mathbf{q}_{min1}$	0.0017	0.0009	-0.0016	-0.5452	0.4478	0.2754	-0.3677	0.5397
$\mathbf{q}_{min2}$	0.2460	0.0407	0.3618	-0.5045	-0.0002	-0.7420	-0.0419	0.0118
$\mathbf{q}_{min3}$	0.0049	-0.0114	0.0045	0.0358	-0.3942	-0.8896	-0.2227	-0.0461

Table 2.5: The three vectors with smallest surface fitting errors

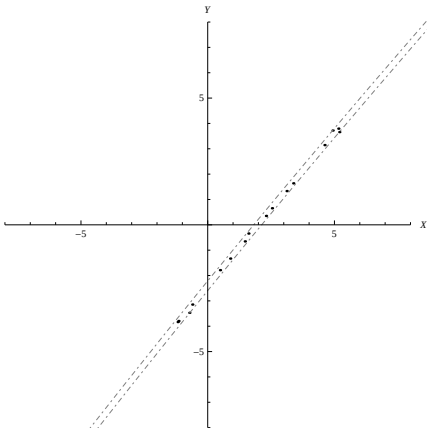

Figure 2.9: The circular ring constraint $\mathbf{q}_{min1}$ and traces of a point of end-effector in the fixed frame. What is being shown here is just a small part of the ring because of the large values of radius.

optimized constraint is plotted in Fig. 2.10. Likewise for $\mathbf{q}_{min3}$, we optimize its circular ring constraint to a line-pair constraint, shown in Fig. 2.11 and 2.12 respectively. The ring constraint of $\mathbf{q}_{min2}$ is shown in Fig. 2.13.

For $\mathbf{q}_{min1}$, its varying D is computed as follows:

$$1.44646 \leq D \leq 1.68979 \tag{2.41}$$

where both two limits have plus sign, meaning two parallel lines are located on the same side relative to origin.

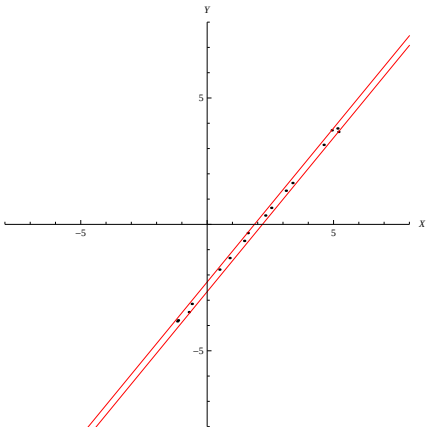

Figure 2.10: The optimized line-pair constraint $\mathbf{q}_{min1}$ and traces of a point of end-effector in the fixed frame.

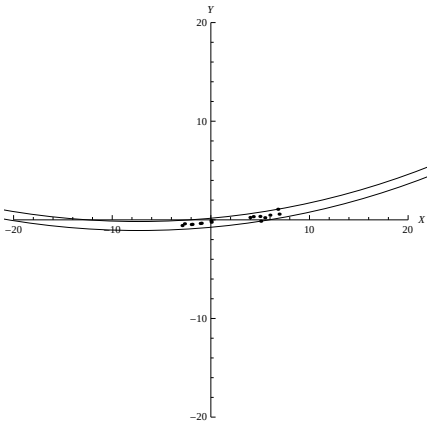

Figure 2.11: The circular ring constraint for $\mathbf{q}_{min3}$ and traces of a point of end-effector in the fixed frame. Only a small part of the ring is being shown here because of the large values of radius.

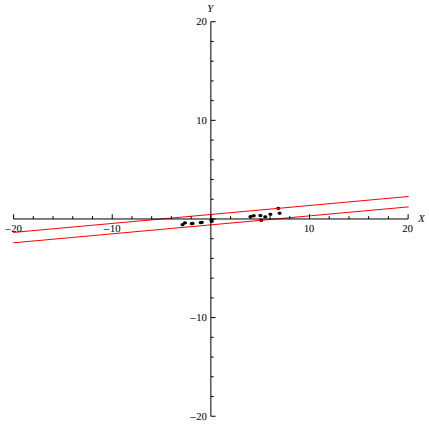

Figure 2.12: The optimized line-pair constraint for $\mathbf{q}_{min3}$ and traces of a point of end-effector in the fixed frame.

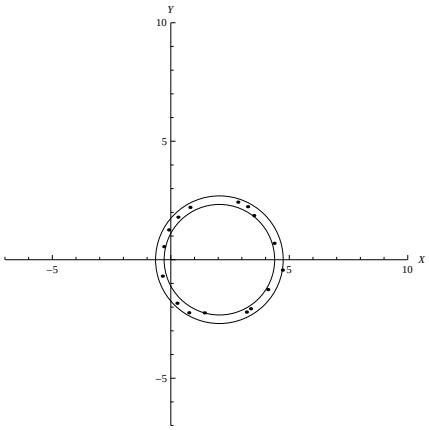

Figure 2.13: The circular ring constraint for $\mathbf{q}_{min2}$ and traces of a point of end-effector in the fixed frame.

L	Link1	x	y
y=1.2175x-2.4706	0.1217	-0.5577	0.9633

Table 2.6: Resulting parameters of PRR triad defined by $\mathbf{q}_{min1}$ and its fitting error

For $\mathbf{q}_{min2}$, its varying r is computed as follows:

$$2.33228 \leq r \leq 2.69346 \tag{2.42}$$

For $\mathbf{q}_{min3}$, its varying D is computed as follows:

$$-0.594281 \leq D \leq 0.454605 \tag{2.43}$$

where two limits have different sign, indicating origin is positioned in between two parallel lines.

Based on the above discussion, $\mathbf{q}_{min1}$ can be used to construct a PRR- or PPR-triad, $\mathbf{q}_{min2}$ to a RRR- or RPR-triad and $\mathbf{q}_{min3}$ to a PRR- and PPR-triad. Constraint manifolds for $\mathbf{q}_{min1}$, $\mathbf{q}_{min2}$ and $\mathbf{q}_{min3}$ are shown in Fig. 2.14, 2.15 and 2.16. Let us choose PRR for $\mathbf{q}_{min1}$ and $\mathbf{q}_{min3}$, denote the line equation for first joint P as L, first link as Link1 (the one directly connected to prismatic joint as shown in Fig. 2.3 and the coordinate of point on end-effector as (x, y) (in moving frame). These four parameters are listed in Tables 2.6 and 2.8. Choosing RRR for $\mathbf{q}_{min2}$, and denoting the coordinate of the fixed pivot as (X_c, Y_c), first link as Link1, second link as Link2 and the coordinate of point on end-effector as (x, y) (in moving frame), the six parameters for this triad are listed in Table 2.7.

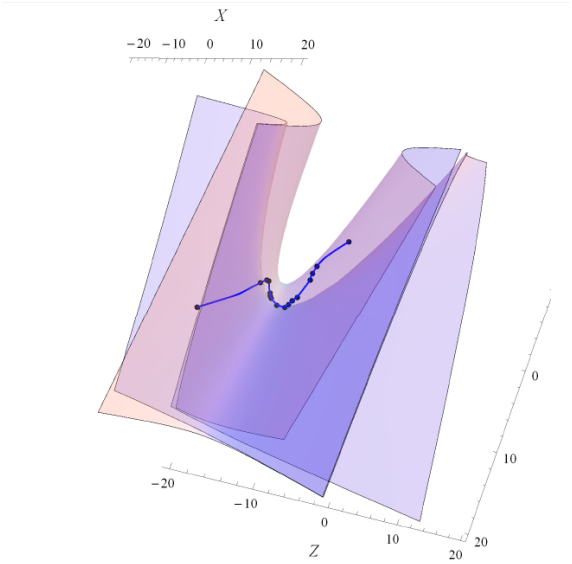

Figure 2.14: A pair of hyperbolic paraboloid I defined by $\mathbf{q}_{min1}$ and its fitting error. Image curve of task motion is in between two surfaces.

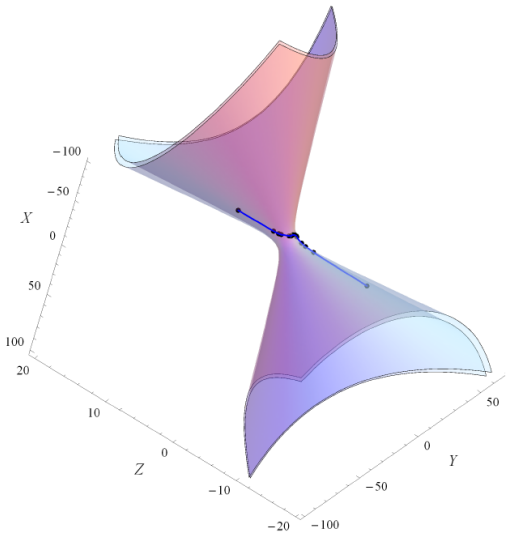

Figure 2.15: A pair of hyperboloid defined by $\mathbf{q}_{min2}$ and its fitting error. Image curve of task motion is in between two surfaces.

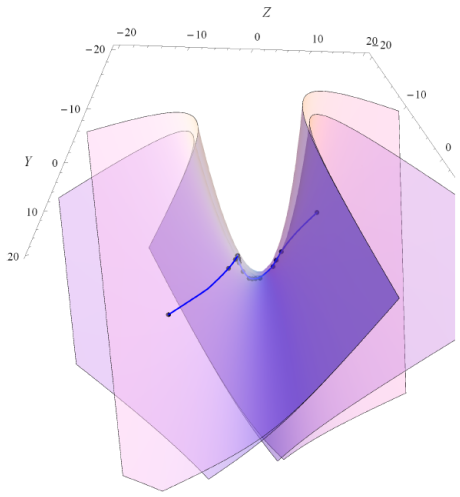

Figure 2.16: A pair of hyperbolic paraboloid I defined by $\mathbf{q}_{min3}$ and its fitting error. Image curve of task motion is in between two surfaces.

X_c	Y_c	Link1	Link2	x	y
2.0508	0.0009162	2.5129	0.1806	-0.1655	-1.4709

Table 2.7: Resulting parameters of RRR triad defined by $\mathbf{q}_{min2}$ and its fitting error

L	Link1	x	y
y=0.09096x-0.07013	0.5244	2.3402	-0.9171

Table 2.8: Resulting parameters of PRR triad defined by $\mathbf{q}_{min3}$ and its fitting error

2.7 Conclusions

In this chapter, we extended our four-bar synthesis algorithm to planar parallel manipulator synthesis by discovering similarities between dyads and triads both in geometric constraints in mechanical space and constraint manifolds in image space. An exact four-bar motion can be fitted using two or three eigenvectors of surface fitting matrix and their corresponding singular values equal to zeros, thus causing fitting error to be zero. However, with the increase of complexity of task motion, three or more eigenvectors would be used and their corresponding singular values are nonzero, therefore producing fitting error. As for parallel manipulators, we need to find three quadrics, satisfying the requirements for RR-, PR- and RP-dyad, with smallest fitting errors to generate their thin-shell version in order to contain task image points inside. The minimum-finding problem is converted to Lagrange multiplier method with equations solved by homotopy continuation approach.

Chapter 3

A Unified Algorithm for Geometric Design of Platform Linkages with Spherical and Planar Constraints

The determination of the geometric parameters of a platform robotic system so that it guides the platform through a number of specified spatial positions is known as the rigid body guidance problem. This problem has been studied extensively for planar, spherical, as well as spatial mechanisms known as the dimensional synthesis or geometric design [4, 35, 72, 73, 74, 75, 76, 77, 78, 79, 80]. The key to the problem is to formulate the design equations for a given type of mechanism and find the feasible solutions for the link parameters for a given set of goal positions.The simplest case, called the Burmester problem, is to find the dimensions of a planar 4R linkages (where R denotes a revolute joint) for a set of five positions of the coupler link. Geometrically, the coupler motion of a planar 4R linkage is constrained such that two points of the coupler lie on two separate circles. In this case, the problem of dimensional synthesis

is to determine the center and radius of each of the two circles that constrains the coupler link while it guides a planar body through five task positions. Each circle defines a planar RR dyad that constrains the coupler link.

Innocenti [35] presented a polynomial solution to the spatial of this Burmester problem, which is to find the center and radius of each of the five spheres that constrains a spatial platform while it guides a spatial body through seven task positions. Each sphere defines a SS chain, where S denotes a spherical (or ball) joint. This dissertation considers not only spatial motion for which one point lies on a sphere but also spatial motion for which one point lies on a plane. It utilizes a unified representation of sphere and a plane to develop a new method for simultaneous synthesis of dimensions as well as types of the constraints, spherical or planar, for a given set of seven positions of the platform linkage. The resulting algorithm is an extension of our recent work for simultaneous type and dimensional synthesis of planar Burmester problem [81].

The organization of the chapter is as follows. Section 3.1 reviews dual quaternion based representation of spatial displacement, which serves as the foundation of our approach. In section 3.2, six manipulator leg types and their corresponding geometric constraint are discussed. Next, we give a unified algebraic equation in section 3.3 for those manipulator legs, thus encapsulating type and geometric constraint information in a single equation. Section 3.4 presents a least square based algorithm to extract geometric parameters from arbitrarily given finite positions. Section 3.5 provides an numerical example to demonstrate our approach. Finally, conclusions and future work are given in section 3.6.

3.1 Representation of Spatial Displacement

A spatial displacement of a rigid body is commonly represented by the following transformation of a moving frame M attached to the moving body with respect to a fixed frame F attached to the fixed space:

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_x & \mathbf{r}_y & \mathbf{r}_z & \mathbf{d} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \tag{3.1}$$

where, $\mathbf{X} = (X_1, X_2, X_3, X_4)$ and $\mathbf{x} = (x_1, x_2, x_3, x_4)$ are homogeneous coordinates of a point in F and M, respectively; $\mathbf{r}_x$, $\mathbf{r}_y$, $\mathbf{r}_z$ and $\mathbf{d}$ are the axes and origin of M expressed in F respectively. The matrix $[R]$, formed by $\mathbf{r}_x$, $\mathbf{r}_y$, $\mathbf{r}_z$, is an orthogonal matrix representing a rotation while $\mathbf{d}$ represents a translation.

Alternatively, one may use a unit dual quaternion $\hat{\mathbf{Q}} = (\mathbf{Q}, \tilde{\mathbf{Q}})$, where the real part $\mathbf{Q} = (q_1, q_2, q_3, q_4)$ and the dual part $\tilde{\mathbf{Q}} = (g_1, g_2, g_3, g_4)$ to represent a spatial displacement [4]. The real part $\mathbf{Q}$ can be constructed with rotation axis $\mathbf{s} = (s_x, s_y, s_z)$ and rotation angle θ, as shown in Fig. 3.1, from the ration matrix $[R]$ using Cayley's formula [6]:

$$\mathbf{Q} = (s_x \sin\frac{\theta}{2}, s_y \sin\frac{\theta}{2}, s_z \sin\frac{\theta}{2}, \cos\frac{\theta}{2}) \tag{3.2}$$

The dual part $\tilde{\mathbf{Q}}$ is given by the formula:

$$\begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -d_3 & d_2 & d_1 \\ d_3 & 0 & -d_1 & d_2 \\ -d_2 & d_1 & 0 & d_3 \\ -d_1 & -d_2 & -d_3 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}. \tag{3.3}$$

It is easy to verify that $\mathbf{Q}$ and $\tilde{\mathbf{Q}}$ satisfy the following relations:

$$\begin{aligned} q_1^2 + q_2^2 + q_3^2 + q_4^2 &= 1, \\ q_1 g_1 + q_2 g_2 + q_3 g_3 + q_4 g_4 &= 0. \end{aligned} \tag{3.4}$$

Inversely, the rotation matrix $[R]$ can be parameterized with $\mathbf{Q} = (q_1, q_2, q_3, q_4)$ as:

$$\begin{bmatrix} q_4^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_3 q_4) & 2(q_1 q_3 + q_2 q_4) \\ 2(q_1 q_2 + q_3 q_4 & q_4^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_1 q_4) \\ 2(q_1 q_3 - q_2 q_4 & 2(q_2 q_3 + q_1 q_4) & q_4^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{3.5}$$

The translation vector $\mathbf{d} = (d_1, d_2, d_3)$ can be recovered from (4.17) in terms of $(\mathbf{Q}, \tilde{\mathbf{Q}})$ by the following:

$$\mathbf{d} = -2 \begin{bmatrix} g_4 q_1 - g_1 q_4 + g_2 q_3 - g_3 q_2 \\ g_4 q_2 - g_2 q_4 + g_3 q_1 - g_1 q_3 \\ g_4 q_3 - g_3 q_4 + g_1 q_2 - g_2 q_1 \end{bmatrix} \tag{3.6}$$

With unit dual quaternion $\hat{\mathbf{Q}}$, we recast (3.1) and obtain the following rela-

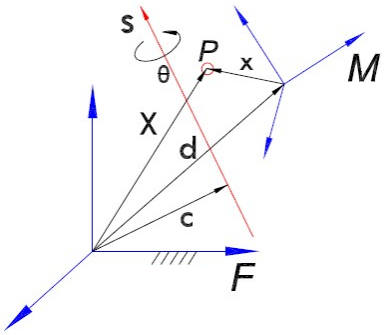

Figure 3.1: A spatial displacement.

tionship between $\mathbf{X} = (X_1, X_2, X_3, X_4)$ and $\mathbf{x} = (x_1, x_2, x_3, x_4)$:

$$\mathbf{X} = \mathbf{Q}x\mathbf{Q}^* + x_4[(\tilde{\mathbf{Q}})\mathbf{Q}^* - \mathbf{Q}(\tilde{\mathbf{Q}})^*] \tag{3.7}$$

where $\mathbf{Q}^*$ and $(\tilde{\mathbf{Q}})^*$ are conjugates of $\mathbf{Q}$ and $\tilde{\mathbf{Q}}$, respectively.

3.2 Constraining a Spatial Displacement

In this section, we study various legs of a spatial parallel manipulator that is subject to either a spherical or planar constraint.

The task-driven synthesis concerns with acquiring geometric constraint from the task motion and uses it to determine type and dimension of mechanisms simultaneously (Wu et al. [63]). When end effector travels in a spatial motion constrained by a manipulator leg, point in the end effector could be subject to certain geometric constraint in fixed frame. Here, we consider spatial motions constrained by simple geometric constraints such as spheres and

planes and their corresponding manipulator leg.

3.2.1 Spherical Constraint

Now let us consider the displacement of a rigid body (the platform) for which a point is constrained to stay on surface of sphere. This single algebraic constraint reduces the degrees of freedom (DOF) of the moving body from 6 to 5. Three types of open kinematic chains that can be used to generate such a constraint: RRS, SS and TS (R stands for revolute joint, S for spherical joint and T for universal joint), although S and T joints can be further decomposed into combinations of R joints.

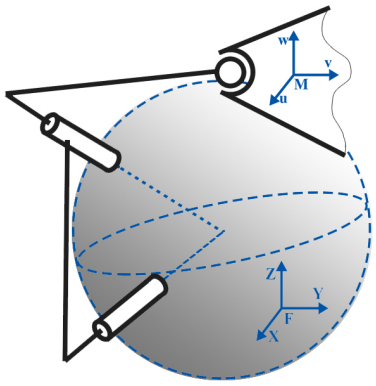

Figure 3.2: RRS Leg.

Fig. 3.2 shows an RRS open chain or a leg, where axes of two revolute joints intersect and form a spherical dyad (2R). The point on the end effector (or platform), coincident with S joint here, always stays on the surface of the sphere shown in the figure. Fig. 3.3 shows an SS leg. The point on the end effector, coincident with S joint here, again always stays on the surface of the

sphere. It is easy to see that a TS leg as shown in Fig. 3.4 also imposes a spherical constraint.

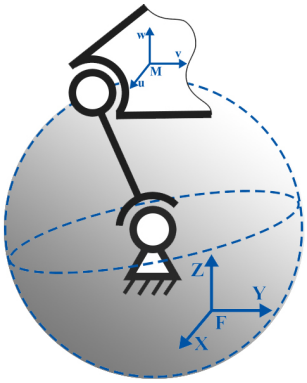

Figure 3.3: SS Leg.

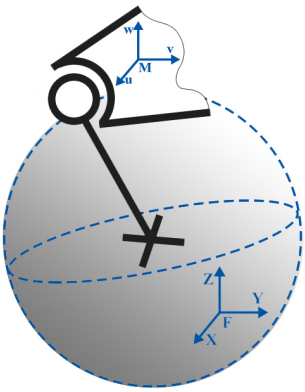

Figure 3.4: TS Leg.

Therefore, when a point on the end effector is found to be constrained on a sphere, we can apply any one of those three legs, RRS, SS and TS, to trace the task motion. For RRS leg, the design parameters can vary as long

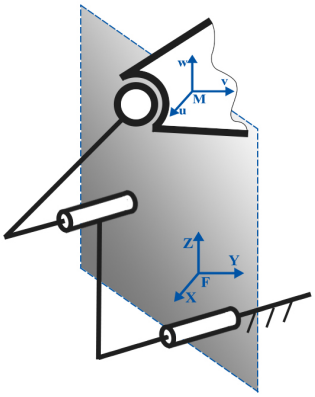

Figure 3.5: RRS Leg.

as it satisfies the same spherical constraint. For SS and TS legs, their design parameters are determined once the sphere constraints are found.

3.2.2 Planar Constraint

When the radius of a sphere is approaching infinity, the spherical constraint reduces to a planar constraint. In this case, the original RRS, SS and TS legs for spherical constraints become RRS, RPS and PRS legs for planar constraints.

For a RRS leg shown in Fig. 3.5, the axes of two revolute joints intersect at infinity, i.e., they are parallel to each other. The point on the end effector, coincident with S joint, stays on a plane in the fixed frame. For a RPS leg shown in Fig. 3.6, the axes of R and P joint are perpendicular to each other. For a PRS leg shown in Fig. 3.7, the axes of R and P joint are perpendicular to each other as well. Thus, we may select any one of them to realize the planar constraint.

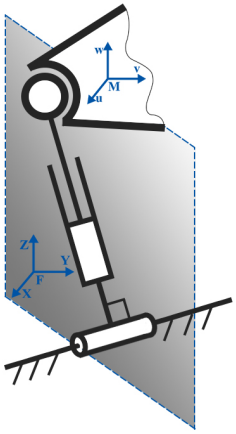

Figure 3.6: RPS Leg.

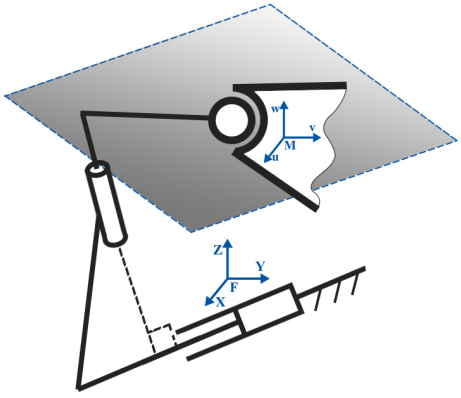

Figure 3.7: PRS Leg.

3.3 A Unified Representation of Spherical and Planar Constraints

In this section, we develop representations of spherical and planar constraints that lead to a unified representation of various open kinematic chains for constraint realization.

3.3.1 Unified Equation for Sphere and Plane

Let $\mathbf{X} = (X_1, X_2, X_3, X_4)$ (where $X_4 \neq 0$) denote the homogeneous coordinates of a point in the fixed frame. Then the homogeneous sphere equation can be written as:

$$2a_1X_1 + 2a_2X_2 + 2a_3X_3 + a_4X_4 = a_0(\frac{X_1^2 + X_2^2 + X_3^2}{X_4}) \tag{3.8}$$

where $a_0 \neq 0$. The center of the sphere is given by the homogenous coordinates:

$$\mathbf{a} = (a_1, a_2, a_3, a_0) \tag{3.9}$$

The radius r of the sphere satisfies:

$$a_0^2r^2 - a_0a_4 = a_1^2 + a_2^2 + a_3^2 \tag{3.10}$$

When $a_0 = 0$, (3.8) becomes the equation of a plane:

$$2a_1X_1 + 2a_2X_2 + 2a_3X_3 + a_4X_4 = 0 \tag{3.11}$$

Hence, Eq. (3.8) is a unified representation for both a sphere and a plane in homogenous form.

3.3.2 Unified Algebraic Equation for Manipulator Chains

With the unified representation for a sphere and a plane as given above, we now derive a unified representation for all six kinematic chains or legs associated

with the spherical and planar constraints.

We substitute fixed frame coordinates $\mathbf{X} = (x_1, x_2, x_3, x_4)$ expressed by (3.7) into (3.8), collect all of these independent terms and obtain the following equation:

$$\begin{aligned}
&p_1(-4g_4q_1 - 4g_3q_2 + 4g_2q_3 + 4g_1q_4) + p_2(-2q_1^2 + 2q_2^2 + 2q_3^2 - 2q_4^2) \\
&+p_3(-4q_1q_2 - 4q_3q_4) + p_4(-4q_1q_3 + 4q_2q_4) \\
&+p_5(4g_3q_1 - 4g_4q_2 - 4g_1q_3 + 4g_2q_4) + p_6(-4q_1q_2 + 4q_3q_4) \\
&+p_7(2q_1^2 - 2q_2^2 + 2q_3^2 - 2q_4^2) + p_8(-4q_2q_3 - 4q_1q_4) \\
&+p_9(-4g_2q_1 + 4g_1q_2 - 4g_4q_3 + 4g_3q_4) + p_{10}(-4q_1q_3 - 4q_2q_4) \\
&+p_{11}(-4q_2q_3 + 4q_1q_4) + p_{12}(2q_1^2 + 2q_2^2 - 2q_3^2 - 2q_4^2) \\
&+p_{13}(4g_1^2 + 4g_2^2 + 4g_3^2 + 4g_4^2) + p_{14}(4g_4q_1 - 4g_3q_2 + 4g_2q_3 \\
&-4g_1q_4) + p_{15}(4g_3q_1 + 4g_4q_2 - 4g_1q_3 - 4g_2q_4) \\
&+p_{16}(-4g_2q_1 + 4g_1q_2 + 4g_4q_3 - 4g_3q_4) + p_{17} = 0.
\end{aligned} \tag{3.12}$$

where p_i $(i = 1, 2, \ldots, 17)$ are homogeneous coefficients and given as follows:

$$\begin{aligned}
&p_1 = a_0x_1x_4, \quad p_2 = a_1x_1x_4, \quad p_3 = a_2x_1x_4, \quad p_4 = a_3x_1x_4 \\
&p_5 = a_0x_2x_4, \quad p_6 = a_1x_2x_4, \quad p_7 = a_2x_2x_4, \quad p_8 = a_3x_2x_4 \\
&p_9 = a_0x_3x_4, \; p_{10} = a_1x_3x_4, \; p_{11} = a_2x_3x_4, \; p_{12} = a_3x_3x_4 \\
&p_{13} = a_0x_4^2, \quad p_{14} = a_1x_4^2, \quad p_{15} = a_2x_4^2, \quad p_{16} = a_3x_4^2 \\
&p_{17} = a_0x_1^2 + a_0x_2^2 + a_0x_3^2 - a_4x_4^2
\end{aligned} \tag{3.13}$$

Husty [82] presented the algebraic equation specifically for spherical constraint.

Here, we obtain a unified representation for all six legs under either spherical or planar constraint.

In (3.13), $(a_1, a_2, a_3, a_4, a_0)$ are homogeneous coordinates of constraint in the fixed frame while (x_1, x_2, x_3, x_4) are homogeneous coordinates of point in the moving frame. They constitute design parameters of six manipulator legs. The inverse relation between p_i and (a_i, x_i) is given as follows:

$$\begin{aligned} x_1 : x_2 : x_3 : x_4 &= p_1 : p_5 : p_9 : p_{13} \\ &= p_2 : p_6 : p_{10} : p_{14} \\ &= p_3 : p_7 : p_{11} : p_{15} \\ &= p_4 : p_8 : p_{12} : p_{16} \end{aligned} \tag{3.14}$$

and

$$\begin{aligned} & a_0 : a_1 : a_2 : a_3 : a_4 \\ &= p_{13} : p_{14} : p_{15} : p_{16} : (\frac{p_1 p_2}{p_{14}} + \frac{p_5 p_6}{p_{14}} + \frac{p_9 p_{10}}{p_{14}} - p_{17}) \\ &= p_{13} : p_{14} : p_{15} : p_{16} : (\frac{p_1 p_3}{p_{15}} + \frac{p_5 p_7}{p_{15}} + \frac{p_9 p_{11}}{p_{15}} - p_{17}) \\ &= p_{13} : p_{14} : p_{15} : p_{16} : (\frac{p_1 p_4}{p_{16}} + \frac{p_5 p_8}{p_{16}} + \frac{p_9 p_{12}}{p_{16}} - p_{17}) \end{aligned} \tag{3.15}$$

It is noted that there exist linear relationships between p_i $(i = 1, 2, \ldots, 16)$

$$\begin{aligned}
&p_1 : p_2 : p_3 : p_4 = p_5 : p_6 : p_7 : p_8 \\
&p_1 : p_2 : p_3 : p_4 = p_9 : p_{10} : p_{11} : p_{12} \\
&p_1 : p_2 : p_3 : p_4 = p_{13} : p_{14} : p_{15} : p_{16} \\
&p_5 : p_6 : p_7 : p_8 = p_9 : p_{10} : p_{11} : p_{12} \\
&p_5 : p_6 : p_7 : p_8 = p_{13} : p_{14} : p_{15} : p_{16} \\
&p_9 : p_{10} : p_{11} : p_{12} = p_{13} : p_{14} : p_{15} : p_{16}
\end{aligned} \tag{3.16}$$

It can be found that p_{17} is an independent coefficient.

In Eq. (3.16), for example, $p_1 : p_2 : p_3 : p_4 = p_5 : p_6 : p_7 : p_8$ actually means

$$\begin{aligned}
&p_1p_6 - p_2p_5 = p_1p_7 - p_3p_5 = p_1p_8 - p_4p_5 = 0 \\
&p_2p_7 - p_3p_6 = p_2p_8 - p_4p_6 = p_3p_8 - p_4p_7 = 0
\end{aligned} \tag{3.17}$$

The same interpretation applies to the other five equations in Eq. (3.16).

Now that seventeen homogeneous coefficients p_i contain seven independent design parameters a_1/a_0, a_2/a_0, a_3/a_0, a_4/a_0, x_1/x_4, x_2/x_4 and x_3/x_4, there exist nine independent constraint equations between p_i. By reducing (3.16), we have:

$$\begin{aligned}
&p_2p_5 - p_1p_6 = 0,\ p_2p_7 - p_3p_6 = 0,\ p_2p_8 - p_4p_6 = 0 \\
&p_2p_9 - p_1p_{10} = 0,\ p_2p_{11} - p_3p_{10} = 0,\ p_2p_{12} - p_4p_{10} = 0 \\
&p_2p_{13} - p_1p_{14} = 0.\ p_2p_{15} - p_3p_{14} = 0,\ p_2p_{16} - p_4p_{14} = 0
\end{aligned} \tag{3.18}$$

It is not difficult to conclude that (3.18) captures the relations in (3.16) except for the cases of $p_1 = p_2 = p_3 = p_4 = 0$ or $p_2 = p_6 = p_{10} = p_{14} = 0$. If these two special cases happen, we need to go back and check (3.16) to see whether all of its relations are satisfied.

In summary, (3.12) together with (3.18) defines a unified algebraic equation for six manipulator legs. Here, we call (3.12) a *generalized equation*, or *G-equation* in short.

3.4 Algebraic Fitting of G-Equation

This section seeks to solve the problem of determining the type and dimensions of a spatial platform mechanism such that the platform guides through a set of given spatial positions. This problem is reduced to that of fitting a family of G-equations to these positions using the geometric parameters of spherical and planar constraints as design parameters.

For simplicity, we write G-equation as $p_i z_i = 0$ $(i = 1, 2, \ldots, 17)$. Therefore, our goal is to find a set of p_i to satisfy $p_i z_{k,i} = 0$ $(k = 1, 2, \ldots, N)$ for N-position synthesis, which can be formulated as a linear problem $[Z]\mathbf{p} = 0$. The coefficient matrix $[Z]$ is given by:

$$[Z] = \begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,16} & z_{1,17} \\ \vdots & & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & & \vdots \\ z_{N,1} & z_{N,2} & \cdots & z_{N,16} & z_{1,17} \end{bmatrix} \tag{3.19}$$

Thus the problem is converted to that of finding the null space of matrix $[Z]$. To this end, one needs to determine the basis of the null space. First, $[Z]$ is factorized using Singular Value Decomposition(SVD) (see Golub [83]) to the following form:

$$[Z] = [U][\Sigma][V]^T \tag{3.20}$$

where $[U]$ is an $N \times N$ orthogonal matrix whose column vectors are the *left singular vectors* of $[Z]$; $[V]^T$ is the transpose of $[V]$, which is a 17×17 orthogonal matrix whose column vectors are the *right singular vectors* of $[Z]$; $[\Sigma]$ is an $N \times 17$ matrix with 17 nonnegative singular values of $[Z]$ on the diagonal. Hence, the basis of the null space can be constructed by the right singular vectors of $[Z]$, i.e., the column vectors of $[V]$ whose corresponding singular values are 0. Furthermore, the right singular vectors and singular values of $[Z]$ are the eigenvectors and eigenvalues of $[Z]^T[Z]$, respectively. Thus, the problem of finding null space of $[Z]$ is equivalent to that of $[Z]^T[Z]$.

Suppose the matrix $[Z]$ has m zero-eigenvalues and denote the corresponding normalized eigenvectors as $\mathbf{v}_i$ $(i = 1, 2, \ldots, m)$. Let $\mathbf{p}$ be a linear combination of $\mathbf{v}_i$, i.e.,

$$\mathbf{p} = \sum_{i=1}^{m} l_i \mathbf{v}_i. \tag{3.21}$$

In order for $\mathbf{p}$ to represent a manipulator leg, its scalar components p_i must satisfy (3.18). Since p_i are homogeneous coefficients, we can impose one more constraint equation requiring $\mathbf{p}$ to be a unit vector, i.e., $|\mathbf{p}| = 1$. In views of $\mathbf{v}_i$ being unit vectors, the constraint equation is equivalent to the following one:

$$\sum_{i=1}^{m} l_i^2 = 1 \tag{3.22}$$

We denote this constraint equation as $C_1(l_1, l_2, \ldots, l_m)$ and those constraints in (3.18) as $C_i(l_1, l_2, \ldots, l_m)$ $(i = 2, \ldots, 10)$, respectively.

In order to get finite solutions on l_i, i.e., finite solutions on $\mathbf{p}$, the number m of unknowns l_i should be equal to that of constraint equations. As there are 10 constraint equations C_i, m is assigned as 10, which means the number of zero-eigenvalues of $[Z]^T[Z]$ is 10, i.e, nullity($[Z]^T[Z]$)=10. It is known that the sum of rank and nullity of the same $N \times N$ square matrix is N. Moreover, we know rank($[Z]$)=rank($[Z]^T[Z]$) and thereby rank($[Z]$)=7. If we pick 7 arbitrary task positions to construct $[Z]$, rank of $[Z]$ would be 7, thus leading to finite solutions to l_i. If number of task positions is reduced by 1, we will have 11 zero-eigenvalues for $[Z]^T[Z]$ and l_i $(i = 1, \ldots, l_{11})$, thus giving us one free choice of l_k $(1 \leq k \leq 11)$ to form $\mathbf{p}$. In this case, the solutions of $\mathbf{p}$ would be ∞^1. In the same manner, the solution of $\mathbf{p}$ is ∞^r if r positions are reduced from 7 positions.

In solving C_i, we again employ the polynomial homotopy based algorithm, as used in chapter 4. With the same algorithm, we also obtain the algebraic fitting error through computing the 7×1 column vector, $[Z]\mathbf{p}$ whose scalar components represent fitting error for every position. By inverting the relations of (3.14) and (3.15), design parameters a_i and x_i are calculated. Due to floating point error, a_0 is nonzero even though it represents a planar constraint. Therefore, when a_0 is so small that r in (3.10) becomes extremely large, we simply assign 0 to a_0 and get a planar constraint.

3.5 Numerical Example

In this section, we present an example to demonstrate the effectiveness of the unified synthesis algorithm.

Seven task positions are given in Table 3.1, which are in the form of rotation axis vector $\mathbf{s}=(s_x, s_y, s_z)$, rotation angle θ and translation vector $\mathbf{d}=(d_1, d_2, d_3)$. After obtaining the eigenvalues and eigenvectors of $[Z]$, the constraint equations C_i $(i = 1, 2, \ldots, 10)$ are generated and fed into PHCpack to find the solutions. The running time is 65.027 sec on a 64 bit Windows 7 laptop with Intel Core i5 CPU running at 2.40 GHz with 2 GB of RAM.

	s_x	s_y	s_z	θ	d_1	d_2	d_3
1	0.0000,	0.0000,	0.0000,	0.0000	0.0000,	0.0000,	2.0000
2	0.4202,	-0.9074,	0.0000,	14.5508	-0.0207,	0.0244,	2.6214
3	-0.1432,	-0.9897,	0.0000,	16.4166	-0.0391,	-0.0116,	2.9175
4	-0.8660,	-0.5000,	0.0000,	19.3874	0.0284,	-0.0491,	2.8319
5	-0.9999,	0.0081,	0.0000,	30.2119	0.1358,	0.0022,	2.6551
6	-0.9918,	0.1275,	0.0000,	28.6399	0.1184,	0.0309,	2.6532
7	-0.9846,	0.1748,	0.0000,	16.8191	0.0402,	0.0147,	2.7875

Notes: The units of s_x, s_y, s_z, d_x, d_y, d_z are meters; the unit of θ is degree.

Table 3.1: 7 Task Positions

The resulting number of real solutions of p_i is 14 and they are separated into two groups that differ with each other only by a sign. Since p_i are homogeneous coefficients, these two groups are considered identical so there are only seven real solutions. These seven solutions are listed in Table 3.2. Through (3.14) and (3.15), a_i and x_i are computed and listed as in Table 3.3. By observing a_0 and radius of each solution, we find solution 2, 4 and 7 define planar constraints while others represent sphere constraints. Figure 3.8

through 3.14 show the constraints with seven given positions, with point on the end effector connecting to the origin of moving frame by dashed line. The x-, y-, z-axis of moving frame are colored in red, green and blue, respectively.

Thus, we have identified seven manipulator legs that are compatible with the seven given task positions. One may select any five of the seven legs to obtain a single DOF spatial linkage, or any four of the seven legs to obtain a 2 DOF spatial linkage. For constructing a 3-DOF parallel manipulator, we can choose three out of those seven solutions. In terms of each solution among three chosen solutions, there are three corresponding manipulator leg types. Therefore, the total number of 3-DOF manipulator is 945. One possible 3-DOF manipulator is shown in Fig. 3.15 in which the first planar constraint, the second planar constraint and spherical constraint determine the leg A, B and C respectively.

	p
1	(-0.0474, -0.5327, -0.2527, -0.1144, 0.0490, 0.5502, 0.2610, 0.1181 -0.0019, -0.0218, -0.0103, -0.0047, 0.0182, 0.2047, 0.0971, 0.0439, -0.4422)
2	(-0.0001, -0.3902, 0.2252, -0.0003, -0.0002, -0.6695, 0.3863, -0.0006 0.0000, -0.0004, 0.0002, 0.0000, 0.0001, 0.3868, -0.2232, 0.0003, -0.0070)
3	(0.1335, 0.3597, 0.2806, 0.3692, -0.0482, -0.1299, -0.1013, -0.1333 0.0078, 0.0211, 0.0164, 0.0216, 0.0319, 0.0859, 0.0670, 0.0881, 0.7540)
4	(-0.0013, 0.3548, -0.2058, -0.0032, -0.0025, 0.6825, -0.3959, -0.0062 0.0000, -0.0044, 0.0025, 0.0000, 0.0014, -0.3903, 0.2264, 0.0035, -0.0738)
5	(-0.0653, 0.0266, -0.0754, -0.1892, 0.1867, -0.0759, 0.2156, 0.5406 0.0096, -0.0039, 0.0111, 0.0278, 0.0327, -0.0133, 0.0377, 0.0946, 0.7494
6	(0.1095, 0.3349, -0.2227, 0.2656, 0.0016, 0.0048, -0.0032, 0.0038 -0.0037, -0.0113, 0.0075, -0.0090, 0.0341, 0.1045, -0.0695, 0.0829, 0.8553)
7	(0.0000, -0.3872,-0.2236, 0.0000, -0.0000, 0.6708, 0.3873, -0.0000 0.0000, 0.0000,0.0000,-0.0000,-0.0000, 0.3873, 0.2236, -0.0000, -0.0000)

Table 3.2: 7 Solutions of **p**

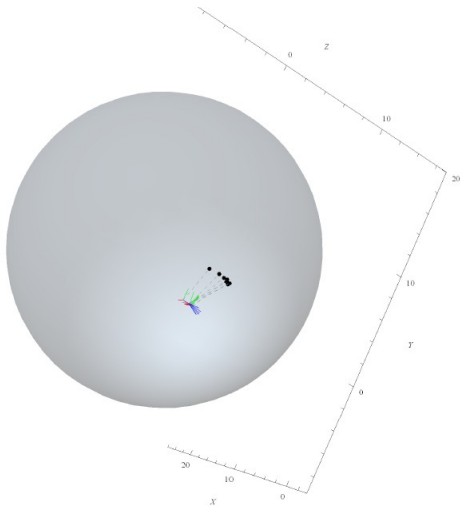

Figure 3.8: Solution 1 defines a spherical constraint

	a_0	a_1	a_2	a_3	a_4	$\frac{x_1}{x_4}$	$\frac{x_2}{x_4}$	$\frac{x_3}{x_4}$
1	0.0182,	0.2047,	0.0971,	0.0439,	0.6975	-2.6029,	2.6883,	0.1064
2	0.0001,	0.3868,	-0.2232,	0.0003,	0.0075	-1.0087,	-1.7305,	-0.0011
3	0.0319,	0.0859,	0.0670,	0.0882,	-0.1202	4.1884,	-1.5119,	0.2454
4	0.0014,	-0.3903,	0.2264,	0.0035,	0.0794	-0.9093,	-1.7489,	0.0112
5	0.0327,	-0.0133,	0.0377,	0.0946,	0.4504	-1.9985,	5.7118,	0.2936
6	0.0342,	0.1045,	-0.0695,	0.0829,	-0.5039	3.2036,	0.0462,	-0.1086
7	0.0000,	0.3873,	0.2236,	0.0000,	0.0000	-1.0507,	1.7320,	0.0000

Table 3.3: Design Parameters a_i and x_i

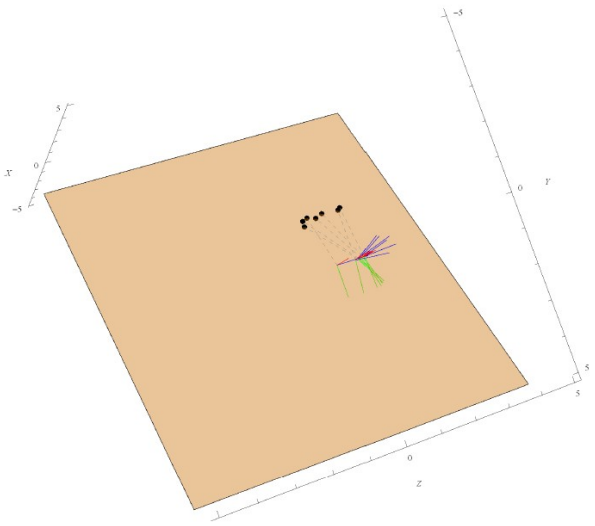

Figure 3.9: Solution 2 defines a planar constraint

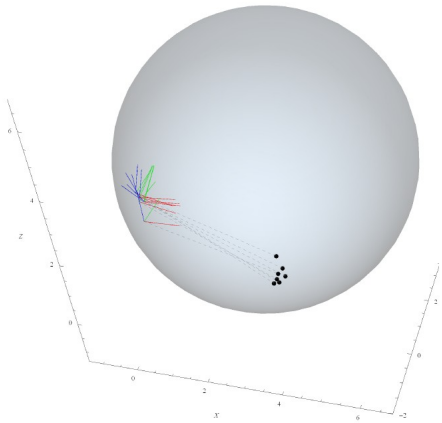

Figure 3.10: Solution 3 defines a spherical constraint

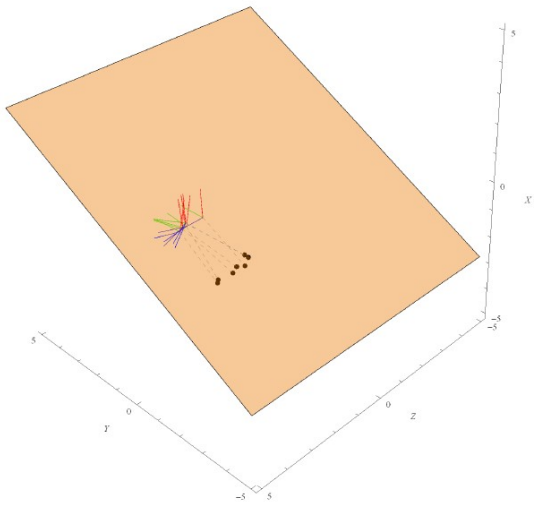

Figure 3.11: Solution 4 defines a planar constraint

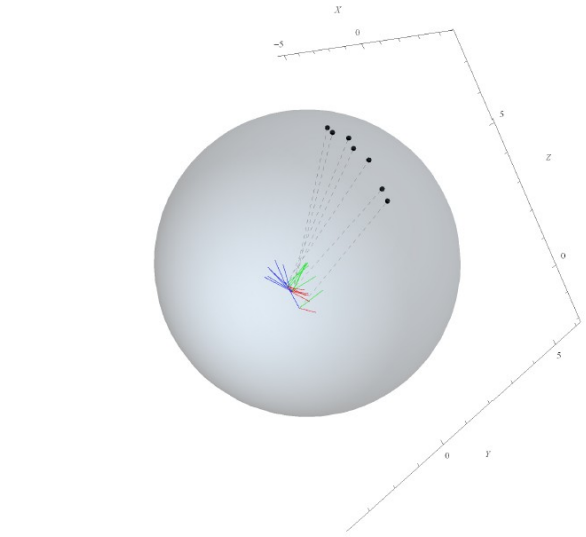

Figure 3.12: Solution 5 defines a spherical constraint

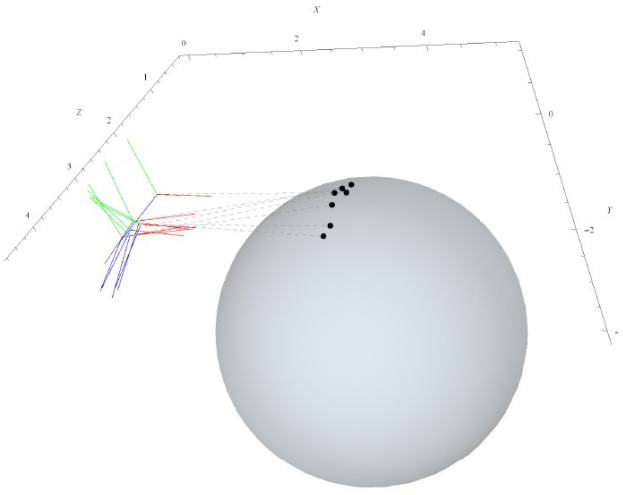

Figure 3.13: Solution 6 defines a spherical constraint

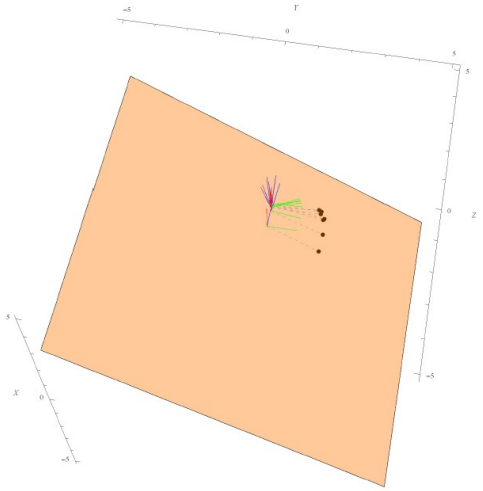

Figure 3.14: Solution 7 defines a planar constraint

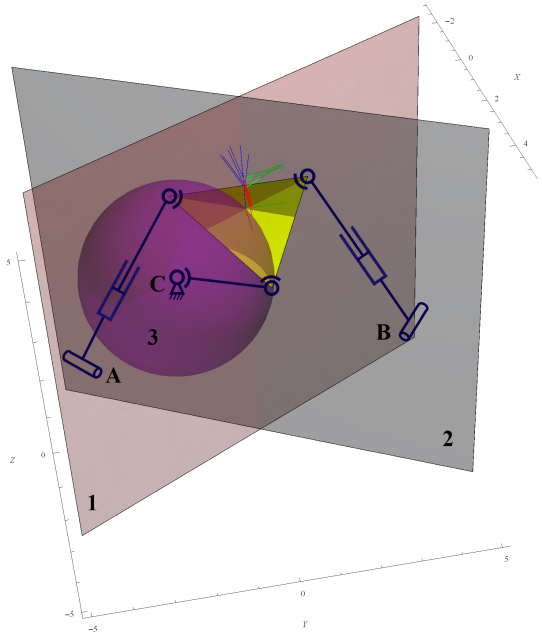

Figure 3.15: A manipulator with leg A as RPS defined by planar constraint 1, leg B as RPS defined by planar constraint 2 and leg C as SS defined by spherical constraint 3

3.6 Conclusions

In this chapter, we extend our previous work in planar and spherical mechanisms to spatial cases in the context of geometric constraints extraction from task motion. The resulting constraints can be used to determine the type and dimension of manipulator legs simultaneously. Work of this dissertation has been focused on spherical and planar constraints. Future work will follow the same approach and explore more geometric constraints related to other types of manipulator leg.

Chapter 4

A Unified Algorithm For Analysis and Simulation of Planar Four-bar Motions Defined With R- and P-Joints

Coupler motion of four-bar linkage is the motion of the moving frame which is rigidly attached to the coupler of four-bar linkage. Motion synthesis seeks to synthesize the four-bar mechanism whose coupler motion could be as close as possible to a prescribed motion or task motion, which is usually specified as either a set of finite or a continuous time-variant sequence of displacements. With the synthesized four-bar linkage, we can observe the live coupler motion through animation or simulation, visually examining the matching between coupler and task motion.

In our previous paper [62], we presented a task driven approach to simultaneous type and dimensional synthesis of planar fourbar linkage mechanism using algebraic fitting of a pencil of G-manifolds. In general, there are totally

six possible types of four-bar mechanisms that can be constructed, which are RRRR, RRRP, RRPR, PRPR, PRRP and RPPR. The advantage of our approach is that given a prescribed task motion, it can determine all the possible four-bar types together with their dimensions that are able to track the task motion. Following the theory, we are currently developing its corresponding software in iOS system. Fig. 4.1 shows the screen shot of the graphical user interface (GUI) of the design program on iPad. In terms of the animation part, we have only implemented RRRR four-bar linkage in the software which builds on loop-closure equation. We plan to include the animation codes for other four-bar types as well. However, if the method of loop-closure equation continues to be used, we have to write six different versions of animation code for each four-bar type as their loop-closure equations differ a lot, leading to code redundancy and maintenance problem. Therefore, it is imperative to have a unified algorithm towards animations of all six types of four-bar mechanisms.

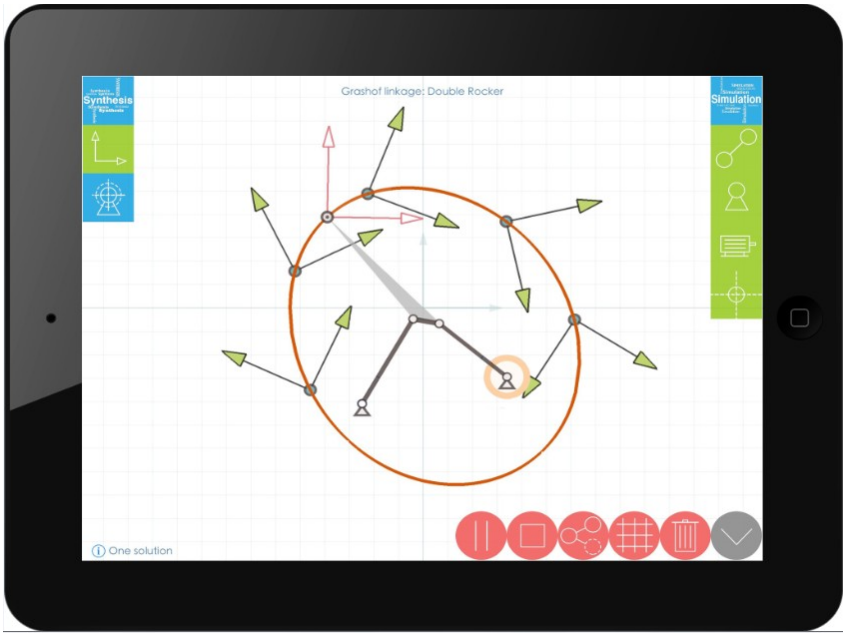

Figure 4.1: The screen shot of graphical user interface on iPad

According to our synthesis theory, the task positions are transformed into image points by planar quaternion in image space and a pencil of G-manifolds are then found to best fit those points. The dyads associated with the G-manifolds could be type of RR, PR or RP. Then two dyads are chosen to compose a four-bar linkage, with each dyad limiting the position and orientation of the coupler in its own workspace. Consequently, in cartesian space the four-bar coupler motion is the intersection of workspaces of its two constituent dyads while the coupler motion is transformed into the intersection curve of two G-manifolds in image space. Therefore, the problem of generating the coupler motion in cartesian space reduces to the problem of computing the intersection curve of two G-manifolds in image space. The image space is the

real projective three-space $\mathbb{P}^3(\mathbb{R})$ and those G-manifolds are quadrics. For RR-, PR- and RP-dyad, the quadrics are hyperboloid of one sheet, hyperbolic paraboloid I (HP I) and hyperbolic paraboloid II (HP II). HP I and II are just regular hyperbolic paraboloids only with the difference of being opening up along negative or positive X-axis.

Computing the intersection of two general quadrics is a fundamental computational geometry problem. An exact parametric representation of the intersection is often desirable. The seminal work for computing a parametric representation of the intersection between two arbitrary quadrics was due to Levin [84, 85]. It is based on an analysis of the pencil generated by the two quadrics, i.e., the set of linear combinations of the two quadrics. Building on Levin's pioneering work, Dupont et al. [86] presented the first practical and efficient algorithm for computing an exact parametric representation of the intersection of two quadric surfaces in three-dimensional real space given by implicit equations with rational coefficients. In view of the works by Levin and Dupont et al., we provide a compact and efficient algorithm to find the parametric form for the intersection of two quadrics which are limited to hyperboloid and hyperbolic paraboloid.

The organization of the chapter is as follows. Section 4.1 reviews the routine method for computing coupler motion. Section 4.2 presents our way of determining coupler motion. Finally, we present six examples in section 4.3 for each type of four-bar mechanism and their visualizations that demonstrate the method before giving concluding remarks.

4.1 Loop-closure Equation Based Coupler Motion Computation for Motion Animation

In this section, we review the common way of computing coupler motion using loop-closure equation. For simplicity, we use the RRRR-type as an example to outline the procedure.

Consider a planar 4R linkage shown in Fig. 6.2 with XOY being the fixed coordinate frame. The fixed pivot A_0 is located at point (x_0, y_0) with A_0B_0 being the ground link and A_0A the input link. Let l_i denote the length of the ith link and θ_1 the angle measured from the X axis of the fixed frame. Let ϕ, λ and ψ be the angles of link A_0A, AB, B_0B as measured from the ground link A_0B_0, respectively. A moving frame is attached to coupler link AB at P with β measured from AB to to its x-axis. Polar coordinates (r, α) represent the position of P with respect to coupler AB. All the quantities except λ and ψ are known values after synthesis. When animating the 4R linkage, we rotate the input link A_0A with ϕ being known at a given moment. In order to calculate the position and orientation of moving frame relative to XOY at each value of ϕ, the key is to find the coupler angle λ.

Using loop closure equations, it has been shown in [87] that the relationship between coupler angle λ and input link angle ϕ is given by

$$e^{i\lambda} = \frac{-B(\phi) \pm \sqrt{\Delta_1(\phi)\Delta_2(\phi)}}{2A(\phi)} \tag{4.1}$$

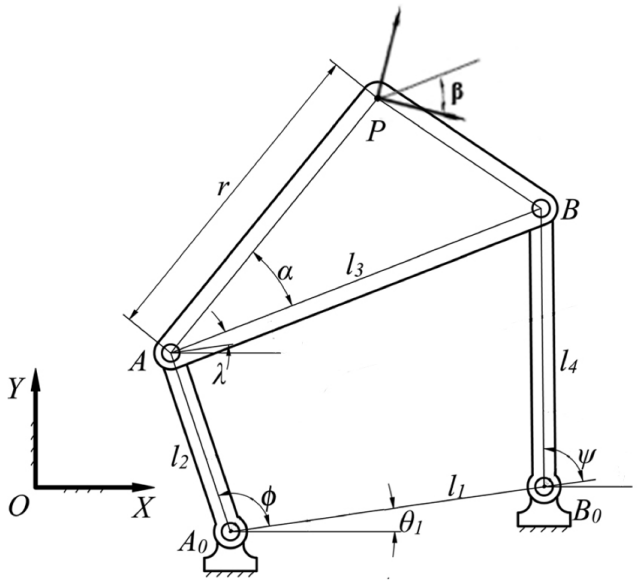

Figure 4.2: A planar 4R mechanism

where

$$A(\phi) = l_{31}(l_{21}e^{-j\phi} - 1) \tag{4.2}$$

$$B(\phi) = 1 + l_{21}^2 + l_{31}^2 - l_{41}^2 - 2l_{21}\cos\phi \tag{4.3}$$

$$\Delta_1(\phi) = 1 + l_{21}^2 - (l_{31} + l_{41})^2 - 2l_{21}\cos\phi \tag{4.4}$$

$$\Delta_2(\phi) = 1 + l_{21}^2 - (l_{31} - l_{41})^2 - 2l_{21}\cos\phi \tag{4.5}$$

$$l_{21} = l_2/l_1, \quad l_{31} = l_3/l_1, \quad l_{41} = l_4/l_1 \tag{4.6}$$

and the sign $\pm$ correspond to the two configurations of the four-bar linkage for the same input angle.

With λ being known, the position of moving frame, i.e., the position vector $\mathbf{P}$ of its origin P can be obtained by the following vector addition

$$\mathbf{OP} = \mathbf{OA}_0 + \mathbf{A}_0\mathbf{A} + \mathbf{AP} \tag{4.7}$$

where

$$\mathbf{OA}_0 = [x_0, y_0] \tag{4.8}$$

$$\mathbf{A}_0\mathbf{A} = [l_2 \cos(\phi + \theta_1), l_2 \sin(\phi + \theta_1)] \tag{4.9}$$

$$\mathbf{AP} = [r \cos(\alpha + \lambda + \theta_1), r \sin(\alpha + \lambda + \theta_1)] \tag{4.10}$$

The orientation of moving frame is calculated as

$$orientation = \theta_1 + \lambda + \beta \tag{4.11}$$

Moreover, the position of moving pivot A and B can be easily obtained as follows

$$\mathbf{OA} = \mathbf{OA}_0 + \mathbf{A}_0\mathbf{A} \tag{4.12}$$

$$\mathbf{OB} = \mathbf{OA}_0 + \mathbf{A}_0\mathbf{A} + \mathbf{AB} \tag{4.13}$$

where

$$\mathbf{AB} = [l_3 \cos(\lambda + \theta_1), l_3 \sin(\lambda + \theta_1)] \tag{4.14}$$

With the computed position and orientation of moving frame and coordinates of moving pivots , we are now able to draw the planar 4R mechanism with its coupler motion at each frame of animation. However, as mentioned earlier, this loop-closure method would be different for the other five four-bar types thus causing six different animation codes to be compiled. In next section, our approach will be presented which has the benefit of unifying the animation codes.

4.2 Image Space Based Coupler Motion Computation for Motion Animation

Our synthesis method in [81] treats the four-bar linkages as mechanisms assembled using two open chains, called dyads, connected together at the ends. Each dyad imposes kinematic constraints that limit the positions and orientations of the moving frame connected to the coupler link. By planar quaternion, kinematic constraints are transformed into geometric constraints, which are G-manifolds, in image space. Since the coupler motion is subject to both dyad constraints simultaneously, it is transformed into the image space as the intersection curve of two G-manifolds.

4.2.1 Dyads and Their G-manifolds

There are three types of dyads in use, which are RR, PR and RP. Their kinematic diagram are shown in Figs. 4.3, 4.4 and 4.5.

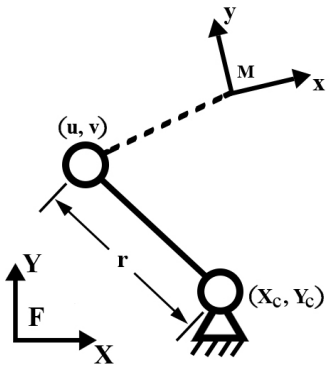

Figure 4.3: RR Dyad

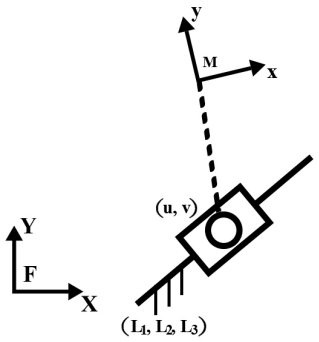

Figure 4.4: PR Dyad

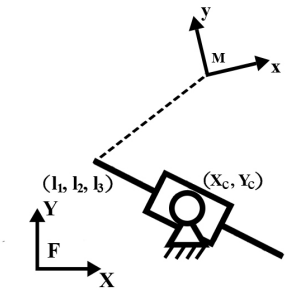

Figure 4.5: RP Dyad

The design parameters of RR dyad are fixed pivot coordinates (X_c, Y_c) relative to the fixed frame $\mathbf{F}$, moving pivot coordinates (u, v) relative to the moving frame $\mathbf{M}$ and the length r of first link. The design parameters of PR dyad are sliding axis coordinates (L_1, L_2, L_3) of P joint relative to $\mathbf{F}$ and moving pivot coordinates (u, v) relative to $\mathbf{M}$. The design parameters of RP dyad are fixed pivot coordinates (X_c, Y_c) relative to $\mathbf{F}$ and sliding axis coordinates (l_1, l_2, l_3) of P joint relative to $\mathbf{M}$.

As shown in chapter 2, The workspace constraint for RR-, PR- and RP-dyads is represented by a surface called G-manifold in image space. The homogeneous algebraic equation of G-manifold is given by

$$\begin{aligned} &q_1(Z_1^2 + Z_2^2) + q_2(Z_1Z_3 - Z_2Z_4) + q_3(Z_2Z_3 + Z_1Z_4) \\ &\quad + q_4(Z_1Z_3 + Z_2Z_4) + q_5(Z_2Z_3 - Z_1Z_4) + q_6Z_3Z_4 \\ &+ q_7(Z_3^2 - Z_4^2) + q_8(Z_3^2 + Z_4^2) = 0 \end{aligned} \tag{4.15}$$

with the coefficients q_i satisfying the following two relations:

$$\begin{aligned} q_1q_6 + q_2q_5 - q_3q_4 &= 0 \\ 2q_1q_7 - q_2q_4 - q_3q_5 &= 0. \end{aligned} \tag{4.16}$$

Upon satisfying Eq.(4.16), Eq.(4.15) represents hyperbolic paraboloid I for PR dyad when $q_1 = q_2 = q_3 = 0$ and hyperbolic paraboloid II for RP dyad when $q_1 = q_4 = q_5 = 0$. Otherwise, Eq.(4.15) represents hyperboloid for RR dyad.

With S being an arbitrary symmetric matrix of size 4, the three-dimensional quadric associated to S is then defined as the set $Q_s = \{\mathbf{Z} \in \mathbb{P}^3(\mathbb{R}) \mid \mathbf{Z}^T S\mathbf{Z} =$

0}. Here, the quadric is treated in projective three-dimensional space. G-manifold belongs to ruled quadric and its symmetric matrix G is given as

$$G = \begin{bmatrix} q_1 & 0 & \frac{q_2+q_4}{2} & \frac{q_3-q_5}{2} \\ 0 & q_1 & \frac{q_3+q_5}{2} & \frac{q_4-q_2}{2} \\ \frac{q_2+q_4}{2} & \frac{q_3+q_5}{2} & q_7+q_8 & \frac{q_6}{2} \\ \frac{q_3-q_5}{2} & \frac{q_4-q_2}{2} & \frac{q_6}{2} & q_8-q_7 \end{bmatrix} \tag{4.17}$$

Please refer to [81] to find out the relationships between design parameters of RR-, PR- and RP-dyads and q_i. Our synthesis approach would give us the values of q_i and design parameters for each G-manifold and its associated dyad. With the intersection algorithm to be presented later on, the parameterization $\mathbf{Z}(t)$ for the intersection curve can be determined, and then the position and orientation of $\mathbf{M}$ can be obtained through Eq. (4.18)

$$\begin{aligned} d_1 &= 2(Z_1Z_3 + Z_2Z_4)/(Z_3^2 + Z_4^2) \\ d_2 &= 2(Z_2Z_3 - Z_1Z_4)/(Z_3^2 + Z_4^2) \\ \cos\alpha &= (Z_4^2 - Z_3^2)/(Z_3^2 + Z_4^2) \\ \sin\alpha &= 2Z_3Z_4/(Z_3^2 + Z_4^2) \end{aligned} \tag{4.18}$$

With the $\mathbf{M}$ being calculated at each time instant, those dyad design parameters relative to $\mathbf{M}$, i.e., moving joints or moving lines, can be determined relative to $\mathbf{F}$. Upon working out all the instantaneous information about $\mathbf{M}$ and moving joint or lines, the four-bar motion can then be animated.

4.2.2 Intersection Algorithm

Given two G-manifolds Q_{G_1} and Q_{G_2}, the outline of intersection algorithm is stated as follows

1. Construct the orthonormal transformation matrix P which sends G_1 into diagonal matrix $\tilde{G}_1$ by computing the eigenvalues and the normalized eigenvectors of G_1. The same matrix P sends Q_{G_1} into canonical form $Q_{\tilde{G}_1}$. Determine the parameterization $\mathbf{Z}(u,v) = [Z_1(u,v), Z_2(u,v), Z_3(u,v), Z_4(u,v)]$ of the canonical quadric $Q_{\tilde{G}_1}$.

2. Compute the matrix $\tilde{G}_2 = P^T G_2 P$ for the quadric Q_{G_2} which transforms Q_{G_2} into $Q_{\tilde{G}_2}$. Substitute $\mathbf{Z}(u,v)$ into the algebraic equation of $Q_{\tilde{G}_2}$, i.e., $\mathbf{Z}^T \tilde{G}_2 \mathbf{Z} = 0$, to get the following equation

$$\mathbf{Z}(u,v)^T \tilde{G}_2 \mathbf{Z}(u,v) = a(v)u^2 + b(v)u + c(v) = 0 \tag{4.19}$$

   solve Eq. (4.19) for u in terms of v and use $\Delta(v) = b^2(v) - 4a(v)c(v) \geq 0$ to determine the domain of v such that real solutions exist for u and we denote the real solutions as $u(v)$. Substitute $u(v)$ into $\mathbf{Z}(u,v)$ to get the parameterization $\mathbf{Z}(v)$ for the intersection of $Q_{\tilde{G}_1}$ and $Q_{\tilde{G}_2}$.

3. Finally, $P\mathbf{Z}(v)$ is the parameterization for the intersection of Q_{G_1} and Q_{G_2}.

In step 1, whether Q_{G_1} be a hyperboloid of one sheet or hyperbolic paraboloid I or hyperbolic paraboloid II, its canonical form $Q_{\tilde{G}_1}$ stays the same as

$$aZ_1^2 + bZ_2^2 - cZ_3^2 - dZ_4^2 = 0 \tag{4.20}$$

and the parameterization $\mathbf{Z}(u, v)$ (Dupont et al. [86]) is given as

$$\mathbf{Z}(u, v) = [\frac{u + av}{a}, \frac{uv - b}{b}, \frac{u - av}{\sqrt{ac}}, \frac{uv + b}{\sqrt{bd}}] \tag{4.21}$$

In step 3, $a(v)$, $b(v)$ and $c(v)$ are polynomials of degree at most two in v due to the bilinearity of $\mathbf{Z}(u, v)$. Therefore, $\Delta(v)$ is a polynomial of degree up to four.

Over the course of the algorithm, we don't distinguish between the three types of G-manifolds, thereby unifying the coupler motion generation code.

4.3 Examples and Discussions

Now, we present six examples for the RRRR, RRRP, RRPR, PRPR, PRRP and RPPR four-bar types. For each example, the inputs are two G-manifolds corresponding to two dyads that make up a specific four-bar linkage of that type. The two G-manifolds are represented by their algebraic equations Eq. (4.15). The outputs are intersection curves of the two G-manifolds. Since homogeneous equation is used to represent G-manifold, for visualization purpose we need to project it on the hyperplane $Z_4 = 1$.

Example: RRRR Consider an example for RRRR four-bar linkage. The two algebraic equations of constraint manifolds corresponding to the two RR

dyads are

$$\begin{aligned} RR1: \quad & -2(Z_1^2+Z_2^2)+9.18(Z_1Z_3-Z_2Z_4)+2.68(Z_2Z_3+Z_1Z_4) \\ & +2.3(Z_1Z_3+Z_2Z_4)+0.76(Z_2Z_3-Z_1Z_4)+0.4064Z_3Z_4 \\ & -5.7877(Z_3^2-Z_4^2)-1.252(Z_3^2+Z_4^2)=0 \end{aligned} \tag{4.22}$$

$$\begin{aligned} RR2: \quad & -2(Z_1^2+Z_2^2)+2.48(Z_1Z_3-Z_2Z_4)+0.2(Z_2Z_3+Z_1Z_4) \\ & -4.4(Z_1Z_3+Z_2Z_4)-0.2(Z_2Z_3-Z_1Z_4)+0.192Z_3Z_4 \\ & +2.738(Z_3^2-Z_4^2)-2.43265(Z_3^2+Z_4^2)=0 \end{aligned} \tag{4.23}$$

The intersection curves are shown in Fig. 4.6. Both G-manifolds are hyperboloids of one sheet, which form two intersection curves. Each corresponds to one configuration of RRRR four-bar mechanism.

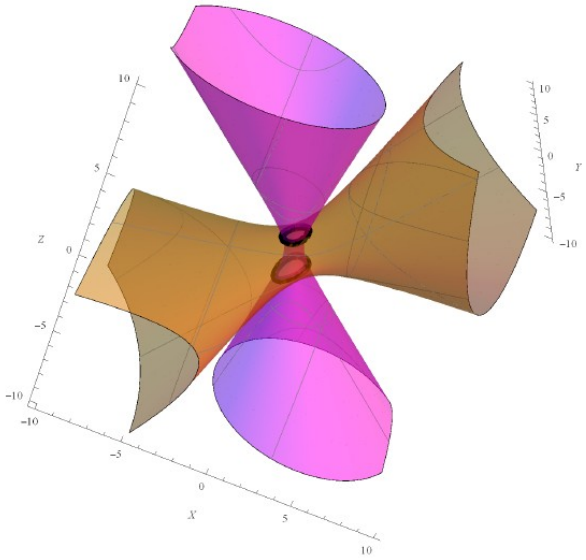

Figure 4.6: The black are intersection curves for constraint manifolds of RRRR

Example: RRRP Consider an example for RRRP (Crank-Slider) mechanism. The two algebraic equations of constraint manifolds corresponding to RR and PR are

$$\begin{aligned} RR: & -2(Z_1^2+Z_2^2)-4(Z_1Z_3-Z_2Z_4)-6(Z_2Z_3+Z_1Z_4) \\ & +0(Z_1Z_3+Z_2Z_4)+2(Z_2Z_3-Z_1Z_4)-4Z_3Z_4 \\ & +3(Z_3^2-Z_4^2)-6.5(Z_3^2+Z_4^2)=0 \end{aligned} \tag{4.24}$$

$$\begin{aligned} PR: \quad & 0(Z_1^2+Z_2^2)+0(Z_1Z_3-Z_2Z_4)+0(Z_2Z_3+Z_1Z_4) \\ & +2(Z_1Z_3+Z_2Z_4)+4(Z_2Z_3-Z_1Z_4)+10Z_3Z_4 \\ & +5(Z_3^2-Z_4^2)+1(Z_3^2+Z_4^2)=0 \end{aligned} \tag{4.25}$$

The intersection curves are shown in Fig. 4.7. One G-manifold is hyperboloid of one sheet and the other one hyperbolic paraboloid I. There are two intersection curves and each corresponds to one configuration of Crank-Slider mechanism.

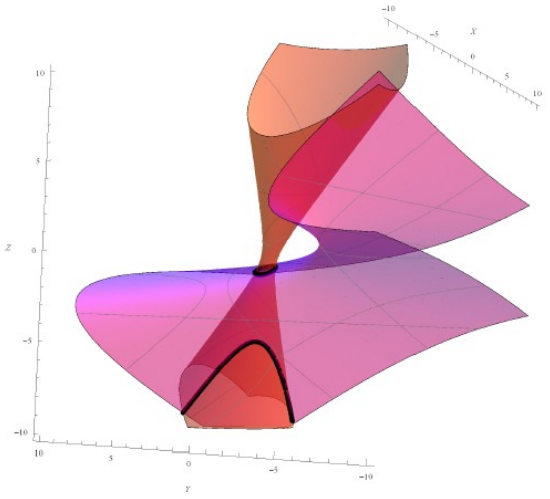

Figure 4.7: The black are intersection curves for constraint manifolds of RRRP

Example: RRPR Consider an example for RRPR (Swing-Block) mechanism. The two algebraic equations of constraint manifolds corresponding to RR and RP are

$$\begin{aligned} RR: & -2(Z_1^2+Z_2^2)-4(Z_1Z_3-Z_2Z_4)-5.98(Z_2Z_3+Z_1Z_4) \\ & +0(Z_1Z_3+Z_2Z_4)+2(Z_2Z_3-Z_1Z_4)-4Z_3Z_4 \\ & +2.99(Z_3^2-Z_4^2)-4.9711(Z_3^2+Z_4^2)=0 \end{aligned} \tag{4.26}$$

$$\begin{aligned} RP: & \quad 0(Z_1^2+Z_2^2)+0(Z_1Z_3-Z_2Z_4)+2(Z_2Z_3+Z_1Z_4) \\ & +0(Z_1Z_3+Z_2Z_4)+0(Z_2Z_3-Z_1Z_4)-4Z_3Z_4 \\ & -3(Z_3^2-Z_4^2)+3(Z_3^2+Z_4^2)=0 \end{aligned} \tag{4.27}$$

The intersection curves are shown in Fig. 4.8. One G-manifold is hyperboloid of one sheet and the other one hyperbolic paraboloid II. There are two intersection curves and each corresponds to one configuration of Swing-Block mechanism.

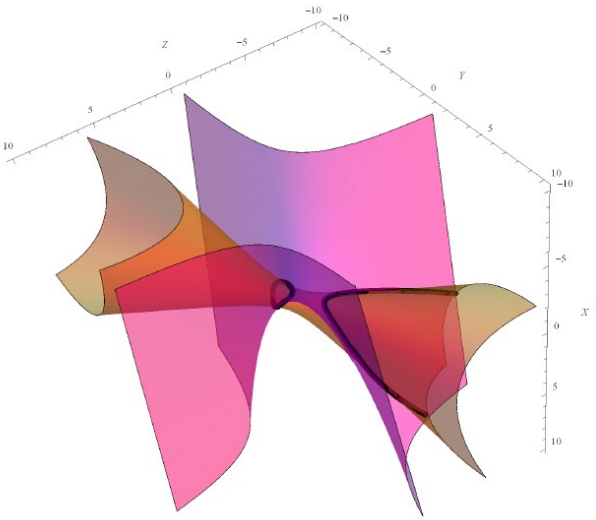

Figure 4.8: The black are intersection curves for constraint manifolds of RRPR

Example: PRPR Consider an example for PRPR (Slider-Swinging Block) mechanism. The two algebraic equations of constraint manifolds corresponding to PR and RP are

$$\begin{aligned} PR: \quad & 0(Z_1^2 + Z_2^2) + 0(Z_1Z_3 - Z_2Z_4) + 0(Z_2Z_3 + Z_1Z_4) \\ & +2(Z_1Z_3 + Z_2Z_4) + 2(Z_2Z_3 - Z_1Z_4) - 16Z_3Z_4 \\ & +4.0034(Z_3^2 - Z_4^2) + 2.0017(Z_3^2 + Z_4^2) = 0 \end{aligned} \tag{4.28}$$

$$\begin{aligned} RP: \quad & 0(Z_1^2 + Z_2^2) + 0(Z_1Z_3 - Z_2Z_4) + 2(Z_2Z_3 + Z_1Z_4) \\ & +0(Z_1Z_3 + Z_2Z_4) + 0(Z_2Z_3 - Z_1Z_4) - 5.9994Z_3Z_4 \\ & +2(Z_3^2 - Z_4^2) - 2(Z_3^2 + Z_4^2) = 0 \end{aligned} \tag{4.29}$$

The intersection curves are shown in Fig. 4.9. One G-manifold is hyperbolic paraboloid I and the other one hyperbolic paraboloid II. There are two intersection curves and each corresponds to one configuration of Slider-Swinging Block mechanism.

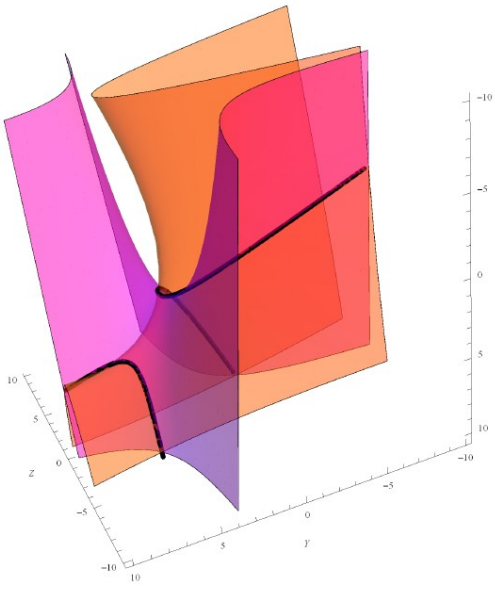

Figure 4.9: The black are intersection curves for constraint manifolds of PRPR

Example: PRRP Consider an example for PRRP (Double-Slider) mechanism. The two algebraic equations of constraint manifolds corresponding to the two PR dyads are

$$\begin{aligned} PR1: \quad & 0(Z_1^2+Z_2^2)+0(Z_1Z_3-Z_2Z_4)+0(Z_2Z_3+Z_1Z_4) \\ & +1.8632(Z_1Z_3+Z_2Z_4)-0.727(Z_2Z_3-Z_1Z_4)-5.8423Z_3Z_4 \\ & +3.0797(Z_3^2-Z_4^2)+0.5684(Z_3^2+Z_4^2)=0 \end{aligned} \tag{4.30}$$

$$\begin{aligned} PR2: \quad & 0(Z_1^2+Z_2^2)+0(Z_1Z_3-Z_2Z_4)+0(Z_2Z_3+Z_1Z_4) \\ & +1.8566(Z_1Z_3+Z_2Z_4)+0.7438(Z_2Z_3-Z_1Z_4)-10.9583Z_3Z_4 \\ & +3.4351(Z_3^2-Z_4^2)+1.2995(Z_3^2+Z_4^2)=0 \end{aligned} \tag{4.31}$$

The intersection curve is shown in Fig. 4.10. Both G-manifolds are hyperbolic paraboloid I. There is only one intersection curve corresponding to the single configuration of Double-Slider mechanism.

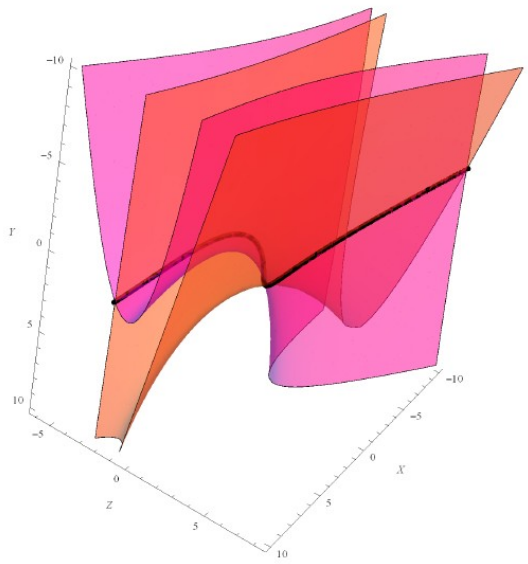

Figure 4.10: The black is intersection curve for constraint manifolds of PRRP

Example: RPPR Consider an example for RPPR (Double-Swinging Block) mechanism. The two algebraic equations of constraint manifolds corresponding to the two RP dyads are

$$\begin{aligned} RP1: \quad & 0(Z_1^2 + Z_2^2) + 0.4712(Z_1Z_3 - Z_2Z_4) + 1.9438(Z_2Z_3 + Z_1Z_4) \\ & +0(Z_1Z_3 + Z_2Z_4) + 0(Z_2Z_3 - Z_1Z_4) + 0.4973Z_3Z_4 \\ & -0.0243(Z_3^2 - Z_4^2) + 4.0818(Z_3^2 + Z_4^2) = 0 \end{aligned} \tag{4.32}$$

$$\begin{aligned} RP2: \quad & 0(Z_1^2 + Z_2^2) + 1.8574(Z_1Z_3 - Z_2Z_4) - 0.7418(Z_2Z_3 + Z_1Z_4) \\ & +06(Z_1Z_3 + Z_2Z_4) + 0(Z_2Z_3 - Z_1Z_4) + 3.4594Z_3Z_4 \\ & -3.7661(Z_3^2 - Z_4^2) + 4.2042(Z_3^2 + Z_4^2) = 0 \end{aligned} \tag{4.33}$$

The intersection curve is shown in Fig. 4.11. Both G-manifolds are hyperbolic paraboloid II. There is only one intersection curve corresponding to the single configuration of Double-Swinging Block mechanism.

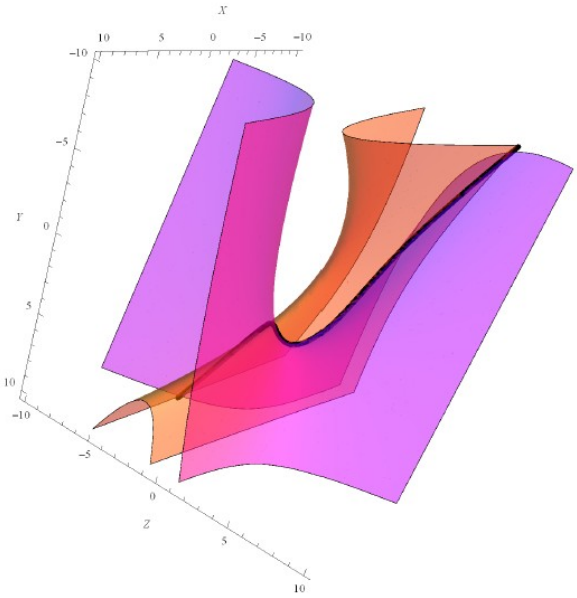

Figure 4.11: The black is intersection curve for constraint manifolds of RPPR

4.4 Conclusions

In this chapter, we presented a unified algorithm to the analysis and simulation of all types of planar four-bar motions.Instead of taking the approach of loop-closure equation towards simulations of different four-bar types, we employ planar quaternion to map a four-bar mechanism into a pair of G-manifold in image space and therefore the four-bar coupler motion ends up being the intersection curves. The animation problem is then reduced to the problem of determining the parameterization for the intersection curves. Since our G-manifolds are in the category of quadrics, and limited to hyperboloid of one sheet and hyperbolic paraboloid, there exists a simple and efficient algorithm to finding the parameterization. The six provided examples fully demonstrate the effectiveness of this algorithm.

Chapter 5

A Fourier Descriptor Based Approach to Design Space Decomposition for Planar Motion Approximation

In this chapter, we use a combination of kinematic mapping and Fourier Descriptors in our approach to motion generation that includes not only whole cycle motion but also motion over an interval. The image space of kinematic mapping is used to visualize the structural error for motion approximation. Instead of formulating the synthesis problem as a nonlinear least squares problem in the image space, we decompose the coupler motion into a point trajectory of the coupler together with a rotational motion of the coupler and use Fourier descriptors of the point trajectory and rotational motion for motion comparison and approximation. Building upon our recent work [87] that extends the Fourier Descriptor based path synthesis problem to include both whole closed and open paths, we found that the link ratios of the desired four

bar linkage can be determined directly from the rotational motion alone. This reduces the dimension of the search space of design parameters from ten to three. This decomposition of design space greatly improves the speed of the synthesis process.

The organization of this chapter is as follows. Section 5.1 reviews how Fourier descriptors may be used for representing whole cycle motion as well as motion over an interval. Section 5.2 summarizes the basics of planar kinematic mapping. Section 5.3 presents the loop closure equation of a four bar linkage in a form that is suitable for the development of this chapter. Then, we show that the high harmonic content (with order 2 or higher) of a four-bar linkage comes exclusively from the harmonic content of the rotational component of a motion. Furthermore, it has been shown that in the space defined by the Fourier descriptors, there is a linear relationship governing the rotational and translational components of a planar four-bar motion. This forms the basis for an efficient algorithm for Fourier Descriptor based method for four-bar linkage synthesis for motion generation. The details of the algorithm are presented in Section 5.4. Three examples are presented in Section 5.5 that include both whole cycle motions and motion segments.

5.1 Fourier Descriptor Based Motion Representation

A planar rigid body is shown in in Fig. 5.1. The position of the moving body relative to a fixed frame F is represented by a frame M attached to the moving

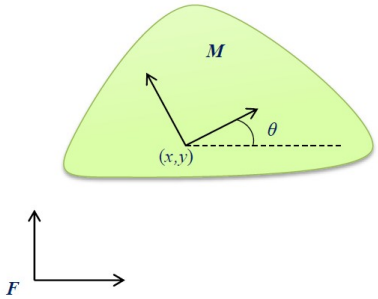

Figure 5.1: A rigid body with M as its moving Frame and F the fixed frame. Point (x, y) represents the location of the origin of M and θ the rotation of the rigid body.

body. While the origin of the moving frame M is given by (x, y), the orientation of M is defined by the angle θ. A continuous motion results when $\{x, y, \theta\}$ are defined as functions of the time parameter t, i.e., $\{x(t), y(t), \theta(t)\}$. We use complex numbers to represent both the point path (translation of motion)and the angle (orientation of motion), i.e., $z(t) = x(t) + jy(t)$ and $e^{j\theta(t)}$. When the motion is closed, both $z(t)$ and $e^{j\theta(t)}$ can be considered as periodic functions and thereby decomposed as harmonics through Fourier transform

$$z(t) = \sum_{k=-\infty}^{\infty} \alpha_k e^{jk\omega_0 t} \tag{5.1}$$

$$e^{i\theta(t)} = \sum_{k=-\infty}^{\infty} \beta_k e^{jk\omega_0 t}$$

where $\omega_0 = 2\pi/T$, T being the period of time, and the Fourier coefficients α_k and β_k are complex numbers given by

$$\alpha_k = \frac{1}{T}\int_0^T z(t)e^{-jk\omega_0 t}dt \tag{5.2}$$
$$\beta_k = \frac{1}{T}\int_0^T e^{i\theta(t)}e^{-jk\omega_0 t}dt$$

In practice, we use a finite number of harmonics to approximate $z(t) = x(t) + jy(t)$ and $e^{j\theta(t)}$ respectively

$$z(t) \approx \sum_{k=-p}^{+p} \alpha_k e^{jk\omega_0 t} \tag{5.3}$$
$$e^{i\theta(t)} \approx \sum_{k=-p}^{+p} \beta_k e^{jk\omega_0 t}$$

where p is a small positive integer that defines the order of harmonic terms used in the approximation. In this case, α_k and β_k ($k = -p, -p+1, \ldots, p-1, p$) are termed Fourier Descriptors.

Usually the task motion is given discretely in terms of an ordered sequence of displacements as well as the parameter values t_i , with $i = 1, \ldots, n$ and $0 \leq t_1 < t_2 \cdots < t_n \leq T$. Therefore, α_k and β_k can be calculated by formulating the least squares problem as follows

$$\Delta_1 = \sum_{i=1}^{n} ||z(t_i) - \sum_{k=-p}^{+p} \alpha_k e^{jk\omega_0 t_i}||^2$$
$$\Delta_2 = \sum_{i=1}^{n} ||e^{i\theta(t)} - \sum_{k=-p}^{+p} \beta_k e^{jk\omega_0 t_i}||^2 \tag{5.4}$$

where Δ_1 and Δ_2 are errors for path (translation) and angle (orientation) approximation. Eq. (5.4) can be applied to both closed and open motion.

For a closed motion, t_1 and t_n have the relationship: $t_1 = 0$ or $t_1 \approx 0$ and $t_n = T$ or $t_n \approx T$; for an open motion, t_1 and t_n have the relationship: $0 < c_1 = t_1 < t_n = c_2 < T$, with c_1 and c_2 being positive reals.

5.2 Kinematic Mapping

In previous chapter, we have already covered the concept of Kinematic Mapping. Here it is used again to visualize the synthesis result and therefore a brief review becomes necessary. Consider a planar displacement in X-Y plane shown in Fig. 5.1. Let x, y denote the coordinates of the origin of the moving frame $\mathbf{M}$ with respect to the fixed frame $\mathbf{F}$, and let θ denote the rotation angle of $\mathbf{M}$ relative to $\mathbf{F}$. Then a planar displacement can be represented by a planar quaternion, $\mathbf{Z} = (Z_1, Z_2, Z_3, Z_4)$:

$$\begin{aligned} Z_1 &= (x/2)\cos(\alpha/2) + (y/2)\sin(\theta/2) \\ Z_2 &= -(x/2)\sin(\alpha/2) + (y/2)\cos(\theta/2) \\ Z_3 &= \sin(\theta/2) \\ Z_4 &= \cos(\theta/2) \end{aligned} \tag{5.5}$$

These four components can be identified as coordinates in the projective three space $\mathbf{P}^3$ called the *Image Space* of planar kinematics. For the purpose of visual comparison, we project an image point back to Euclidean three-space

E^3 by

$$\begin{aligned} Z_1/Z_4 &= [(x/2)\cos(\theta/2) + (y/2)\sin(\theta/2)]/\cos(\theta/2) \\ Z_2/Z_4 &= [-(x/2)\sin(\theta/2) + (y/2)\cos(\theta/2)]/\cos(\theta/2) \\ Z_3/Z_4 &= \sin(\theta/2)/\cos(\theta/2) \\ Z_4/Z_4 &= 1 \end{aligned} \tag{5.6}$$

Thus, a planar motion can be graphically represented by a series of points in E^3 with coordinate axes $(Z_1/Z_4, Z_2/Z_4,\ Z_3/Z_4)$.

5.3 Harmonic Analysis of the Coupler Motion

In this section, we give the Fourier decomposition of translational and rotational component of the four-bar coupler motion. With the aid of their harmonic representations, there exists a clear-cut relationship between their Fourier descriptors, which serves as the basis for design space decoupling and therefore leads to drastic improvement in optimization routine for four-bar linkage synthesis.

Harmonic Content of the Rotation of the Coupler Link Consider a planar four-bar linkage shown in Fig. 5.2 with XOY being the fixed coordinate frame. The fixed pivot A_0 is located at point (x_0, y_0) with A_0B_0 being the ground link and A_0A the input link. Let l_i denote the length of the ith link and and θ_i the angle measured from the X axis of the fixed frame. Let ϕ, λ and ψ be the angles of link A_0A, AB, B_0B as measured from the ground link A_0B_0, respectively.

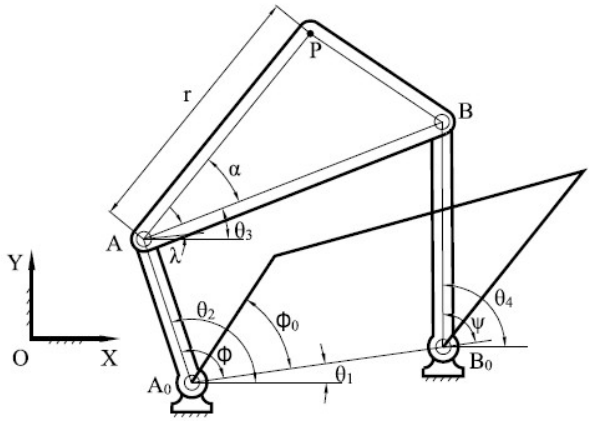

Figure 5.2: A four-bar mechanism.

Assume that the input link rotates with constant angular velocity ω, we have

$$\phi = \omega t + \phi_0 \tag{5.7}$$

where ϕ_0 is the initial input angle. It is well known that the coupler angle λ depends only on the input angle ϕ as well as link ratios

$$l_{21} = l_2/l_1, \quad l_{31} = l_3/l_1, \quad l_{41} = l_4/l_1 \tag{5.8}$$

This suggest that one may extract link ratios from the harmonic content of the rotational motion.

Using loop closure equations, it has been shown in [87] that the coupler angle λ is given by

$$e^{i\lambda} = \frac{-B(\phi) \pm \sqrt{\Delta_1(\phi)\Delta_2(\phi)}}{2A(\phi)} \tag{5.9}$$

where

$$A(\phi) = l_{31}(l_{21}e^{-j\phi} - 1) \tag{5.10}$$

$$B(\phi) = 1 + l_{21}^2 + l_{31}^2 - l_{41}^2 - 2l_{21}\cos\phi \tag{5.11}$$

$$\Delta_1(\phi) = 1 + l_{21}^2 - (l_{31} + l_{41})^2 - 2l_{21}\cos\phi \tag{5.12}$$

$$\Delta_2(\phi) = 1 + l_{21}^2 - (l_{31} - l_{41})^2 - 2l_{21}\cos\phi \tag{5.13}$$

and the sign $\pm$ correspond to the two configurations of the four-bar linkage for each input angle. Eq. (5.9) leads to the following well-known feasibility condition:

$$\Delta_1(\phi)\Delta_2(\phi) \leq 0. \tag{5.14}$$

The input link is a crank if this inequality holds for all $\phi \in [0, 2\pi]$; otherwise, it is a rocker.

Fourier series representation of $e^{i\lambda}$ is given as follow

$$e^{i\lambda} = \sum_{k=-\infty}^{\infty} C_k e^{ik\phi} = \sum_{k=-\infty}^{\infty} C_k e^{ik\phi_0} e^{ik\omega t} \tag{5.15}$$

The coefficients $C_k e^{ik\phi_0}$ are calculated by least square fitting method (see [87] for detail).

Harmonic Content of the Path of a Coupler Point Let us consider Fourier representation of the coupler point path of a four-bar mechanism. Let $A_0 = x_0 + iy_0$ be the complex number specifying the fixed pivot A_0 and let $z = re^{i\alpha}$ represent the position of point P with respect to the coupler link AB. The position of P with respect to global frame XOY is given by

$$P = A_0 + l_2 e^{i\theta_2} + ze^{i\theta_3} = A_0 + l_2 e^{i\theta_1} e^{i\phi} + ze^{i\theta_1} e^{i\lambda} \tag{5.16}$$

The path P can also be described by Fourier series as

$$P = \sum_{k=-\infty}^{\infty} P_k e^{ik\omega t} \tag{5.17}$$

Equating the right sides of both (5.16) and (5.17), after the substitution of (5.15) into (5.16), we obtain

$$P_0 = ze^{i\theta_1}C_0 + A_0 \qquad k = 0 \tag{5.18}$$

$$P_1 = ze^{i\theta_1}C_1e^{i\phi_0} + l_2e^{i(\theta_1+\phi_0)} \qquad k = 1 \tag{5.19}$$

$$P_k = ze^{i\theta_1}C_ke^{ik\phi_0} \qquad k \neq 0, 1 \tag{5.20}$$

Condition For Coupling Rotation With Translation As a planar motion decomposes into the motion of a point and rotation motion about that point, one may ask what coupling conditions the two motion components should satisfy in order for the given motion to be a four-bar motion? Let T_k $(k = -p, \ldots, p)$ denote the complex-number Fourier descriptors of the point trajectory and let Q_k $(k = -p, \ldots, p)$ denote those of the orientation trajectory. In view of (5.20), we have

$$T_k = P_k = ze^{i\theta_1}C_ke^{ik\phi_0}, \quad k \neq 0, 1 \tag{5.21}$$

In view of Eq. (5.15), we have $Q_k = C_ke^{ik\phi_0}$ for all k. This leads to the following relationship between T_k and Q_k:

$$T_k = ze^{i\theta_1}Q_k = re^{i(\alpha+\theta_1)}Q_k, \qquad k \neq 0, 1 \tag{5.22}$$

It follows that

$$|T_k| = r|Q_k| \qquad k \neq 0, 1 \tag{5.23}$$

where $r = |z|$ is the distance between the coupler point and the moving pivot A. We now have the following theorem regarding coupling of the harmonic contents of rotation and translation components of a planar four-bar motion:

Theorem: *For a given motion to be compatible with a planar four-bar motion, Fourier descriptors of its rotational and translational components must be linearly proportional:*

$$\cdots \frac{|T_{-3}|}{|Q_{-3}|} = \frac{|T_{-2}|}{|Q_{-2}|} = \frac{|T_{-1}|}{|Q_{-1}|} = \frac{|T_2|}{|Q_2|} = \frac{|T_3|}{|Q_3|} \cdots \tag{5.24}$$

If a task motion satisfies such a relationship, we can find a four-bar linkage that can generate the given motion. In general, a task motion may satisfy (5.24) only approximately so we define the following approximation error:

$$I = \sum_{k \neq 0,1} (|T_k| - r|Q_k|)^2 \tag{5.25}$$

As long as I is limited to very small value, the given motion can be generated approximately by a four-bar mechanism. In order to minimize I, r must satisfy

$$r = \frac{\sum_{k \neq 0,1} |T_k|}{\sum_{k \neq 0,1} |Q_k|} \tag{5.26}$$

5.4 Fourier Descriptor Based Synthesis Method

In this section, we seek to match or best approximate a task motion with a four-bar motion in terms of their Fourier Descriptors.

Match of Fourier Descriptors of Rotation For matching orientation, the given motion has orientation function specified as $e^{j\beta(t)}$, which can be approximated by Fourier Descriptors as

$$e^{j\beta(t)} \approx \sum_{k=-p}^{+p} Q_k e^{j\omega t} \tag{5.27}$$

In view of Eq. (5.15), we have

$$Q_0 = C_0 \qquad k = 0 \tag{5.28}$$

$$Q_k = C_k e^{jk\phi_0} \qquad k \neq 0 \tag{5.29}$$

Match of Fourier Descriptors of Translation A point path, $x(t) + y(t)$, can be approximated by Fourier Descriptors as

$$x(t) + y(t)j \approx \sum_{k=-p}^{p} T_k e^{jk\omega t} \tag{5.30}$$

In view of Eq. (5.17), we require

$$T_0 = Q_0 r e^{j(\alpha+\theta_1)} + x_0 + y_0 i \qquad k = 0 \tag{5.31}$$

$$T_1 = Q_1 r e^{j(\alpha+\theta_1)} + l_2 e^{j\theta_1} e^{j\phi_0} \qquad k = 1 \tag{5.32}$$

$$T_k = Q_k r e^{j(\alpha+\theta_1)} \qquad k \neq 0, 1 \tag{5.33}$$

The Synthesis Method From Eqs. (5.31) to (5.33), we can see the problem of path synthesis for a four-bar mechanism involves ten design variables $\{\phi_0, l_1, l_2, l_3, l_4, x_0, y_0, \theta_1, r, \alpha\}$. This can be separated into two groups. One is $S_1 = \{\phi_0, l_{21}, l_{31}, l_{41}, \mathcal{M}, \mathcal{N}\}$, where

$$\mathcal{M} = r\cos(\alpha + \theta_1), \quad \mathcal{N} = r\sin(\alpha + \theta_1). \tag{5.34}$$

The other group is $S_2 = \{l_1, x_0, y_0, \theta_1\}$. The set of variables S_1 is used to match the high harmonics of the point path. From Eqs. (5.28) to (5.29), the problem of orientation synthesis involves only four design variables $S_3 = \{\phi_0, l_{21}, l_{31}, l_{41}\}$. Thus we have the following algorithm:

1. Use the set S_3 to synthesize the orientation.

2. Then use the set S_1 to synthesize high harmonics part of path. Because $S_3 \in S_1$ and S_3 is acquired in step one, that leaves $S_1 - S_3 = \{\mathcal{M}, \mathcal{N}\}$ to match the high harmonics of path. It means we may sacrifice certain accuracy in path matching for orientation match.

3. Now we have the value of S_1. Together with S_2, we can match the low harmonics part of path.

Based on the work by Freudenstein [42] and our numerical experiments, we notice that the Fourier descriptor decreases asymptotically as its order goes higher if the four-bar is a crank-rocker or double-crank mechanism. This means that the loss in accuracy for high harmonics is very small. Fig. 5.3 shows the distribution of Fourier descriptors of various harmonics from 1000 test runs of $\{l_{21}, l_{31}, l_{41}\}$ for crank-rocker and double-crank mechanisms. It verifies our claim that the magnitudes of high order harmonics decrease fast and asymptotically. Only terms of $-2, -1$ and 2 have relatively large effect on path matching and other terms are negligible.

We now investigate this issue analytically by deriving a representation of the coupler angle λ in a form that is similar to the result in Freudenstein [42]

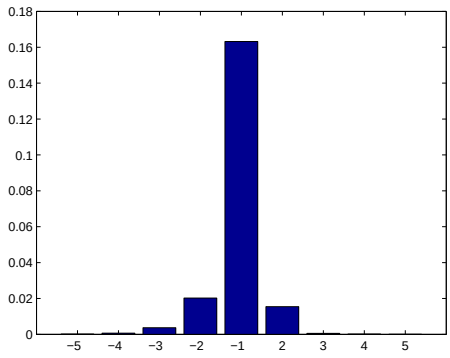

Figure 5.3: Fourier descriptors of a four-bar linkage at $k\omega$ where k=-5,-4,-3,-2,-1, 2, 3, 4, 5.

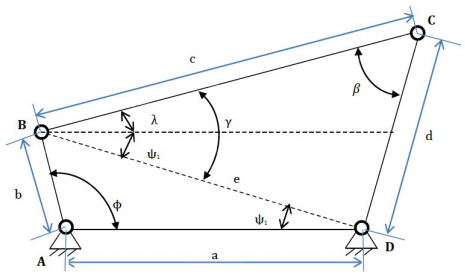

Figure 5.4: Four-bar linkage with ground link AD, crank AB, coupler link BC and output link CD. $AB = b, BC = c, CD = d, AD = a, BD = e$.

to provide a proof that the magnitudes of the Fourier coefficients associated with λ decrease as their order increases. We use notations shown in Fig. 5.4, to relate the coupler angle λ and the input angle ϕ. Apply the laws of cosines and sines to the triangle BCD in Fig. 5.4 to obtain

$$\frac{\sin\gamma}{d} = \frac{\sin(\pi-\beta)}{e} = \frac{\sin\beta}{e}$$

$$\cos\gamma = \frac{c^2+e^2-d^2}{2ce}$$

This leads to

$$\tan\gamma = \frac{2cd\sin\beta}{c^2+e^2-d^2}. \tag{5.35}$$

For the triangle ABD, we have

$$e^2 = a^2+b^2+2ab\cos\phi. \tag{5.36}$$

Substitute e in Eq. (5.35) with Eq. (5.36) to obtain

$$\tan\gamma = \frac{2cd\sin(\beta)}{c^2+a^2+b^2+2ab\cos(\phi)-d^2} \tag{5.37}$$

Furthermore, we have

$$\cos\beta = \mu_1+\mu_2\cos\phi \tag{5.38}$$

where $\mu_1 = (a^2+b^2-c^2+d^2)/2cd$ and $\mu_2 = ab/cd$. Thus, Eq. (5.37) is simplified to

$$\tan\gamma = \frac{\delta\sin\beta}{1+\delta\cos\beta} = \frac{\delta\sqrt{1-(\mu_1+\mu_2\cos\phi)^2}}{1+\delta(\mu_1+\mu_2\cos\phi)} \tag{5.39}$$

where $\delta = cd/(c^2-d^2)$. Similarly, we can show that

$$\tan\psi_1 = \frac{\rho\sin\phi}{1+\rho\cos\phi}, \qquad \rho = \frac{b}{a} \tag{5.40}$$

Finally, the coupler angle λ can be expressed directly in terms of the input angle ϕ as

$$\begin{aligned} \lambda &= \gamma-\psi_1 \\ &= \arctan\left(\frac{\delta\sqrt{1-(\mu_1+\mu_2\cos\phi)^2}}{1+\delta(\mu_1+\mu_2\cos\phi)}\right) - \arctan\left(\frac{\rho\sin\phi}{1+\rho\cos\phi}\right) \end{aligned} \tag{5.41}$$

The right-hand side of Eq. (5.41) can be expanded into a Fourier series in ϕ. Based on Freudenstein [42], we conclude that the resulting Fourier coefficients decrease as their order increases.

The Synthesis Algorithm To find a least squares solution to Eq.(5.33), we define the following error function for path error:

$$I_1 = \sum_{k\neq 0,1} \|C_k r e^{j(\alpha+\theta_1+k\phi_0)} - T_k\|^2 \tag{5.42}$$
$$= \sum_{k\neq 0,1} [(A_k \cdot \mathcal{M} - B_k \cdot \mathcal{N} - T_{kx})^2 + (A_k \cdot \mathcal{N} + B_k \cdot \mathcal{M} - T_{ky})^2]$$

where $T_k = T_{kx} + iT_{ky}$, $C_k e^{ik\phi_0} = A_k + iB_k$, and $\mathcal{M}, \mathcal{N}$ are given by (5.34). To minimize I_1, it is required that

$$\frac{\partial I_1}{\partial \mathcal{M}} + \frac{\partial I_1}{\partial \mathcal{N}} j = 0 \tag{5.43}$$

which results in

$$\mathcal{M} + j\mathcal{N} = \frac{\sum_{k\neq 0,1} T_k(A_k - B_k)}{\sum_{k\neq 0,1} |A_k^2 + B_k^2|} \tag{5.44}$$

In addition, we express the set of S_2 in Eqs. (5.31) and (5.32) by the set of S_3

$$x_0 + y_0 j = T_0 - C_0(\mathcal{M} + j\mathcal{N}) \tag{5.45}$$

$$l_1 e^{j\theta_1} = T_1 e^{-j\phi_0} - C_1(\mathcal{M} + j\mathcal{N}) \tag{5.46}$$

To find a least squares solution to Eqs. (5.28) and (5.29), we define the following error function for the orientation error:

$$I_2 = \sum_{k=-p}^{+p} \|C_k e^{ik\phi_0} - Q_k\|^2 \tag{5.47}$$

The path error I_1 and the orientation error I_2, taken together, define the motion approximation error in terms of Fourier descriptors of the given motion and desired motion.

The detailed algorithm is presented as follows

1. The task motion is approximated with Fourier Descriptors $\{T_k, Q_k\}$ using Eqs. (5.30) and (5.27), both including the basis functions to the pth order.

2. In order to match orientation, we initiate the search for $S_3 = \{\phi_0, l_{21}, l_{31}, l_{41}\}$ where $\phi_0 \in [0, 2\pi]$. As shown in [87], this can be done either with a direct search or more efficiently with a simulated annealing algorithm. We can set the maximum link ratio to be K_{max} and require

$$\frac{l_n}{l_1} \in [\frac{1}{K_{max}},\ K_{max}], \quad n \in [2, 3, 4] \tag{5.48}$$

3. Examine the feasibility condition for current S_3 using Eq. (5.14). If Eq. (5.14) is satisfied by all values of the angle ϕ in $[0, 2\pi]$, go to Step 4; Otherwise, go back to Step 2 for generating a new group of S_3. By doing this, we assume that the desired linkage is either a crank-rocker or double-crank mechanism.

4. Evaluate the error function I_2 and store the value of I_2.

5. Search for the minimum value of the error function I_2. The corresponding values of S_3 yield the orientation that best approximates the given motion.

6. Compute $\mathcal{M} + j\mathcal{N}$ by Eq. (5.44) and then evaluate the error function I_1. This results in values for all six elements of S_1.

7. Solve for S_2 with Eqs. (5.45) and (5.46).

8. The resulting values of $S = S_1 \bigcup S_2$ produces a design for a four-bar mechanism such that its coupler motion approximates a task motion.

5.5 Examples

In this section, we present three examples for four-bar linkage synthesis using the Fourier Descriptor based algorithms. In the first two examples, the former task motion is a closed coupler motion generated from a known four-bar linkage and the latter is a closed motion generated by sinusoidal functions. In the last example, the task motion is an open motion generated by certain functions.

Example 1: Closed Motion First presented is an example such that the given motion is an ideal motion generated from known dimensions of four-bar linkage. This is to verify if our algorithm can recover the given four-bar mechanism. The result is positive, as shown in Table 5.1 and Fig 5.5.

	Task motion	Synthesized motion
x_0	5	5.0057
y_0	6	6.0358
θ_1	0.1745	0.1877
l_1	11	11.1535
l_2	6	6.0229
l_3	8	8.0305
l_4	10	10.2612
r	7	7.0013
α	0.6981	0.7022
ϕ_0	0.7854	0.7679

Table 5.1: Comparison between parameters of a given four-bar linkage and those of the synthesized four-bar linkage

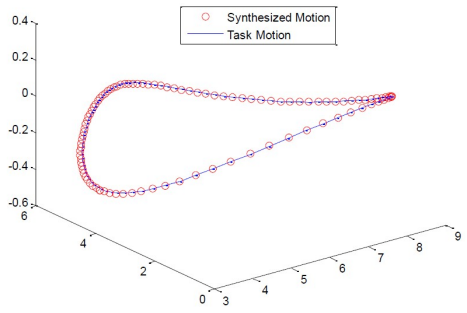

Figure 5.5: The graph of task motion and synthesized motion in the image space. The structural error I_1 is 3.7184×10^{-4} and I_2 is 5.5131×10^{-5}.

Example 2: Closed Motion In this example, the task motion is given analytically as

$$x_c = 40\cos\theta, \quad y_c = 45\sin\theta, \quad \phi_c = 25\sin(\theta/2)$$

where $\theta = k\pi/180$ with $k = 1, 2, \cdots, 360$ in degree, i.e., the task motion is constrained such that one point moves on an ellipse and its angular position follows a sinusoidal function. The result is shown in Fig. 5.6, which indicates a reasonably good match.

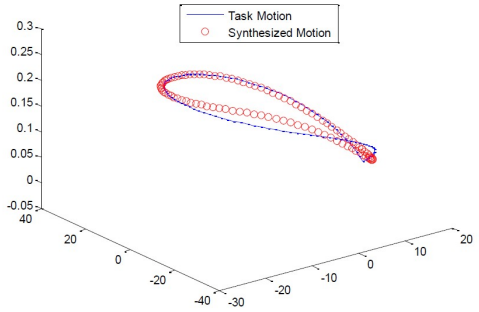

Figure 5.6: The graph of task motion and synthesized motion in the image space. The structural error I_1 is 0.5244 and I_2 is 5.6873×10^{-4}.

Example 3: Open Motion Now consider the case where only a segment of motion is to be followed by a four-bar linkage. Here, we assume that the input link of the four-bar motion is a crank that can take up a full rotation. When approximating the given open motion, only part of the coupler motion would be used. The given motion is a segment of the following

$$x_c = 19\cos\theta, \quad y_c = 21\sin\theta, \quad \phi_c = \frac{1}{10}\theta(\theta - 2\pi). \tag{5.49}$$

We consider four motion segments as defined by the angular ranges $\theta \in [0, \zeta]$ where $\zeta = 0.8\pi, 0.9\pi, \pi, 1.1\pi$, respectively. The results are shown Figs. 5.7, 5.8, 5.9 and 5.10 respectively. Corresponding design parameters are listed in Table 5.2. Among these four motion segments, $\zeta = \pi$ give us the best approximation to the task motion. Path errors I_1 of the four cases are of order 10^{-1}, which are quite reasonable. However, while the orientation error for $\zeta = \pi$ is of order 10^{-4}, the orientation error for $\zeta = 1.1\pi$ is significant as shown in Fig. 5.10.

ζ	0.8π	0.9π	1π	1.1π
x_0	-0.9571	-1.1133	-1.2760	0.9534
y_0	0.2236	0.3326	0.9249	2.0733
θ_1	-2.0224	-2.0572	-2.0858	-1.0754
l_1	21.7798	21.7450	24.9624	34.3784
l_2	19.6018	19.5705	19.9699	20.6270
l_3	23.9578	26.0940	32.4511	48.1297
l_4	26.1358	26.0940	29.9549	34.3784
r	1.4777	1.6179	1.9623	2.5630
α	2.8105	2.7195	2.3876	-1.5925
ϕ_0	2.0595	2.0944	2.1293	1.0472

Table 5.2: The design parameters of synthesized four-bar linkages

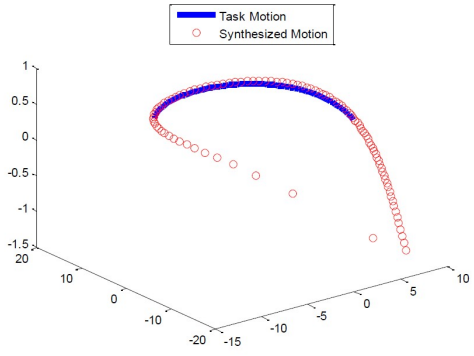

Figure 5.7: The graph of task motion and synthesized motion in the image space. The structural error I_1 is 0.5286 and I_2 is 0.0048. $\zeta = 0.8\pi$

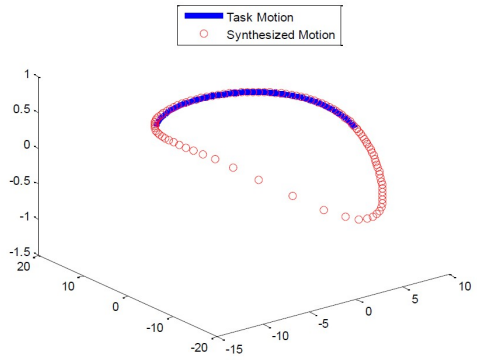

Figure 5.8: The graph of task motion and synthesized motion in the image space. The structural error I_1 is 0.4918 and I_2 is 0.0118. $\zeta = 0.9\pi$

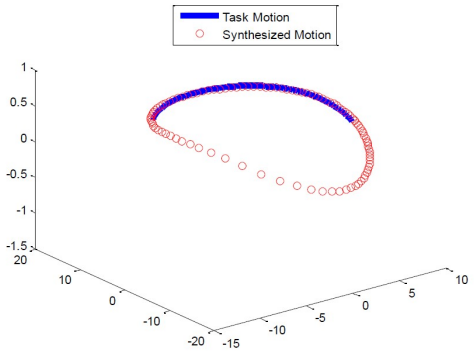

Figure 5.9: The graph of task motion and synthesized motion in the image space. The structural error I_1 is 0.4041 and I_2 is 7.41×10^{-4}. $\zeta = 1\pi$

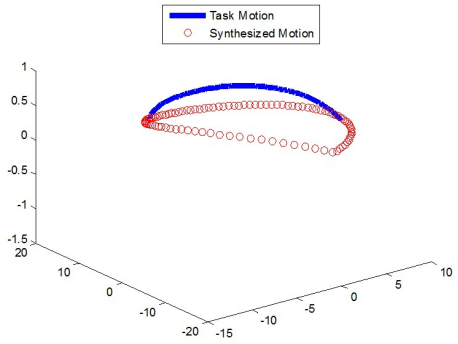

Figure 5.10: The graph of task motion and synthesized motion in the image space. The structural error I_1 is 0.3529 and I_2 is 0.1161. $\zeta = 1.1\pi$

5.6 Conclusions

In this chapter, we apply the Fourier Descriptor method together with least square fitting to dimensional synthesis of four-bar mechanism for best approximating closed or open task motion. Because a crank-rocker or double crank mechanism has a pattern that magnitudes of high harmonics content in coupler point path decreases asymptotically, we can match orientation and point path separately without causing a significant deviation in path error.

Chapter 6

Parametrization-independent Non-uniform Fourier Approach to Path Synthesis of Mechanism

This chapter deals with the classical problem of dimensional synthesis of planar four-bar linkages for path generation. Using Fourier Descriptors, a given path is represented by harmonic series. Extensive research has been done on this approach in mechanism synthesis with constraints that the path is given a prescribed parametrization or timing beforehand and that the input link should rotate with constant angular velocity. Little research effort has been put into pure path synthesis independent of parametrization. Therefore, we present an exact method that can efficiently and accurately carry out the pure path matching using arc-length parametrization. Meanwhile, curve normalization combined with artificial neural network is used to decompose the search space, which leads naturally to a fast synthesis approach.

When processed by Fourier Transform, different parametric forms of a task curve would yield different Fourier descriptors. In Fig. 6.1, the unit circle is

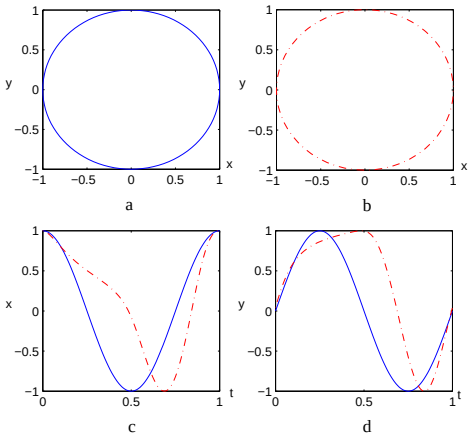

Figure 6.1: Solid line: paramterization 1. Dash line: parametrization 2. a) Shape of unit circle in parametrization 1; b) Shape of unit circle in paramterization 2; c) x component of paramterization function z(t) under two different parametrizations; d) y component of paramterization function z(t) under two different paramterizations

assigned two different parametrization $z_1(t)$ and $z_2(t)$. The Fourier transform of $z_1(t)$ is $z_1(t) = e^{j2\pi t}$ while $z_2(t)$ as:

$$\begin{aligned} z_2(t) &= (0.1607 + 0.3138i) \\ &+ (-0.0759 + 0.0285i)e^{-j2\pi t} + (0.6117 - 0.6020i)e^{j2\pi t} \\ &+ (0.0053 - 0.0018i)e^{-j4\pi t} + (0.3111 + 0.1758i)e^{j4\pi t} \\ &+ (-0.0045 + 0.0019i)e^{-j6\pi t} + (-0.0090 + 0.0674i)e^{j6\pi t} \end{aligned} \tag{6.1}$$

It is clear that two sets of Fourier Descriptors are completely different from each other, though they both define the same geometric curve, i.e., the unit circle.

As with task curve, the mechanism coupler curve shares the same problem. When the crank rotates, a curve would be traced out by coupler point of the mechanism. For a mechanism of fixed dimensions, different crank rotation functions could lead to different parametrization or timing for the coupler curve. Traditional ways of path synthesis approaches (see Chu [55] and Wu [87]) assume that crank always rotates with constant angular velocity and hence predetermine the timing of coupler curve. As a consequence, a best coupler curve could be missed if its Fourier Descriptors, under that predetermined timing, don't match those of task curve. In view of the timing issue, it is suggested that matching of task and coupler curve be implemented in terms of same parametrization.

In this chapter, we use arc-length parametrization to unify the parametrization for both task and four-bar coupler curve. Arc-length parametrization is a parametrization based on the inherent property of curve: arc length. When comparing two curves by their Fourier Descriptors, we could reparameterize both of them using arc length as parameter and then initiate the comparison. However, direct arc-length parametrization for coupler curve of four-bar mechanism requires determination of ten design variables simultaneously, which would incur tremendous computational cost. In Wu's [87] and Chu's [55] method, the ten-design-parameter synthesis problem is converted to a four-design-parameter problem because of the assumption that crank rotates constantly. In our case, we need to find different way to reduce the design cost. Considering that a curve has position, orientation, size and shape, we can first

match the shape of task and four-bar coupler curve by curve normalization [88]. The process of curve normalization takes out information about curve's position, orientation and size and merely keep the shape of curve. In later section, it will be shown that the shape of four-bar coupler curve only depends on three link ratios and the choice of coupler point. Therefore, we try to match Fourier Descriptors of four-bar coupler curve after curve normalization with that of task curve after curve normalization, both under arc-length parametrization, so as to find link ratios and the choice of coupler point. Because we record the position, orientation and size of the task curve along its normalization process, those data are used to find the other five design variables for four-bar mechanism. In order to efficiently determine link ratios and choice of coupler point, an Artificial Neural Network (ANN) is trained to establishing the relationship between them and the Fourier Descriptors of normalized coupler curve under arc-length parametrization.

The organization of the chapter is as follows. Section 6.1 gives a detailed Fourier Analysis of the four-bar mechanism. The difference from Chapter 5 is that there's no assumption of constant angular velocity for crank rotation. Section 6.2 presents a new way of decoupling design space via curve normalization. Section 6.3 discusses on arc-length parametrization and Artificial Neural Network. Section 6.4 gives results and discussion of our approach to justify its accuracy in pure path generation, together with comparison with other established Fourier-based path generation algorithms.

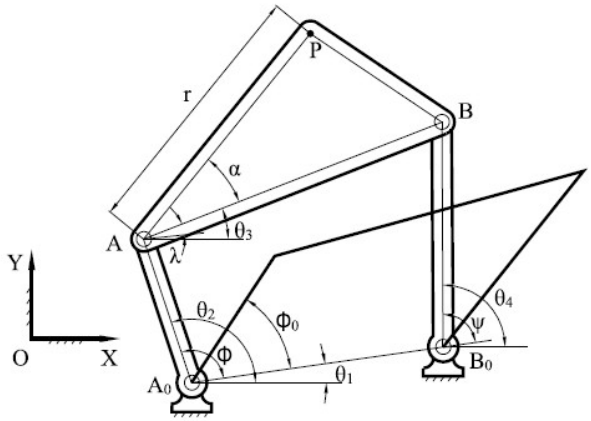

Figure 6.2: A four-bar mechanism.

6.1 Fourier Analysis of the Four-bar Mechanism

Consider a planar four-bar linkage shown in Fig. 6.2 with XOY being the fixed coordinate frame. The fixed pivot A_0 is located at point (x_0, y_0) with A_0B_0 being the ground link and A_0A the input link. Let l_i denote the length of the ith link and and θ_i the angle measured from the X axis of the fixed frame. Let ϕ, λ and ψ be the angles of link A_0A, AB, B_0B as measured from the ground link A_0B_0, respectively.

Assume that the input link rotates with angular velocity of function $\omega(t)$, we have

$$\phi = t\omega(t) + \phi_0 \qquad t \in [0, 1] \tag{6.2}$$

where ϕ_0 is the initial input angle. Here $\omega(t)$ suggests that the angular velocity

may be inconstant, thus allowing various parametrizations.

As in the chapter 4 and 5, the coupler angle λ is given by

$$e^{i\lambda} = \frac{-B(\phi) \pm \sqrt{\Delta_1(\phi)\Delta_2(\phi)}}{2A(\phi)} \tag{6.3}$$

where

$$\begin{aligned} l_{21} &= l_2/l_1, \quad l_{31} = l_3/l_1, \quad l_{41} = l_4/l_1 && (6.4)\\ A(\phi) &= l_{31}(l_{21}e^{-j\phi} - 1) && (6.5)\\ B(\phi) &= 1 + l_{21}^2 + l_{31}^2 - l_{41}^2 - 2l_{21}\cos\phi && (6.6)\\ \Delta_1(\phi) &= 1 + l_{21}^2 - (l_{31} + l_{41})^2 - 2l_{21}\cos\phi && (6.7)\\ \Delta_2(\phi) &= 1 + l_{21}^2 - (l_{31} - l_{41})^2 - 2l_{21}\cos\phi && (6.8) \end{aligned}$$

and the sign $\pm$ correspond to the two configurations of the four-bar linkage for each input angle.

Now let us consider Fourier representation of the coupler curve of a four-bar mechanism. Let $A_0 = x_0 + iy_0$ be the complex number specifying the fixed pivot A_0 and let $z = re^{i\alpha}$ represent the position P with respect to the coupler link AB. The position of the coupler point relative to global frame XOY can be represented as

$$P = A_0 + l_2e^{i\theta_2} + ze^{i\theta_3} = A_0 + l_2e^{i\theta_1}e^{i\phi} + ze^{i\theta_1}e^{i\lambda} \tag{6.9}$$

Ten design variables $\{l_1, l_2, l_3, l_4, x_0, y_0, \theta_1, r, \alpha, \phi_0\}$ are encapsulated in the above equation. Also, P can be described by Fourier series as

$$P = \sum_{k=-\infty}^{\infty} \beta_k e^{ik2\pi t} \qquad t \in [0, 1] \tag{6.10}$$

Note that constant angular velocity for crank rotation cannot be presumed and separation of design variables like Eq. (5.18)-Eq. (5.20) in chapter 5 is impossible. If we were to match task curve FDs α_k directly with coupler curve FDs β_k, we would have to simultaneously search aforementioned ten design variables, compute P based on Eq. (6.9) and obtain β_k from Eq. (6.10) by Fourier transform to see whether $\alpha_k = \beta_k$. Obviously, such brutal-force searching would consume impractical amount of time. For the sake of decreasing synthesis cost, design space decomposition must be carried out. In next section, curve normalization is introduced to efficiently reduce the cost by matching Fourier Descriptors of the normalized task and four-bar coupler curve. As a result, three link ratios and choice of coupler point can be determined separately from the other five design variables.

6.2 Decoupling of Design Variables

In this section, we will introduce curve normalization.

Ten design variables $\{l_1, l_2, l_3, l_4, x_0, y_0, \theta_1, r, \alpha, \phi_0\}$ are here noted through previous discussion. By look at those design variables, we found that only l_1, l_2, l_3, l_4, r and α determine the shape and size of coupler curve. x_0 and y_0 play a role in translating the coupler curve as a whole; θ_1 rotates the coupler curve as whole; ϕ_0 decides the starting point of coupler curve, which has nothing to do with shape and size. Another reason to ignore ϕ_0 is that Fourier transform have a property of shift-invariance, i.e., it does not depend on the starting

point. Therefore, Eq. (6.9) can be split into two parts:

$$P = P_1 + P_2 e^{i\theta_1} \tag{6.11}$$

where $P_1 = A_0 = x_0 + y_0$ and $P_2 = l_2 e^{i\phi} + z e^{i\lambda}$. P_1 is only relevant in deciding the position of the coupler curve inasmuch it translates the whole curve by a vector of (x_0, y_0). ϕ_0 in P_2 can be chosen according to user demand because it is only related to the starting point of a coupler curve and thereby just let ϕ_0 be 0. P_2 represents the shape and actual size of coupler curve. Furthermore, we can divide P_2 by l_1 to get $\overline{P}_2 = l_2/l_1 e^{i\phi} + z/l_1 e^{i\lambda}$, which still keeps the shape of coupler curve. Then z is expressed as $x_c + y_c$ instead of $re^{i\alpha}$. Finally, in view of Eq. (6.3), the shape of coupler curve is determined by five design variables $\{l_{21}, l_{31}, l_{41}, \frac{x_c}{l_2}, \frac{y_c}{l_2}\}$. For clear representation of coupler point coordinate (x_c, y_c) on coupler link l_3, $\frac{x_c}{l_1}$ and $\frac{y_c}{l_1}$ are changed to $\frac{x_c}{l_3}$ and $\frac{y_c}{l_3}$ respectively and five design variables become:

$$\{l_{21}, l_{31}, l_{41}, \frac{x_c}{l_3}, \frac{y_c}{l_3}\} \tag{6.12}$$

The above process gives us the advantage that only have five design variables needs to be searched while the shape of curve remains untouched. However, the position, orientation and size of the coupler curve are changed. If we were to match it with the task curve, curve normalization has to be used to transform both task curve and the new coupler curve $\bar{P}_2$ into their canonical configuration. After normalization, both curves would be within a bounding unit rectangle and thereby pure shape match ensues.

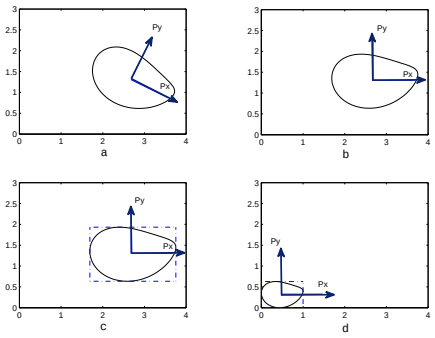

Figure 6.3: Curve normalization process. in a), we have the original curve; in b), the curve is rotated around its center to align its major and minor principal axis of inertia of moment with x-axis and y-axis of fixed frame respectively; in c), a bounding box with width w and height h is produced to tightly confine the curve and w/h is determined merely by the shape of the curve; in d), we resize the curve with its bounding box so that width of bounding box becomes 1 and height w/h and then relocate the bounding box to the origin of fixed frame.

Curve Normalization Curve normalization procedure was first proposed by Dikabar and Mruthyunjaya [88] and later used by Sánchez Marín and Pérez González [90]. Recently, Galán-Marín et al. [91] applied it with wavelet descriptors approach to efficiently synthesize crank-rocker mechanism constrained by optimal transmission angle. Principle of this procedure is that we treat the closed curve along which mass uniformly distributed. Through curve normalization, the major principal axis of moment of inertia is aligned with the x-axis; then, the width w and height h of the bounding box of the curve is evaluated; finally, the curve with bounding box is scaled by a factor of $1/w$ and translated to be located at origin. Here we say the curve is in its canonical configuration. Please refer to Appendix for the details of curve

normalization procedure.

Though the process of curve normalization changes the position, orientation and size of curve, it holds the shape unchanged. As mentioned previously, shape exclusively depends on $\{l_{21}, l_{31}, l_{41}, \frac{x_c}{l_3}, \frac{y_c}{l_3}\}$. As long as these parameters are known, we can compare task curve with four-bar curve in their canonical configurations, thus efficiently reducing the search space of design variables.

6.3 Numerical Synthesis

In this part, we explain how to find the five design variables in (6.12). Our method is to approximate the relationship between Fourier Descriptors of curve and those five design parameters. The Fourier Descriptors should come from the curve after being normalized and reparametrized with arc length as parameter. For the purpose of better accuracy, we add direct search method as the post-procesing stage for neural network to obtain final values of five design variables. Finally, we will use recovering approach, which recovers the position,orientation and size of task curve taken out during curve normalization, to obtain the values for the other five design variables.

Arc-length Parametrization As said earlier, a curve can have distinct parametrizations, which are called various representatives of the curve. We want to single out a unique representative of the curve in a geometrically significant way. This is done by referring a curve to its arc length as a parameter. A curve $\mathbf{x}(\mu)$ is said to be defined as a function of its arc length if the tangent

vector $\mathbf{x}'(\mu) = d\mathbf{x}/d\mu$ is a unit vector, $|\mathbf{x}'(\mu)| = 1$. Then, μ becomes the parameter of arc length s.

Now consider the task curve with its canonical configuration and assume the parametrization is given as continuous function of $\mathbf{z}(t) = x(t) + y(t)j$. Theoretically, the arc-length parametrization can be computed as following steps: **a**) by $s(t) = \int_0^t \|\mathbf{z}'(\mu)\| d\mu$, we get the arc length function s against t; **b**) compute $s^{-1}(t)$, the inverse function of $s(t)$, and we get time t function $t(s) = s^{-1}(t)$ against arc length; **c**) substitute $t(s)$ into $\mathbf{z}(t)$ and finally the arc length parametrization $\mathbf{z}(s)$ is obtained. However, it's impossible to derive an explicit formula for $s^{-1}(t)$.

Nonetheless, $\mathbf{z}(t)$ is usually given as a sequence of N points and thereby numerical approach can be used. Assume we have the sequence of $\mathbf{z}(t)$ taken as $\mathbf{z}(\frac{0}{N}), \mathbf{z}(\frac{1}{N}), ..., \mathbf{z}(\frac{N-1}{N})$. Next, we treat the curve as polygonal curve and compute the arc length as follows:

$$s(n) = \begin{cases} 0, & n = 0, \\ \sum_{k=1}^{n} \|\mathbf{z}(\frac{k}{N}) - \mathbf{z}(\frac{k-1}{N})\|, & n = 1, ..., N-1. \end{cases} \tag{6.13}$$

For the sequence of $\mathbf{z}(\frac{n}{N})$ $(n = 0, 1, \ldots, N-1)$, we obtain a corresponding sequence of $s(n)$ by Eq.(6.13). Therefore, we formulate the arc-length parametrization $\mathbf{z}_s(s) = \mathbf{z}_s(s(n))$, $0 \leq s(n) < L$ (L is the total length of curve). Applying Fourier transform (FT)to $\mathbf{z}_s(s)$ requires that the domain of $s(n)$ be $[0, 1]$. So we normalize $s(n)$ by a factor of $\frac{1}{L}$.

Artificial Neural Network Inspired by biological neural networks, an artificial neural network is a computational structure consisting of a collection

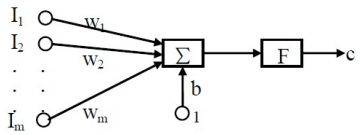

Figure 6.4: Typical neuron

of interconnected elements, known as neurons, to define a function [92, 93]. The network function is largely determined by the nature of the connections, which can be adjusted to map an input space to the corresponding desired output space.

Neuron is the elemental component of an artificial neural network (Fig. 6.4). The core of a neuron is a transfer function F which maps the sum of the weighted input $\mathbf{w}^T\mathbf{I}$ and the bias b to the output, i.e.

$$c = F(\mathbf{w}^T\mathbf{I} + b) \tag{6.14}$$

where $\mathbf{I}$ denotes the vector of all the inputs $\mathbf{I}_i$ to the neuron, and $\mathbf{w}$ denotes the vector of all the weights w_i of the connections between the inputs and the neuron. A neural network is formed by layers of neurons, where the outputs of one layer become the inputs of next layer. A typical neural network architecture is sown in Fig 6.5. The weights between the kth and $(k+1)th$ layers are defined in a weight matrix $\mathbf{W}^{k+1,k}$ whose (j,i) element represents the weight of the connection between the ith output of the $(k+1)th$ layer. Consequently the output vector of the $(k+1)th$ layer becomes

$$\mathbf{c}^{k+1} = \mathbf{F}^{k+1}(\mathbf{W}^{k+1,k}\mathbf{I}^{k+1}\mathbf{b}^{k+1}) \tag{6.15}$$

In order to let a neural network map an input space to the desired output space, the weights and biases are often adjusted through an iterative training process until the resulting outputs match the targeted outputs. The training process can be either incremental, where the weights and biases are updated after each input-target pair, or batch, where the weights and biases are updated after all the inputs and targets are provided.

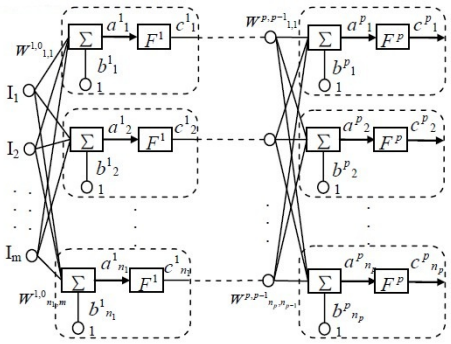

Figure 6.5: Typical neural network architecture

Here, we use the widely-adopted backpropagation (BP) algorithm to train a network to fit the input-output relationship embedded in the sample data. BP is a supervised learning method which fits a function based on samples of input-output data pairs. For each input vector, the algorithm estimates the error between the actual and desired network outputs, and backpropagates it from the output layer to hidden neurons to estimate the contribution of each hidden neuron to the output error. It calculates the gradient of each weight, which indicates the direction of error increase, and updates the weight in the

opposite direction of the gradient.

After curve normalization and reparametrization using arc length, we have the Fourier coefficients α_k, $k = -p, \ldots, p$. By the work of Li et al. [94], it is known that for any four-bar mechanism with a input crank, magnitudes of Fourier coefficients decrease noticeably as the order p goes higher. So we choose coefficients of order from -3 to 3, i.e., $\alpha_{-3,}\alpha_{-2}, \ldots, \alpha_2, \alpha_3$ to be Fourier Descriptors of a curve. In order to train the neural network, we produced 101700 sets of Fourier Descriptors of different four-bar curves generated by varying design variables (6.12) to train the neural network while using the other 101700 sets to test the validity of our train neural network to see whether it can effectively approximate the relationship between harmonic components and design variables (6.12).

In practical design, the ratio between any two links is not expected to be extremely large or small. So it is reasonable to predefine a max link ratio in the design process, say K_{max}. Correspondingly, the minimum link ratio is $\frac{1}{K_{max}}$. Therefore, we vary three link ratios l_{21}, l_{31}, l_{41} in $[\frac{1}{K_{max}}, K_{max}]$. Here, K_{max} is taken to be 6. Likewise, it is required that the coordinate of coupler point on coupler link needs to be dimensionally compatible with the length of coupler link. So we have similar constraints for $\frac{x_c}{l_3}$ and $\frac{y_c}{l_3}$ and K_{max} for either of them is 3. Hence, input data for training process are seven Fourier Descriptors and output data are five design variables.

Recovering Method At this moment, we know the values of five design variables $\{l_{21}, l_{31}, l_{41}, \frac{x_c}{l_3}, \frac{y_c}{l_3}\}$. From the discussion of section 4, $\overline{P}_2$ can be computed. According to Eq. (6.11), our goal is to find the four-bar curve P that best matches the task curve, say, T. Hence, we let $T = P$ and (6.11) becomes:

$$T = P_1 + P_2 e^{i\theta_1} \tag{6.16}$$

Next, there are three steps to go in order to match the size, orientation and position of T and P, during which x_0, y_0, θ_1, l_2 could be found

1. First, size match between T and P. Rotate T and $\overline{P}_2$ to align their major principal axis with x-axis of fixed frame respectively and denote transformed curves to be $\mathcal{R}(T)$ and $\mathcal{R}(\overline{P}_2)$. Compute the width or height of bounding box for $\mathcal{R}(T)$ and $\mathcal{R}(\overline{P}_2)$ and denote the ratio as w_1/w_2 or h_1/h_2, which is the size ratio between T and $\overline{P}_2$. According to Eq. (6.11), size of P is determined by P_2 and P_2 is equal to $l_1\overline{P}_2$. Therefore, $l_1 = w_1/w_2 = h_1/h_2$.

2. Second, orientation match between T and P. We obtain the value of P_2 at step 1 by $l_1\overline{P}_2$. According to Eq. (6.11), the orientation difference between P and P_2 lies on θ_1. Therefore, θ_1 is measured as the angle from the major principal axis of P_2 to that of T.

3. Finally, position match between T and P. Until now, we know $P_2 e^{i\theta_1}$. Then, we compute the center for T and $P_2 e^{i\theta_1}$ correspondingly and denote them as C_1 and C_2. The distance vector from C_2 to C_1 equals to P_1.

Up until now, values of nine design variables in (6.11) have been found except for ϕ_0. As said earlier, ϕ_0 only determines the starting point of curve and is irrelevant to position, orientation, size and shape of curve. In practical use, the starting point could be chosen in accordance with user demand. The flowchart synthesis algorithm is shown in Fig. 6.6 and 6.7.

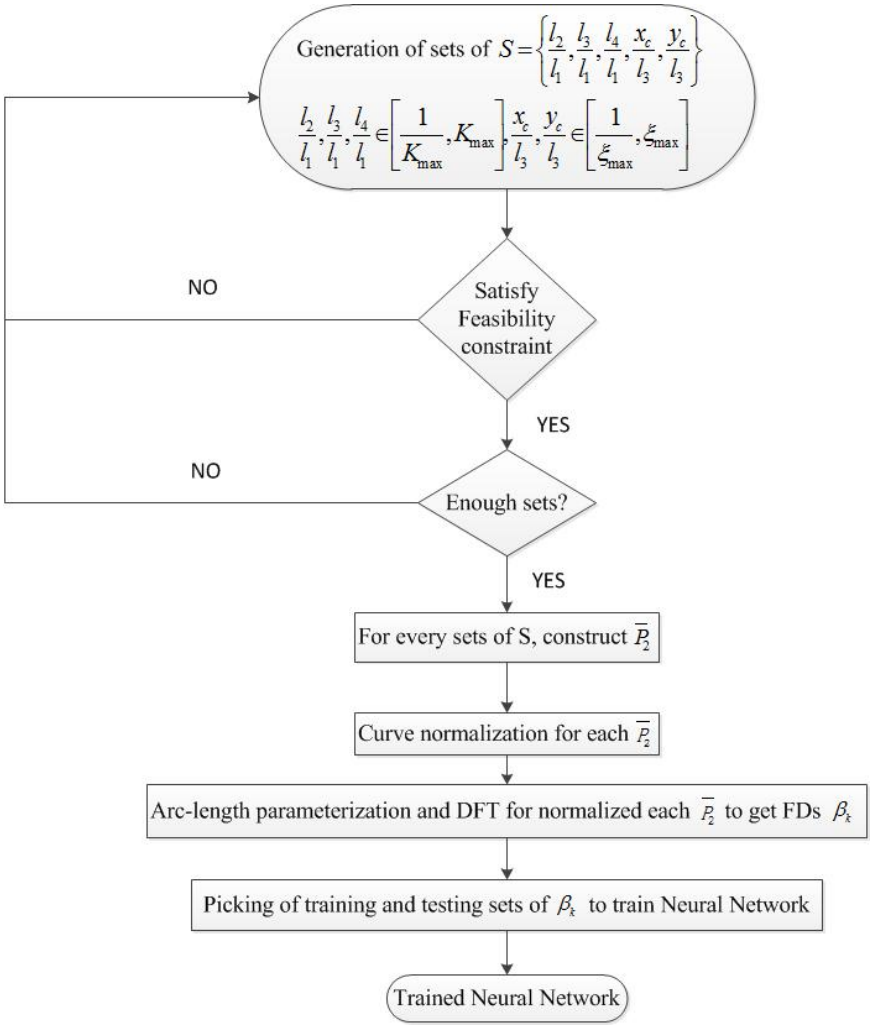

Figure 6.6: Synthesis Algorithm Flowchart (1)

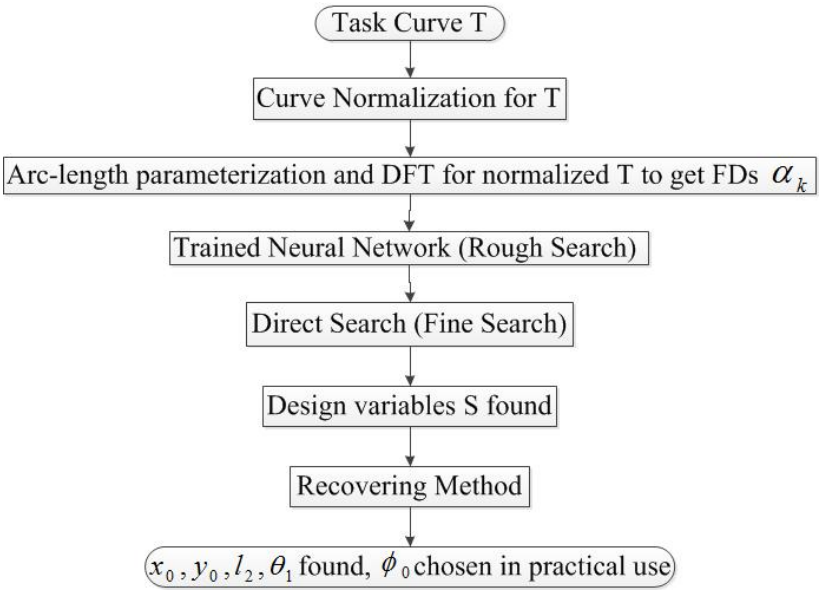

Figure 6.7: Synthesis Algorithm Flowchart (2)

6.4 Results and discussion

In this section, we present some examples to show the effectiveness of our approach and compare with Wu and Chu's methods respectively. First of all, we present a task curve that is generated by coupler point P of the Watt II six-bar shown in Fig. 6.8. Four-bar $AEFG$ functioning as driver linkage is serially chained with four-bar $ABCD$ of which P serves as the coupler point to trace out a coupler curve. $AEFG$ must be a double-crank mechanism since both GF and AE should be able to rotate in full circle respectively. When GF rotates with constant angular velocity, AE usually rotates with varying angular velocity. Hence, three different timings are produced by altering the lengths of GF, FE and AG to change the rotating pattern of AE. Link AB, BC, CD, BP and CP remain the same to keep the closed curve traced by P unchanged under these three timings.

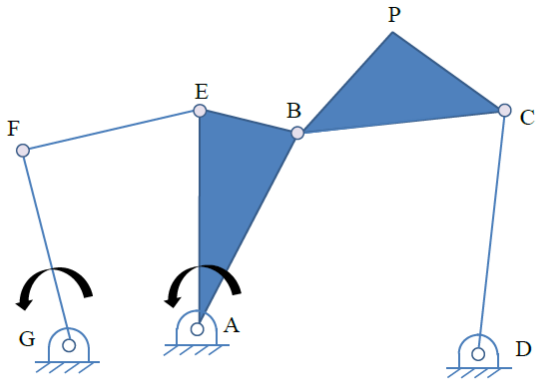

Figure 6.8: Watt II Six-bar

Nine design parameters are chosen as: $x_0 = 2.2, y_0 = 3.5, l_1 = 4.4, l_{21} = 0.5, l_{31} = 2.8, l_{41} = 2.7, x_c = 1.0, y_c = 0.8, \theta_1 = 0.55$. Three sets of AE, EF, FG and AG are given as follows:

$$Parametrization \quad I : AE = 0.50 \quad EF = 1.60 \tag{6.17}$$
$$FG = 1.60 \quad AG = 0.35$$
$$Parametrization \quad II : AE = 0.50 \quad EF = 0.80 \tag{6.18}$$
$$FG = 0.75 \quad AG = 0.40$$
$$Parametrization \quad III : AE = 0.50 \quad EF = 2.35 \tag{6.19}$$
$$FG = 2.40 \quad AG = 0.10$$

Rotation functions of link AB corresponding to each parametrization are presented in Fig. 6.9. By observing the figure, we can see that parametrization III is close to perfect timing in which link AB rotates uniformly while parametrization I and parametrization II deviate from pefect timing, which indicates that link AB rotates with varying angular velocity.

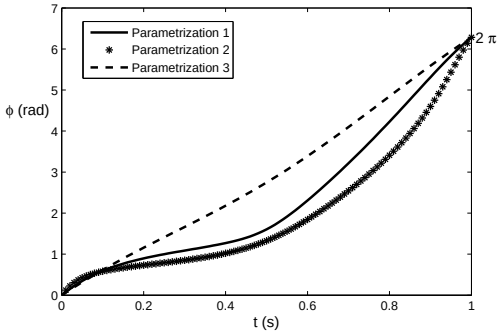

Figure 6.9: Rotation functions of link AB for three parametrizations

For these different parametrizations, their corresponding sets of Fourier Descriptors are listed in Table 6.1. It is clear from the table that different parametrizations will render distinct sets of Fourier Descriptors even though all correspond to the same curve.

	I	II	III
C_{-3}	0.9743	1.0661	0.6969
C_{-2}	1.7290	1.5949	1.4765
C_{-1}	2.3956	2.0666	3.2828
C_0	16.7382	17.2055	17.0292
C_1	1.7154	1.4730	2.3377
C_2	1.1184	1.0410	0.7418
C_3	0.4841	0.5770	0.2318

Table 6.1: Magnitudes of Fourier Descriptors of the task curve for three different parametrizations

For the coming two examples, we are to compare with traditional ways of synthesis approach. As mentioned in the introduction, they would generate different Fourier Descriptors given different parametrizations. In order to

compare their methods with ours based on the same measurement, arc-length parametrization is employed here to unify the results.

Parametrization	I	II	III
l_{21}	0.5045	0.5072	0.5077
l_{31}	2.8020	2.8039	2.7963
l_{41}	2.7070	2.7096	2.7080
x_c	1.0028	0.9942	1.0011
y_c	0.8075	0.7908	0.8036
l_1	4.3286	4.3549	4.3228
θ_1	0.5495	0.5615	0.5599
x_0	2.3715	2.4686	2.4719
y_0	3.5416	3.4464	3.5058

Table 6.2: The design parameters of synthesized four-bar linkages for three parametrizations by our method

	I	II	III
C_{-3}	0.4118	0.4118	0.4120
C_{-2}	0.1898	0.1918	0.1906
C_{-1}	3.7119	3.7090	3.7159
C_0	17.8644	17.8649	17.8641
C_1	1.9250	1.9216	1.9246
C_2	0.2560	0.2585	0.2573
C_3	0.1550	0.1539	0.1547

Table 6.3: Magnitudes of Fourier Descriptors of three synthesized coupler curves for three parametrizations after process of arc-length parametrization by our method

Example 1 First presented are the results of our method. In Table 6.2, design parameters corresponding to three parametrizations are displayed. We

can see that those parameters are close to those of the four-bar mechanism presented at the beginning that generates the task curve. In Table 6.3, Fourier Descriptors for three parametrizations are shown.

Parametrization	I	II	III
l_{21}	0.7980	0.8554	0.6052
l_{31}	1.7515	4.9052	1.8951
l_{41}	1.9051	4.9980	1.8050
x_c	0.8903	0.2190	1.3764
y_c	0.6551	0.4433	0.6998
l_1	4.2372	3.2539	4.3040
θ_1	1.0531	2.7821	0.7563
x_0	10.4813	10.8015	5.4999
y_0	0.7741	3.4920	2.1181

Table 6.4: The design parameters of synthesized four-bar linkages under three parametrizations by Wu's method

	I	II	III
C_{-3}	0.3588	0.3148	0.3954
C_{-2}	0.2429	0.2303	0.2058
C_{-1}	3.8635	3.8858	3.7170
C_0	18.1680	18.6918	17.8926
C_1	1.5478	1.4226	1.8367
C_2	0.2866	0.2687	0.2839
C_3	0.1569	0.1279	0.1532

Table 6.5: Magnitudes of Fourier Descriptors of three synthesized coupler curves by Wu's method

Second, we use Wu's synthesis approach to obtain three four-bar mechanisms and show results in Tables 6.4 and 6.5. According to Table 6.4, design

parameters are quite different from the task four-bar mechanism. Also from Table 6.5 and Table 6.3 , difference in Fourier Descriptors is notable. The graphical comparisons of our method and Wu's are demonstrated in Figs. 6.10, 6.11 and 6.12. Among three parametrizations, parametrization III is close to perfect timing as pointed out earlier and therefore Wu's method can yield good match with original curve. The other two parametrizations deviate from perfect timing to the extent that reduces the exactness of Wu's method.

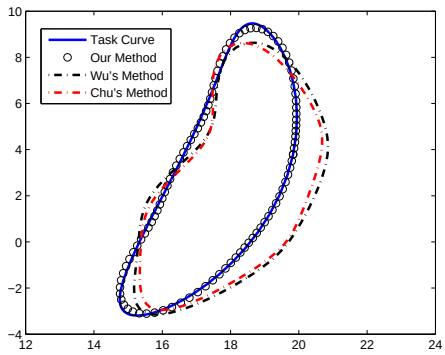

Figure 6.10: Comparison of Curves under parametrization I

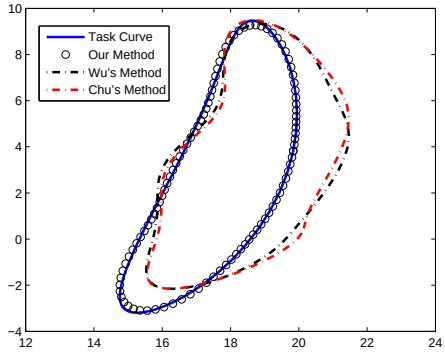

Figure 6.11: Comparison of Curves under parametrization II

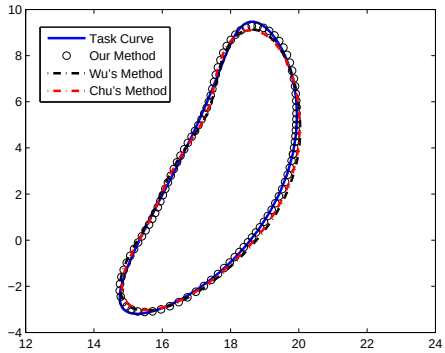

Figure 6.12: Comparison of Curves under parametrization III

Example 2 In this example, we would compare our method with Chu's, another widely used Fourier-based synthesis approach. The task curve and three parametrizations are same as those specified previously.

Then, we use Chu's approach to obtain three four-bar mechanisms and show results in Tables 6.6 and 6.7. The graphical comparisons of our method and Chu's are revealed in Figs. 6.10, 6.11 and 6.12. By comparing Chu's results with those from Wu's in Example 1, we find that both approaches

can output good match when the parametrization is nearly perfect, as in the case of parametrization III. Besides, the more the parametrization is deviated from perfect one, the more their synthesized curves differ from the task curve, which is justified by comparing synthesized curves of both methods under parametrization I with those of parametrization II in Figs. 6.10 and 6.11. Clearly, the results for parametrization I are better than those of II because parametrization I is closer to perfect timing than II as shown in Fig. 6.9.

Parametrization	I	II	III
l_{21}	0.8051	0.9011	0.6050
l_{31}	1.8950	4.9550	1.7516
l_{41}	2.0522	5.0032	1.8452
x_c	0.7978	0.2112	1.2720
y_c	0.6179	0.4468	0.5649
l_1	4.1185	3.0472	4.8738
θ_1	3.3154	0.3687	1.0926
x_0	10.8081	11.3207	6.3888
y_0	0.6750	3.5521	1.1477

Table 6.6: The design parameters of synthesized four-bar linkages under three parametrizations by Chu's method

	I	II	III
C_{-3}	0.3704	0.3994	0.3939
C_{-2}	0.2667	0.2338	0.2020
C_{-1}	3.7364	3.8833	3.6882
C_0	18.1640	18.7623	17.8883
C_1	1.5323	1.4473	1.8343
C_2	0.3102	0.3250	0.2736
C_3	0.1415	0.0523	0.1512

Table 6.7: Magnitudes of Fourier Descriptors of three synthesized coupler curves by Chu's method

6.5 Conclusions

In this chapter, we apply the technique of arc-length parametrization to eliminate the side effect of parametrization (timing) in traditionally Fourier-based path synthesis algorithms and realize the pure shape match. By implementing curve normalization, we successfully reduce the search space of design parameters; with Neural Network, five design parameters can be determined quickly once the network is trained and rest of four parameters are solved by recovering method, specifically by comparing size, orientation and position with task curve. The results are compared with Wu's and Chu's methods to justify that our method is truly independent of parametrization embedded in the path.

Chapter 7

Conclusions

The goal of this dissertation is to put forward a Kinematic Mapping and Fourier Framework to deal with dimensional and type thesis problem. The Kinematic Mapping based method gives us a novel way to treat the problem of type synthesis and that of dimensional optimization simultaneously. Using Fourier approach, we focus primarily on dimensional synthesis and optimization given the type of mechanism, which is demonstrated as being capable of tackle both path and motion synthesis problem.

By discovering the connection between dyads and triads both in mechanical and image space, our task driven approach to planar four-bar mechanism synthesis is naturally extended to planar parallel manipulator synthesis. Thus, we are able to meet the requirement of exactly following any number of arbitrarily given poses in planar space. Spatial motion synthesis is also performed based by extending our methodology in planar case. By characterizing spatial serial chain as kinematic geometry constraint of sphere and plane, we try to bridge the gap between type and dimensional synthesis for the structurally

complicated spatial parallel manipulator. For both planar and parallel manipulator, homotopy continuation method is used to effectively speed up the entire design process.

By using Fourier descriptor method, we present an effective algorithm for planar mechanism synthesis in the context of motion approximation. It builds on the recent work that mostly targets at path synthesis and apply Fourier transform to motion synthesis. By observing the relationship between translation and rotational component of a planar motion, a well-formed algorithm is proposed to nicely decouple the design space, which leads to drastic improvement in optimization routine. Following motion synthesis problem, we revisit the old fashioned Fourier based path synthesis problem to address the often-ignored parametrization issue. Prior to comparing task and four-bar coupler curve, we reparametrize them both with arc-length as parameter. In order to decouple the design space, only shape of two curves would be matched by Fourier descriptors and other design variables can be computed easily with recovering method. For minimizing search cost, Artificial Neural Network (ANN) is employed to establish the relationship between Fourier descriptors and design variables determining curve shape.

It is hoped that the fruits of this research would lead to an innovation and a commercialization in mechanism design as well as other fields connected with kinematics.

Bibliography

[1] Reuleaux, F., 1875. *Theoretical Kinematics: Outline of a Theory of Machines.*

[2] Hunt, K., 1978. *Kinematic Geometry of Mechanisms.* Oxford University Press, New York.

[3] Phillips, J., 1984. *Freedom in Machinery.* Cambridge University Press.

[4] Bottema, O., and Roth, B., 1979. *Theoretical Kinematics.* North Holland, Amsterdam.

[5] Erdman, A. G., and Sandor, G. N., 1997. *Mechanism Design: Analysis and Synthesis*, 3rd ed. Prentice Hall, NJ.

[6] McCarthy, J. M., 1990. *Introduction to Theoretical Kinematics.* MIT, Cambridge, Mass.

[7] McCarthy, J. M., 2000. *Geometric Design of Linkages.* Springer-Verlag, New York.

[8] Erdman, A. G., 1993. *Modern Kinematics: Developments in the Last Forty Years.* J.Wiley and Sons.

[9] Perez, A., and McCarthy, J. M., 2004. "Dual quaternion synthesis of constrained robotic systems". *ASME Journal of Mechanical Design,* **126**(3), pp. 425–435.

[10] Lee, E., and Mavroidis, C., 2006. "An elimination procedure for solving the geometric design problem of spatial 3r manipulators". *ASME J. Mechanical Design,* **128**(1), pp. 142–145.

[11] Su, H.-J., and McCarthy, J. M., 2005. "The synthesis of an rps serial chain to reach a given set of task positions". *Mechanism and Machine Theory,* **40**(7), pp. 757–775.

[12] Raghavan, M., and Roth, B., 1995. "Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators". *Journal of Mechanical Design,* **117**(B), pp. 71–79.

[13] Wampler, C. W., M. A. P., and Sommese, A. J., 1990. "Numerical continuation methods for solving polynomial systems arising in kinematics". *Journal of Mechanical Design,* **112**(1), pp. 59–68.

[14] Krovi, V., A. G. K., and Kumar, V., 2001. "Kinematic synthesis of spatial r-r dyads for path following with applications to coupled serial chain mechanisms". *Journal of Mechanical Design,* **113**(3), pp. 359–366.

[15] Su, H.-J., M. J. M., and Watson, L. T., 2004. "Generalized linear product homotopy algorithms and the computation of reachable surfaces". *Journal of Computing and Information Science in Engineering,* **4**(3), pp. 226–234.

[16] Blaschke, W., 1911. "Euklidische kinematik und nichteuklidische geometrie". *Zeitschr. Math. Phys.,* **60**, pp. 61-91.

[17] Grunwald, J., 1911, "Ein abbildungsprinzip, welches die ebene geometrie und kinematik mit der raumlichen geometrie verknupft". *Sitzber. Ak. Wiss. Wien, bf 120*, pp. 677-741.

[18] Ravani, B., and Roth, B., 1983. "Motion Synthesis Using Kinematic Mappings". *ASME J. Mech., Transm., Autom. Des.,* **105**(3), pp. 460-467.

[19] Ravani, B., and Roth, B., 1984. "Mappings of Spatial Kinematics". *ASME J.Mech., Transm., Autom, Des.,* **106**(3), pp. 341-347.

[20] McCarthy, J. M., 1990. *Introducation to Theoretical Kinematics*, MIT, Cambridge, MA.

[21] Bodduluri, R. M. C., and McCarthy, J. M., 1992. "Finite Position Synthesis Using the Image Curve of a Spherical Four-bar Motion". *ASME J. Mech. Des.,* **114**(1), pp. 55-60.

[22] Bodduluri, R., 1990. "Design and Planned Movement of Multi-Degree of Freedom Spatial Mechanisms". Ph.D. thesis, University of California, Irvine.

[23] Larochelle, P., 1994. "Design of cooperating robots and spatial mechanisms". Phd dissertation, University of California.

[24] Larochelle, P., 1996. "Synthesis of planar RR dyads by constraint manifold projection". In Proceedings of the ASME Design Engineering Technical Conferences.

[25] Ge, Q.J., and Larchelle, P., 1999. "Algebraic Motion Approximation With NURBS Motions and Its Application to Spherical Mechanism Synthesis". *J. Mech. Des.,* **121**(4), pp. 529-532.

[26] Husty, M.L., Pfurner, M., Schrocker, H.-P., and Brunnthaler, K., 2007, "Algebraic methods in mechanism analysis and synthesis". *Robotica*, 25:661-675.

[27] Wu, J. and Purwar, A. and Ge, Q. J., 2010. "Interactive Dimensional Synthesis and Motion Design of Planar 6R Single-Loop Closed Chains via a Constraint Manifold Modification". *ASME Journal of Mechanisms and Robotics,* **2**, pp. 31012(8 pages)

[28] Hayes, M., 1999. "Kinematics of general planar stewart-gough platforms". Ph.d. dissertation.

[29] Hayes, M. J. D., Zsombor-Murray, P. J., and Chen, C., 2004. "Kinematic analysis of general planar parallel manipulators". *ASME J. of Mechanical Design,* **126**(5).

[30] 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. Article Part 4.

[31] Brunnthaler, K., Schrocker, H., and Husty, M., 2006. "Synthesis of spherical four-bar mechanisms using spherical kinematic mapping". *Advances in Robot Kinematics*. Springer, Netherlands.

[32] Venkataramanujam, V., and Larochelle, P., 2007. "Approximate motion synthesis of spherical kinematic chains". In Proceedings of ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering, Paper No. DETC2007-34372.

[33] Dou, R., 2010, "Optimum design of 3-RRR planar parallel manipulators". Proceedings of the Institution of Mechanical Engineers, Part C: *J*ournal of Mechanical Engineering Science, Vol. 224, No. 2, pp. 411-418.

[34] Purwar, A., Anantwar, S., Zhao, P., 2012. "An Interactive Approach to Designing Planar Parallel Manipulators using Image Space Representation". ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, Paper No. DETC2012-70880.

[35] Innocenti, CC. 1995, "Polynomial Solution of the Spatial Burmester Problem", *J. Mech. Des.*, 117(1):64-68. doi:10.1115/1.2826118.

[36] Erdman, A. G., and Gustafson, J. E., 1981, "Lincages: Linkage Interactive Computer Analysis and Graphically Enhanced Synthesis Packages," ASMEDesign Engineering Technical Conferences, Paper No. 77-DET-5.

[37] Erdman, A. G., and Riley, D., 1981, "Computer-Aided Linkage Design Using the Lincages Package," ASME Design Engineering Technical Conferences, ASME, New York, Paper No. 81-DET-121.

[38] Larochelle, P. M., Dooley, A. P., Murray, A. P., and McCarthy, J. M., 1993, "Sphinx: Software for Synthesizing Spherical 4R Mechanisms," NSF Design and Manufacturing Systems Conference, pp. 607C611.

[39] Larochelle, P. M., 1998, "Spades: Software for Synthesizing Spatial 4C Linkages," ASME Design Engineering Technical Conferences (DETC).

[40] Su, H., Collins, C., and McCarthy, J. M., 2002, An Extensible Java Applet for Spatial Linkage Synthesis, 2002 ASME Design Engineering Technical Conferences, ASME, New York.

[41] Norton, R. L., 2012. *Design of Machinery: An Introduction to the Synthesis and Analysis of Mechanisms and Machines, Fifth Edition*, Mcgraw-Hill, New York.

[42] Freudenstein, F., 1960. "Harmonic analysis of crank-and-rocker mechanisms with application". *J. Appl. Mech.,* **26**, pp. 673-675.

[43] Funabashi, H., and Freudenstein, F., 1979. "Performance criteria for high-speed crank-and-rocker linkages-Part I: plane crank-and-rocker linkages". *ASME J. Mech. Des,* **101**(1), pp. 20-25.

[44] Fanhang, K., Midha, A., and Bajaj, A., 1988. "Synthesis of harmonic motion generating linkages-Part I: function generation". *Journal of*

Mechanisms Transmissions and Automation in Design-Transactions of The ASME, **110**(1), MAR, pp. 16-21.

[45] Fanhang, K., Midha, A., and Bajaj, A., 1988. "Synthesis of harmonic motion generating linkages-Part II: path and motion generation". *Journal of Mechanisms Transmissions and Automation in Design-Transactions of The ASME,* **110**(1), MAR, pp. 22-27.

[46] Chu, J., and Cao, W., 1993. "Synthesis of coupler curves of planar four-bar linkages through fast fourier transform". *Chinese Journal of Mechanical Engineering,* **29**(5), pp. 117-122.

[47] McGarva, J., and Mullineux, G., 1993. "Harmonic representation of closed curves". *Applied Mathematical Modelling,***17**(4), pp. 213-218.

[48] McGarva, J., 1994. "Rapid search and selection of path generating mechanisms from a library". *Mechanism and Machine Theory,* **29**(2), FEB, pp. 223-235.

[49] Ullah, L., and Kota, S., 1997. "Optimal synthesis of mechanisms for path generation using fourier descriptor and global search methods". *Journal of Mechanical Design,* **119**(4), pp. 504C510.

[50] Nie, X., and Krovi, V., 2005. "Fourier Methods for Kinematic Synthesis of Coupled Serial Chain Mechanisms". *ASME J. Mech. Des.,* **127**(1), pp. 232-241.

[51] Wu, J., Ge, Q.J., Gao, F., 2009. "An Efficient Method for Synthesizing Crank-Rocker Mechanisms for Generating Low-Harmonic Curves", Proceedings of 2009 ASME Mechanisms and Robotics Conference, San Diego, CA, Paper No. DETC2009-87140.

[52] Chu, J., and Sun, J., 2010. "A New Approach to Dimensional Synthesis of Spatial Four-Bar Linkage Through Numerical Atlas Method". *ASME J. Mechanisms and Robotics, Vol. 2*,pp. 041004-14.

[53] Chu, J., and Sun, J., 2010. "Numerical Atlas Method for Path Generation of Spherical Four-Bar Mechanism". *Mechanisms and Machine Theory,* **45**, pp. 867-879.

[54] Sun, J., and Chu, J., 2010. "Fourier Series Representation of the Coupler Curves of Spatial Linkages". *Applied Mathematical Modeling,* **34**, pp. 1396-1403.

[55] Chu, J. K., and Wang, L. D., 2004. "Relationship Between Properties of Coupler Curve and Link Dimensions in 4-Bar Mechanisms". *Sci. China, Ser. E: Technol. Sc.* **347**, pp. 753-762.

[56] Xie, J., Chen, Y., 2007. "Application Backpropagation Neural Network to Synthesis of Whole Cycle Motion Generation Mechanism". 12th IFToMM World Congress, Besancon, France.

[57] Vasiliu, A., Yannou, B., 2001. "Dimensional Synthesis of Planar Mechanisms using neural networks: application to path generator Linakges". *Mechanism and Machine Theory,* **36**, pp. 299-310.

[58] Merlet, J. P., 2006. *Parallel Robots*. Springer.

[59] Gough, V., E., and Whitehall, S., G., "Universal Tyre Test Machine", Proceedings of the FISITA Ninth Int. Technical Congress, London, UK, pp. 117-137, 1962.

[60] Stewart, D., "A Platform with Six Degrees of Freedom", Proc of the IMechE, Vol. 180, Pt. 1, No. 15, pp. 371 - 385, 1965-66.

[61] Ge, Q. J., Zhao, P., Purwar, A., and Li, X. 2012. "A novel approach to algebraic fitting of pencil of quadrics for planar 4R motion synthesis". *ASME Journal of Computing and Information Science in Engineering, 12*(4), p. 7.

[62] Ge, Q. J., Zhao, P., Purwar, A., 2013. "A Task Driven Approach to Unified Synthesis of Planar Four-bar Linkages using Algebraic Fitting of a Pencil of G-manifolds". ASME International Design and Engineering Technical Conferences, Portland, Paper No. DETC2013-12977

[63] Wu, J., Ge, Q. J., Su, H.-J., and Gao, F., 2010. "Kinematic acquisition of geometric constraints for task centered mechanism design". In Proceedings of the 2010 ASME Mechanisms and Robotics Conference, ASME, pp. DETC2010–28287.

[64] Purwar, A., and Gupta, A. "Visual synthesis of RRR- and RPR-legged planar parallel manipulators using constraint manifold geometry". In ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering, ASME. Paper No. DETC2011-48830.

[65] Sommerville, D., 1959. *Analytical geometry of three dimensions.* Cambridge University Press.

[66] 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.

[67] Verschelde, J., Cools, R., 1996, "Polynomial Homotopy Continuation, A Portable Ada Software Package." The Ada-Belgium Newsletter, 4:59-83, 1996. Proceedings of the Ada-Belgium Seminar, 22 November 1996, Eurocontrol, Brussels, Belgium.

[68] Verschelde, J. "Algorithm 795: PHCpack: A generalpurpose solver for polynomial systems by homotopy continuation," ACM Transactions on Mathematical Software 25(2): 251-276, 1999.

[69] Su, H.-J., McCarthy, J. M., Sosonkina, M., and Watson, L. T., 2006. "Algorithm 857: Polsysglp-a parallel general linear product homotopy code for solving polynomial systems of equations". ACM Trans. Math. Softw., 32(4), pp. 561-579.

[70] Bates, D. J., Hauenstein, J. D., Sommese, A. J., and Wampler, C. W. Bertini: Software for numerical algebraic geometry. Available at http://www.nd.edu/∼sommese/bertini.

[71] Lee, T. L., Li, T. Y., and Tsai, C. H., 2008. "Hom4ps-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method". Computing, 83(2-3), pp. 109-133.

[72] Chen, P., and Roth, B., 1967, "Design Equations for the Finitely and Infinitesimally Separated Position Synthesis of Binary Links and Combined Link Chains", *ASME J. Eng. Ind.*, 91, pp. 209-219.

[73] Suh, C.H., Radcliffe, C.W., 1978, Kinematics and Mechanisms Design, John Wiley.

[74] Tsai, L. W., and Morgan, A. P., "Solving the kinematics of the most general six- and five-degree of freedom manipulators by continuation methods", *ASME Journal of Mechanisms, Transmissions and Automations in Design*, 1985.

[75] Liao Q, and McCarthy J.M., 1997, "On the Seven Position Synthesis of a 5-SS Platform Linkage", *J. Mech. Des.*, 123(1):74-79. doi:10.1115/1.1330269.

[76] Kong, X., and Gosselin, C.M., 2004, "Type synthesis of 3-DOF translational parallel manipulators based on screw theory",*ASME J of Mechanical Design*, 126(1):83-92.

[77] Kong, X., and Gosselin, C.M., 2008, "Type synthesis of 4-DOF SP-equivalent parallel manipulators: A virtual chain approach", *Mechanisms and Machine Theory*, 41(11):1306-1319.

[78] Lee, E., Mavroidis, C., 2004, "Geometric design of spatial PRR manipulators", *Mechanism and Machine Theory*, 39(4):395-408.

[79] Li, Q., Huang, Z., Herve, JM, 2004, "Type synthesis of 3R2T 5-DOF parallel mechanisms using the Lie Group of displacements", *IEEE Robotics and Automation*, 20(2):173-180.

[80] McCarthy, JM., and Soh, G., 2011, Geometric Design of Linkages, Springer, New York.

[81] Ge, Q.J., Zhao, P., and Purwar, A., 2013, "Decomposition of planar Burmester problems using Kinematic Mapping", Advances in Mechanisms, Robotics and Design Education and Research Mechanisms and Machine Science. Volume 14, 2013, pp. 145-157.

[82] Husty, M. L., 1996, "An algorithm for solving the direct kinematics of general Stewart-Gough platforms", *Mech. Mach. Theory*, 31(4), pp. 365–380.

[83] Golub, G. and Van Loan, C., 1996. "Matrix Computations", Johns Hopkins Univ Press, Baltimore, MD

[84] Levin, J., 1976. "A parametric algorithm for drawing pictures of solid objects composed of quadric surfaces". *Communications of the ACM*, 19(10), pp. 555–563.

[85] Levin, J., 1979. "Mathematical models for determining the intersections of quadric surfaces". *Computer Graphics and Image Processing*,11(1), pp. 73–87.

[86] Dupont, L., Lazard, D., Lazard S. and Petitjean, S., 2008."Near-optimal parameterization of the intersection of quadrics: I. The generic algorithm, II. A classification of pencils and III. Parameterizing singular intersections", *Journal of Symbolic Computation*, 43(3), pp. 168–232.

[87] Wu, J., Ge, Q. J., Gao, F., and Guo, W. Z., 2011. "On the Extension of a Fourier Descriptor Based Method for Four-Bar Linkage Synthesis for the Generation of Open Curves". *ASME Journal of Mechanisms and Robotics,* **3**(3), pp. 031002.

[88] Dikabar, S., Mruthyunjaya, T. S., 1999. "Synthesis of workspaces of planar manipulators with arbitrary topology using shape representation and simulated annealing", *Mechanisms and Machine Theory, 34*, pp. 391-420.

[89] Bagchi, S., Mitra, S., 1999. *The Nonuniform Discrete Fourier Transform and its Applications in Signal Processing*, Kluwer, Boston, MA.

[90] Sánchez Marín, F. T., Pérez González, A., 2003. "Global optimization in path synthesis based on design space reduction", *Mechanisms and Machine Theory, 38*, pp. 579-594.

[91] Galán-Marín, G., Alonso, F. J., Del Castillo, J. M., 2009. "Shape optimization for path synthesis of crank-rocker mechanisms using a wavelet-based neural network", *Mechanisms and Machine Theory, 44*, pp. 1132-1143.

[92] Veelenturf, L. P. J., 1995. *Analysis and Applications of Artifical Neural Networks*, Prentice Hall International Ltd, Hemel Hempstead, Hertfordshire, UK.

[93] Anastasio, T. J., 2010. *Tutorial on Neural Systems Modeling*, Sinauer Associates Inc., Sunderland, MA, USA.

[94] Li, X., Zhao, P., Ge, Q. J., 2012. "A Fourier Descriptor Based Approach to Design Space Decomposition for Planar Motion Approximation", Proceedings of ASME 2012 International Design Engineering Technical Conferences Computers and Information in Engineering Conference, Chicago, IL, Paper No. DETC2012-71264.

Appendix: Normalization of A Closed Planar Curve

The curve is assumed to be a simple closed polygon of n sides where (x_i, y_i) represents the ith vertex of the polygon and (x_0, y_0) is considered identical to (x_n, y_n). All the calculations are done with respect to the length elements of the polygon.

Step 1: evaluate the length L of the curve. For that, let l_i denote the ith edge of the polygon. Then,

$$l_i = \sqrt{(x_{i-1} - x_i)^2 + (y_{i-1} - y_i)^2} \tag{7.1}$$

$$L = \sum_{i=1}^{n} l_i \tag{7.2}$$

Step 2: determine the location of the center of gravity (c.g.) of the curve given by (c_x, c_y).

$$c_x = \frac{1}{2L}\left(\sum_{i=1}^{n}(x_{i-1} + x_i)\sqrt{(x_{i-1} - x_i)^2 + (y_{i-1} - y_i)^2}\right) \tag{7.3}$$

$$c_y = \frac{1}{2L}\left(\sum_{i=1}^{n}(y_{i-1} + y_i)\sqrt{(x_{i-1} - x_i)^2 + (y_{i-1} - y_i)^2}\right) \tag{7.4}$$

Step 3: evaluate the moments of inertia I_{xx}, I_{yy} and I_{xy} of the polygon with respect to its c.g. (c_x, c_y).

$$I_{xx} = \frac{1}{3}\sum_{i=1}^{n}\{l_i[(y_{i-1}-c_y)^2 + (y_i - c_y)^2 + (y_{i-1}-c_y)(y_i-c_y)]\} \tag{7.5}$$

$$I_{yy} = \frac{1}{3}\sum_{i=1}^{n}\{l_i[(x_{i-1}-c_x)^2 + (x_i - c_x)^2 + (x_{i-1}-c_x)(x_i-c_x)]\} \tag{7.6}$$

$$I_{xy} = \frac{1}{6}\sum_{i=1}^{n}\{l_i[(x_{i-1}-c_x)(y_i-c_y) + (x_i-c_x)(y_{i-1}-c_y)] + 2l_i[(x_{i-1}-c_x)(y_{i-1}-c_y) + (x_i-c_x)(y_i-c_y)]\} \tag{7.7}$$

Step 4: determine the direction, α, of the major principal axis with respect to the x-axis

$$\alpha = \frac{1}{2}\arctan\left(\frac{2I_{xy}}{I_{yy}-I_{xx}}\right) \tag{7.8}$$

The direction of the major principal axis with respect to the x-axis is given by α if $I_{xx} < I_{yy}$ or $\alpha + \pi/2$ if $I_{xx} > I_{yy}$.

Step 5: rotate the polygon by an angle-α by the rotation to the vertices of the polygon in order to align the major principal axis with the x-axis.

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix}_{new} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix}_{old} \tag{7.9}$$

Step 6: evaluate the width w of the bounding box of the resulting polygon.

Let (P_x, P_y) and (Q_x, Q_y) represent the corners of the bounding box obtained by comparing the coordinates of the points of the polygon. Then the width w and height h of the bounding box are $|Q_x - P_x|$ and $|Q_y - P_y|$, respectively. Since the major principal axis is aligned with the x-axis after rotation, $w > h$.

Step 7: bring the polygon to its normalized configuration.

$$\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}_{normalized} = \begin{bmatrix} \frac{\cos\alpha}{w} & \frac{\sin\alpha}{w} & -\frac{P_x}{w} \\ -\frac{\sin\alpha}{w} & \frac{\cos\alpha}{w} & -\frac{P_y}{w} \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}_{old} \tag{7.10}$$