Stony Brook University

Variational Kinematic Geometry for Task Centered Design of Mechanisms and Robotic Systems

A Dissertation Presented
by

Jun Wu

to

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

Doctor of Philosophy
in
Mechanical Engineering

Stony Brook University
December 2010

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

Dr. Qiaode Jeffrey Ge, Advisor,
Professor, Department of Mechanical Engineering

Dr. Anurag Purwar, Co-Advisor,
Research Assistant Professor, Department of Mechanical Engineering

Dr. Jahangir S. Rastegar, Chair of Thesis Committee,
Associate Professor, Department of Mechanical Engineering

Dr. Yu Zhou, Member,
Assistant Professor, Department of Mechanical Engineering

Dr. Venkat N. Krovi, Outside Member,
Associate Professor, Department of Mechanical and Aerospace Engineering,
State University of New York at Buffalo

This dissertation is accepted by the Graduate School.

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation
Variational Kinematic Geometry for Task Centered Design of Mechanisms and Robotic Systems

by
Jun Wu
Doctor of Philosophy
in
Mechanical Engineering
Stony Brook University
2010

This dissertation deals with the problem of task centered design of mechanisms and robotic systems via novel concept of variational kinematic geometry. The results can be extended to other field such as biomechanics, structure chemistry and protein kinematics, as well as micro- and nano- systems, in which kinematics plays an important roles.

Since the early days of Industrial Revolution, machine theorists and kinematicians have sought to develop a theory to analyze and synthesize mechanisms so that engineers could approach the problem in a rational way. This dissertation introduces a task-centered approach to mechanism design using a constraint based paradigm, and conducts a comparative study on the kinematic and the geometric constraints of the motion.

Kinematic-constraint based synthesis approach, derived from the classical viewpoint that a kinematic mechanism is a collection of kinematic links connected with kinematic pairs (or joints), deals with the determination of mechanism types (type

synthesis) and/or their link dimensions (dimensional synthesis). Hence, this approach is referred to as two-step-based mechanism-centric design paradigm.

This dissertation advocates a geometric-constraint based approach. Following this approach, a designer would focus on the analysis of point (or line) trajectories associated with the motion, the goal of which is to obtain a trajectory that can be constructed as a geometric condition or constraint that best describes the motion. Typically this is done in a geometric constraint identification and acquisition process, i.e., by comparing various trajectories of a specified motion with known constraints from a library of mechanically realizable constraints. The resulting feasible constraints can be used directly for the simultaneous type and dimensional synthesis of a physical device such as mechanical linkage that generates the specified motion task. This effectively reduces the problem of mechanism synthesis to that of constraint identification and acquisition and thus bridges the gap between type and dimensional synthesis. Furthermore, as a mechanism is defined by a combination of geometric constraints, the geometric-constraint based approach reduces the complexity in type synthesis significantly. This approach to motion modeling has similarity to constraint based shape modeling in Variational Geometry and is therefore referred to as Variational Kinematic Geometry.

To my family, especially...
to Muxiang Zhang, Haiquan Wu, Xia Huang,
Wenjing Jiang and Xiaolin Luo

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, and Professor Anurag Purway, my research co-advisor, for their guidance, encouragement and patience throughout the work of this dissertation.

I would like to thank my committee members, Professor Jahangir S. Rastegar, Professor Yu Zhou, and Professor Venkat N. Krovi for their review of this manuscript.

My thanks also go to my lab mates Zhe Jin, Carlos A. Trujillo, Ping Zhao, Xiangyun Li for their suggestions and friendship; and my department staff Diane Van Tronk, Augusta Kuhn and Melissa Castelbuono for their valuable assistance.

Finally, I gratefully acknowledge the financial support of this work by the National Science Foundation under grant number DMI-0500064 and CMMI-0856594 as well as by National Natural Science Foundation of China (Oversea Scholar Research Collaboration Grant No.50728503).

Chapter 1

Introduction

Since the early days of Industrial Revolution, machine theorists and kinematicians have sought to develop a theory to analyze and synthesize mechanisms so that engineers could approach the problem in a rational way. Franz Reuleaux, the author of "Theoretische Kinematik" [1], laid the foundation for modern kinematics by determining the basic mechanical building blocks and developing a system for classifying mechanism types (There is an excellent digital library at Cornell University that documents the work done by Reuleaux called Kinematic Models for Design Digital Library). Another giant in mechanism classification is Ivan Artobolevsky who published in 1975 an encyclopedia of known mechanisms titled "Mechanisms in Modern Engineering Design" [2]. More recent comprehensive treatment of kinematic geometry in English language include Hunt [3], Phillips [4], Bottema and Roth [5], Erdman and Sandor [6, 7], McCarthy [8, 9]. Erdman and Sadler [10] summarizes the development of modern kinematics in the forty years preceding 1992.

While there exists comprehensive treatment of kinematics as a fundamental science in its own right such as the seminal work titled "Theoretical Kinematics" by Bottema and Roth [5], much of the existing work in kinematic geometry deals with

the application of kinematics to the analysis and synthesis of mechanisms. Kinematic synthesis deals with 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. As type synthesis constitutes the genesis of innovation and creativity in the design of mechanism-based products and systems, it is the most critical part of the mechanism design process.

The dimensional synthesis of linkages is an extensively researched subject with many textbooks. The synthesis equations can be systematically derived with various mathematical formulation such as vector loop closure equations [6], quaternion/dual quaternion [11], homogeneous matrices [12]. Recently Su and McCarthy [13, 14] 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 mechanism design is a well studied topic [15, 16] and the solution techniques include iterative optimization [17], exact analytical methods [18, 19] and continuation (homotopy) method [20, 13].

Compared to dimensional synthesis, type synthesis is a much less researched

subject. Nonetheless, there exists a sizable amount of literature, especially for the case of planar mechanisms. The main approaches include augmentation of polygonal-link patterns through introduction of binary links; transformation of binary chains; building up of desired chains by addition of links to simpler chains with fewer links; and direct algebraic determination of matrices representing chains. The tools used to implement these approaches can be based on direct visual inspection and intuition, graph theory, group theory or matrix representation based algebraic procedures. The focus of all these work, however, is not on the task to be accomplished but rather on the classification and enumeration of mechanisms. A comprehensive review of research in this area can be found in Mruthyunjaya [21]. In spite of all the advances in this area, type synthesis remains to be the most elusive part of the mechanism design process.

Deriving from the two-step based design approach, there have been a lot of academic research efforts in the development of software systems for the synthesis of planar, spherical, and spatial mechanisms (see KINSYN III from Rubel and Kaufamn [22], LINCAGES from Erdman and colleagues [23, 24], Kihonge et al. [25], Spades from Larochelle [26], Sphinx from Larochelle et al. [27], Sphinxpc from Ruth and McCarthy [28], Osiris from Tse and Larochelle [29], Perez and McCarthy [30], Su and McCarthy [31], Synthetica from Su et al. [32]). In the commercial domain, SyMech [33] and WATT [34] are two well-known software systems for planar mechanisms design.

The two-step based paradigm for mechanism design, which has been around for over a century, is derived from the classical viewpoint that a kinematic mechanism

is a collection of kinematic links connected with kinematic pairs (or joints). Links, joints and the pattern of their interconnections are viewed as a natural descriptor for the characteristics of a mechanism and provide an effective means for mechanism classification and enumeration. However, these basic mechanical elements are in general not natural descriptor for the characteristics of the output motion of a mechanism. Detailed kinematic analysis is in general required in order to understand the motion or functional characteristics of a mechanism. Since the important work by Freudenstein and Maki [35], there have been numerous attempt to separate structural and functional (or task) considerations in mechanism design. For example, Chiou and Kota [36] identified a finite set of kinematic building blocks that provide simple kinematic functions and devised a matrix representation scheme for automatic generation of conceptual designs. Yan [37] presented a comprehensive approach to creative design of mechanisms. In spite of all the advances in this area, including the application of artificial intelligence (Hoeltzel and Chieng [38], Yang et al. [39]), this "mechanism-centric" approach to mechanism design makes type synthesis a very challenging task, even for those who have been well trained with mechanism theory.

It is believed that only through combing these two seeming distinct tasks in the design process, one can find the best mechanism (optimal type and its ideal dimensions) for a specific task [40]. There have been attempts to solve the combined problem of type synthesis and dimensional synthesis through the use of Genetic Algorithm (GA) (Sedlaczek et al. [41], Frecker et al.[42]) as well as a uniform polynomial system by Hayes and Zsombor-Murray [43]. However, they have been

carried out for very restricted applications with very limited number of mechanism types. Furthermore, these approaches do not reduce the complexity of the type synthesis problem. Researchers in AI (Artificial Intelligence) community have also sought with very limited success to bridge the gap between type and dimensional syntheses and developed what is known as Qualitative Kinematics [44] in the context of qualitative spatial reasoning.

This dissertation deals with the task-centered design of mechanisms and robotic systems. Typical task-driven problem, also known as motion synthesis problems, seeks to synthesize the mechanisms that guides a rigid body to navigate through, or as close as possible to a prescribed motion. In the processes to design a mechanical system, task-driven design occurs in conceptual design, the critical stage leading to the architectural solutions of a machine. The ways to constrain the motion is the prime conditions for the design method. This document provides a comparative study on the planar task-driven design with two approaches to constrain the motion task: kinematic constraints and geometric constraints.

Mechanism design based on kinematic constraints is well-developed (see Sandor and Erdman [7], Suh and Radcliffe [45], and McCarthy [9]). This design approach, falling into the category of two-step based paradigm, describes the task motion as constrained by the mechanical joints. In this dissertation, we present an design example of planar 6R closed chain synthesis to illustrate this design approach; the problem that we solve here is as follows: given a prescribed planar rational motion, what are the dimensions of the links of a planar 6R closed chain that accomplishes the given task? We use the well-known kinematic mapping ap-

proach pioneered by Bottema and Roth [46] and McCarthy [8] to model the design problem. In kinematic mapping, a planar displacement is represented by planar quaternion. In the four-dimensional quaternion space, a planar displacement corresponds to a point, referred as the image point, and analogically the quaternion space is known as image space. Hence, the task motion transformed to a image curve, and specifically the task given as a sequence of discrete positions is interpolated to be a rational motion (see Jin and Ge [47, 48] and Purwar et. al. [49, 50, 51]). Meanwhile, the workspace of one mechanical joint is converted to the volume bounded by a manifold, called the constraint manifold. Thereafter, the design problem can be expressed as finding the proper dimensions of the 6R closed chain such that the constraint manifold contains the image curve of the task motion. In details, our design method treats the 6R closed chains as mechanisms assembled using two open chains connected together at the ends. Each open chains impose a kinematic constraint that limit the positions and orientations of the end effector. Algebraically, the workspace of each chain are in inequality form and geometrically, that corresponds to a quadratic manifold in the projective four dimensional space. The constraint manifold is visualized in the three dimensional space by projecting it onto a hyperplane, which is a pair of co-centric and co-oriented hyperboloid. Favorably, the position, the size and the orientation of the constraint hyperboloid correlate to the dimensions of the each open chain in a simple and decoupled way. This nature gives the birth to an interactive design scheme with which designer can manipulate with the constraint manifolds to indirectly yet intuitively modify the dimension of the links. In this dissertation, we studied the convexity of the image curve of the

rational motion, and developed an interactive software for the design of planar 6R closed chain.

In contrast to the prevailing mechanism-centric approach, this dissertation advocates a task-centered approach to mechanism design using a constraint based paradigm. Following this approach, a designer would focus on the *analysis* of point (or line) trajectories associated with a given motion task. The goal of the trajectory analysis is obtain a trajectory that can be construed as a geometric condition or constraint that best constrains the given motion. Typically this is done in a constraint identification and acquisition process, i.e., by comparing various trajectories of a specified motion with known constraints from a library of mechanically realizable constraints. The resulting constraints are feasible constraints that best approximate the given motion in some measure, say, in a least squares sense. A motion task can be given in various ways. It can be defined parametrically or discretely in terms of an ordered sequence of displacements or in geometric means. The resulting feasible constraints can be used directly for the simultaneous type and dimensional synthesis of a physical device such as mechanical linkage that generates the specified motion task. This effectively reduces the problem of mechanism synthesis to that of constraint identification and acquisition and thus bridges the gap between type and dimensional synthesis. Furthermore, as a mechanism is defined by a combination of geometric constraints, this constraint based approach reduces the complexity in type synthesis significantly. Only combinations of the feasible constraints are considered for type synthesis. This constraint based approach to motion modeling has similarity to constraint based shape modeling in Variational Geometry and is there-

fore referred to as *Variational Kinematic Geometry*. The focus of this dissertation is on the introducing the framework of this constraint based approach. We propose some simple solutions to the constraint identification and acquisition problem by directly comparing corresponding points associated with a point (or line) trajectory and a given constraint. More sophisticated and indirect methods are also developed for the solution to the problem. In the field of *Computational Shape Analysis* [52], it is routine to process and simplify shapes before comparisons are made. The simplified representation of shapes is called *shape descriptor* or *shape signature*. The use of shape descriptors often simplifies the process of not only shape comparison but also shape storage and classification. We explore these methods for better solutions to the problem of constraint identification.

The contribution of this dissertation is summarized as below: This dissertation focuses on the task-centric design problem, and advocates a new geometric constraint based design paradigm, namely "variational kinematic geometry". For the purpose to identify the merits of geometric constraint approach, we conduct a comparative research on both the classical mechanical-centric (see chapter 4 to 6) and new geometric-centric (see chapter 7 to 10) methods. For the mechanical-centric method, we study the synthesis methods for planar 6R closed chain and four-bar mechanism, both of which follow the same design pattern that the type of the mechanism is required to know first and the optimization is used to revise the dimension. In the other words, there is a clear boundary between type synthesis and dimensional synthesis if mechanical-centric approach is applied for mechanism design. We propose the variational kinematic geometry method to deal with

the task-centric design from the perspective of geometry. Our method is able to simultaneously synthesize a mechanism both topologically and dimensionally. As shown in chapter 7 to 10), our design approach emphasizes on the geometric feature of the task motion and follows such a procedure that geometric constraints are firstly identified by using shape analysis method, and then multiple design solutions are yielded by choosing diverse feasible mechanisms to generate the geometric constraints individually. Our design approach has similarity to ATLAS method in view of both of them are using the geometric information of the task. ATLAS method works in this way: the curves generated by different mechanisms are preprocessed and stored in a library, and to design a mechanism is to look up a curve to match the desired curve and choose the according mechanism. Our method is different from ATLAS method from three ways. Firstly, up to our literature review, existing ATLAS method works for the synthesis of gear 2R chain, gear 5R closed chain, cam and four-bar linkage. Obviously, ATLAS method requires to type of the mechanism to be known. However, our method has no boundary between type and dimension synthesis, and we can with more types than existing ATLAS method can do. Secondly, ATLAS method can be only applied to trajectory tasks, but the geometric method works for motion generation, and we use ATLAS as part of our geometric constraint library to build more complex mechanisms that are not available from ATLAS methods. Thirdly, we use the FD based shape comparison algorithms to retrieve the geometric features. This kind of constraint identification methods are compatible with human vision according to the study by cognitive scientists[107].

This dissertation deals with the task-driven design problem through a compar-

ative study on the mechanical- and geometrical-centric approach and is organized as following. Chapter 2 and 3 briefly review kinematic mapping using quaternion representation and Fourier descriptors, respectively. The mechanical-centric design approach is covered by chapter 4 to 6: chapter 4 develops the concept of kinematic convexity of the rational motion as a basic tool for geometric computations involving planar displacements; chapter 5 presents an interactive, visual design approach for the dimensional synthesis of planar 6R single-loop closed chains for a given rational motion using constraint manifold modification, an example of mechanical-centric design method; and chapter 6 introduces a Fourier descriptor based method for four-bar linkage synthesis for generation of the open and closed curves, which is viewed as the complicated point-trajectory constraints. The geometrical-centric design approach is explored by chapter 7 to 10: chapter 7 proposes the problem of kinematic acquisition of point geometric constraints and provides a least-square solution to the point-trajectory constraint retrieval; chapter 8 seeks to apply the concept of Fourier Descriptors (FD) to the problem of estimating or extracting geometric constraints from a given planar motion task; for the interest to find a better type of FD for constraint retrieval, chapter 9 conducts a comparative study on four types of FDs based on different curve signatures; and finally, chapter 10 extends the work of geometric acquisition of point constraints to line constraints with the observation of the duality between points and lines. We draw conclusion remarks in chapter 11.

Chapter 2

Kinematic Mapping

2.1 Planar Quaternions

Consider a planar displacement in X-Y plane shown in Figure 2.1. Let d_1, d_2 denote the coordinates of the origin of the moving frame **M** in the fixed frame **F** and ϕ denote the rotation angle of **M** relative to **F**. Then a planar displacement can be represented by a planar quaternion, $\mathbf{Z} = (Z_1, Z_2, Z_3, Z_4)$ (see McCarthy [8]):

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

They can be considered as a set of homogeneous coordinates that define the image space of planar displacements (Ravani and Roth [53], McCarthy [8]).

These four components can be identified as coordinates of a point in a four dimensional space. 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 displacements and is denoted as Σ_p. In view of Eq. (2.1), the

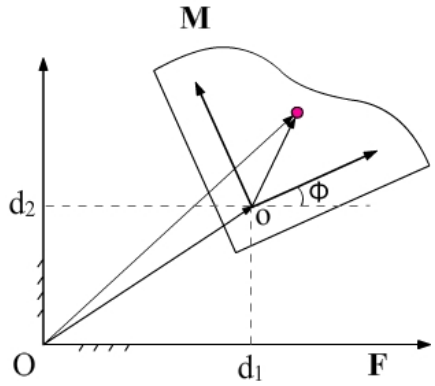


Figure 2.1: A planar displacement.

coordinates of an image point must satisfy the equation:

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

The above equation may be interpreted as defining a hyper-circular cylinder in a 4D space.

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

$$[A] = \frac{1}{Z_3^2 + Z_4^2} \begin{bmatrix} Z_4^2 - Z_3^2 & -2Z_3Z_4 & 2(Z_1Z_4 - Z_2Z_3) \\ 2Z_3Z_4 & Z_4^2 - Z_3^2 & 2(Z_1Z_3 + Z_2Z_4) \\ 0 & 0 & Z_3^2 + Z_4^2 \end{bmatrix}. \tag{2.3}$$

Note that when Z_i $(i = 1, 2, 3, 4)$ is replaced by wZ_i, where w is a nonzero scalar, the matrix $[A]$ is unchanged. From this perspective, the four components of a planar quaternion can also be considered as a set of homogeneous coordinates for a planar displacement. Ravani and Roth [53] considered the components of **Z** as defining a point in a projective three-space, called the Image Space of planar kinematics.

2.2 Dimensionless Planar Quaternions

The topic of distance metric has attracted significant interest in kinematics community in recent years. An excellent review of papers on the subject can be found in Larochelle et al. [54]). One interesting approach is to approximate a planar displacement ($SE(2)$) with a spherical displacement ($SO(3)$) in order to obtain a distance metric that is approximately bi-invariant [55, 56, 57]. In order to resolve the inconsistency in units between rotation and translation, it is commonly suggested that the translation vector be divided by a characteristic length R so that the translation vector will be dimensionless, just as rotation matrix is dimensionless as well. Following this approach, in this paper, we propose a dimensionless version of planar quaternion, $\mathbf{F} = (F_1, F_2, F_3, F_4)$, by replacing d_i with d_i/R in (2.1):

$$\begin{aligned} F_1 &= \frac{d_1}{2R}\cos(\phi/2) + \frac{d_2}{2R}\sin(\phi/2), \\ F_2 &= -\frac{d_1}{2R}\sin(\phi/2) + \frac{d_2}{2R}\cos(\phi/2), \\ F_3 &= \sin(\phi/2), \\ F_4 &= \cos(\phi/2). \end{aligned} \tag{2.4}$$

In this way, all four components of the planar quaternion are dimensionless and this makes it possible to define a distance metric in a consistent way. However, the dimensionless planar quaternions as defined above still do not allow a bi-invariant distance metric.

A planar quaternions is a special case of a dual quaternion associated with a spatial displacement ($SE(3)$) (McCarthy [8]). A spatial displacement consists of a rotation and a translation. The rotation can be represented by a unit quaternion

$\mathbf{Q} = Q_1\mathbf{i} + Q_2\mathbf{j} + Q_3\mathbf{k} + Q_4$, where $(\mathbf{i}, \mathbf{j}, \mathbf{k}, 1)$ denote the quaternion basis. The four components of $\mathbf{Q}$ are defined by the unit vector along the axis of rotation, $\mathbf{s} = (s_1, s_2, s_3)$, and the angle of rotation, ϕ, as

$$\mathbf{Q} = s_1 \sin(\frac{\phi}{2})\mathbf{i} + s_2 \sin(\frac{\phi}{2})\mathbf{j} + s_3 \sin(\frac{\phi}{2})\mathbf{k} + \cos(\frac{\phi}{2}). \tag{2.5}$$

Let $\mathbf{d} = d_1\mathbf{i} + d_2\mathbf{j} + d_3\mathbf{k}$ denote a vector quaternion representing the translation component and let R denote the characteristic length. In the dual quaternion representation of a spatial displacement, the translation part is represented by another quaternion $\mathbf{Q}^0$. The dimensionless version of the quaternion is given by the quaternion product:

$$\mathbf{Q}^0 = \frac{\mathbf{dQ}}{2R}. \tag{2.6}$$

For a planar displacement in the X-Y coordinate plane, we have $\mathbf{Q} = \sin(\phi/2)\mathbf{k} + \cos(\phi/2)$ and $\mathbf{d} = d_1\mathbf{i} + d_2\mathbf{j}$. Substituting them into Eq. (2.6), and after some algebra, we obtain

$$\mathbf{Q}^0 = F_1\mathbf{i} + F_2\mathbf{j} \tag{2.7}$$

where F_1, F_2 are given by (2.4). It is clear that four nonzero components of the dual quaternion, $(\mathbf{Q}, \mathbf{Q}^0)$ reduce to those of the planar quaternion as given by (2.4).

2.3 Connection Between Planar Quaternion and Double Quaternion

It is known that a general rotation in $SO(4)$ can be represented by a pair of unit quaternions **G** and **H** [58, 59] such that

$$\tilde{\mathbf{X}} = \mathbf{GXH}^* \tag{2.8}$$

where $\mathbf{X}$ and $\tilde{\mathbf{X}}$ represent coordinates of a point before and after the rotation. The quaternion pair is called a *double quaternion* in Study form as it keeps the magnitudes of the vectors under the rotational transformation invariant. The double quaternions can be recasted in the following form by introducing

$$\mathbf{B} = \frac{\mathbf{G} + \mathbf{H}}{2}, \quad \mathbf{B}^0 = \frac{\mathbf{G} - \mathbf{H}}{2}, \tag{2.9}$$

They satisfy the Plücker condition

$$B_1 B_1^0 + B_2 B_2^0 + B_3 B_3^0 + B_4 B_4^0 = 0.$$

In addition, they also keep invariant the Plücker condition for line coordinates before and after the rotation, and thefore, they are called *double quaternion in Plücker form* (see Ge [59] and Purwar and Ge [60]). It has further been shown in [60] that the dual quaternion $(\mathbf{Q}, \mathbf{Q}^0)$ for $SE(3)$ differs from the double quaternion $(\mathbf{B}, \mathbf{B}^0)$ only by scaling factor, i.e., by extending the magnitude of $(\mathbf{B}, \mathbf{B}^0)$ so that $\mathbf{B}$ becomes a unit quaternion $\mathbf{Q}$. The choice of the scaling factor determines how closely a dual quaternion can be approximated by a double quaternion.

Let $\hat{\mathbf{Q}} = \mathbf{Q} + \epsilon \mathbf{Q}^0$ denote a dual quaternion where ϵ is the dual unit with the property $\epsilon^2 = 0$. Then a planar quaternion in dual quaternion form is given by $\hat{\mathbf{Q}} = F_3\mathbf{k} + F_4 + \epsilon(F_1\mathbf{i} + F_2\mathbf{j})$. This means that

$$\mathbf{B} = w(0, 0, F_3, F_4), \quad \mathbf{B}^0 = w(F_1, F_2, 0, 0) \tag{2.10}$$

where w is the scaling factor. In view of (2.9), we have

$$\mathbf{G} = \mathbf{B} + \mathbf{B}^0, \quad \mathbf{H} = \mathbf{B} - \mathbf{B}^0. \tag{2.11}$$

After the substitution of (2.10) into (2.11), we obtain

$$\mathbf{G} = w(F_1, F_2, F_3, F_4), \;\; \mathbf{H} = w(-F_1, -F_2, F_3, F_4). \tag{2.12}$$

This means that we can approximate a planar displacement as defined by a planar quaternion **F** with a spherical displacement as defined by the unit quaternion **G**, which is obtained by normalizing the planar quaternion **F**, i.e., by letting

$$w = (F_1^2 + F_2^2 + F_3^2 + F_4^2)^{-1/2} = \frac{2R}{\sqrt{d_1^2 + d_2^2 + 4R^2}}. \tag{2.13}$$

Thus the larger the R, the better is the approximation. (More discussion of R can be found in [61, 62]). Alternatively, one can also use **H** in (2.12) instead of **G**. In this paper, we use **G** to represent a planar displacement.

2.4 Unconstrained Planar B-Spline Motion

This section reviews the idea of synthesis of unconstrained (also known as, free-form) planar rational motion. For the purpose of this paper, if the input motion is specified in terms of the given coupler positions, a rational motion that interpolates to the given positions can be constructed using the method outlined in this section.

Since a planar quaternion **Z** can be seen as a point in the image space, a polynomial curve in the image space corresponds to a rational motion in the Cartesian space. Rational motions are defined as ratio of two polynomial functions of time. By applying Computer Aided Geometric Design (CAGD) techniques for designing curves in the image space, we obtain rational motions in the Cartesian space. The rationale behind the choice of rational motions stems from the fact that the trajectories of an object undergoing rational motions are rational curves and surfaces,

thereby making them convenient for integration into existing Non-uniform Rational B-spline (NURBS) based CAD/CAM system.

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

$$\mathbf{Z}(u) = \sum_{i=0}^{n} N_{i,p(u)} \mathbf{Z}_i, \tag{2.14}$$

where $N_{i,p}(u)$ are pth-degree basis functions. See Farin [63], Hoschek and Lasser [64], and Piegl and Tiller [65] for details on the B-spline curves. Since the given positions $\mathbf{Z}_i$ (called control positions in the language of B-spline formulation) represent planar displacements, we get a B-Spline quaternion curve $\mathbf{Z}(u)$ in the space of planar quaternions.

A representation for the rational B-spline motion in the Cartesian space can be obtained by substituting the coordinates of $\mathbf{Z}(u)$ into the matrix $[A]$ (Eq 7.1), where each element of the matrix is now given as a rational function in parameter u.

Chapter 3

Fourier Descriptors

3.1 Complex Form of Fourier Transform

Consider a plane curve defined by $\mathbf{z}(t) = x(t) + iy(t)$ where x and y are the coordinates of a point on the curve and i is the imaginary unit. If the curve is closed, the complex function $\mathbf{z}(t)$ can be considered as a periodic function. Let T denote the minimum positive period of the function. The Fourier transform of a continuous periodic function $\mathbf{z}(t)$ in complex form is given by

$$\mathbf{z}(t) = \sum_{n=-\infty}^{\infty} \alpha_n e^{i2n\pi t/T}, \tag{3.1}$$

where the Fourier coefficients α_n are complex numbers given by

$$\alpha_n = \frac{1}{T} \int_0^T \mathbf{z}(t) e^{-i2n\pi t/T} dt. \tag{3.2}$$

An intuitive interpretation of this complex form of Fourier transform comes from the viewpoint of orthogonal projection in the *inner product space*[66]. Eq. (3.1) represents $\mathbf{z}(t)$ as a linear combination of the Fourier basis. According to the definition of the inner product, $e^{i2n\pi t/T}$ is orthognol to any term other than $e^{-i2n\pi t/T}$ in the Fourier basis $\{e^{i2k\pi t/T} | k = -\infty \cdots \infty\}$. Essentially, the Fourier transfrom (see

Eq.(3.2)) project the vector $\mathbf{z}(t)$ orthogonally onto basis $e^{-i2n\pi t/T}$ which is done by taking and normalizing their inner product. Thus, the Fourier coefficient α_n is reserved as the projection and all other Fourier coefficient vanish because of the orthogonality.

Among the coefficients, α_0 defines the amplitude of the *fundamental*, α_{-1} and α_1 defines the amplitude of the *first harmonic*, and α_{-2} and α_2 the amplitude of the *second harmonic*, and so on. The fundamental coefficient, α_0, defines the centroid, or average of all points on the curve. Thus it defines the location of the curve but not its shape. Now let us consider the contribution of the first harmonic terms to the shape of the curve:

$$\mathbf{z}_1(t) = \alpha_{-1}e^{-2\pi it/T} + \alpha_1 e^{2\pi it/T} \tag{3.3}$$

As each coefficient is a complex number, we have

$$\alpha_{-1} = r_{-1}e^{i\phi_{-1}}, \;\; \alpha_1 = r_1 e^{i\phi_1} \tag{3.4}$$

where r_{-1} and r_1 are the magnitudes of the complex coefficients. It can be shown that Eq.(3.3) defines an ellipse whose semi-major and semi-minor axes are given by $(r_{-1} + r_1)$ and $(r_1 - r_{-1})$, respectively. The angle that the semi-major axis makes with the horizontal is $(\phi_1 + \phi_1)/2$. When $r_1 > r_{-1}$, points on the ellipse are traced out in counter-clockwise direction as t varies from 0 to T. Similarly, when $r_1 < r_{-1}$, the ellipse is traced out in clockwise direction as t varies from 0 to T. When $r_1 = r_{-1}$, the ellipse collapses into a straight line as the semi-minor axis vanishes. When one of them is equal to zero, the ellipse becomes a circle.

The shape defined by the second harmonic can be studied similarly:

$$\mathbf{z}_2(t) = \alpha_{-2}e^{-4\pi it/T} + \alpha_2 e^{4\pi it/T}. \tag{3.5}$$

It also defines an ellipse in general. The only difference is that as t changes from 0 to T, the ellipse gets traced out twice. For a smooth curve, the amplitudes α_n decreases as n increases. Thus lower harmonics captures the gross shape of $\mathbf{z}(t)$ and higher harmonics add more details. In this paper, however, we are interested in the development of an efficient algorithm for synthesizing a crank-rocker mechanism such that its coupler curve traces out a smooth curve that can be approximated by the first and second harmonics.

3.2 Discrete Fourier Transform (DFT)

Compared with the Fourier transform, which is useful for continuous function, the DFT is a powerful technique to analyze discrete data. It is often used for numerical approximation of the Fourier coefficients of a continuous periodic function. The forward and inverse DFT of a finite-length sequence of N data points of a uniformly sampled, periodic function $\mathbf{z}(t)$ are represented as:

$$\mathbf{z}(2\pi\frac{t_i}{T}) = \sum_{n=0}^{N-1} \alpha_n e^{j2n\pi\frac{t_i}{T}} \tag{3.6}$$

and

$$\alpha_n = \sum_{i=0}^{N-1} \mathbf{z}(2\pi\frac{t_i}{T})e^{-jn2\pi\frac{t_i}{T}}\frac{\Delta t_i}{T} \tag{3.7}$$

where t_i is the parameter for the ith sample point, $\Delta t_i = t_{i+1} - t_i$ denotes the deviation of the parameter, and N is the number of samples. DFT transforms the series

of discrete data from the time domain, $\mathbf{z}(2\pi\frac{t_i}{T})$, to the frequency domain, v_n. The frequency for DFT is $\omega_n = 2\pi n/T$, and both amplitude and phase spectrums have definition identical to those for the continuous function. If the periodic function $\mathbf{z}(t)$ are uniformly sampled, then $\frac{t_i}{T}$ is simplified to be $\frac{i}{N}$ and $\frac{\Delta t_i}{T}$ becomes $\frac{1}{N}$; in contrast, if the sample is non-uniform, Δt_i captures the sample intervals.

3.3 Curve Fitting with Finite Fourier Series

This section reviews the approximation of a complex curve function with finite Fourier series in complex form. When a closed curve is defined by a sequence of points rather than a continuous curve, a Discrete Fourier Transform (DFT) is used instead of the continuous Fourier Transform. The forward and inverse DFT of a finite-length sequence of *N* data points of a uniformly sampled, periodic function $\mathbf{z}(t) = x(t) + jy(t)$ are represented as:

$$\mathbf{z}(\frac{2\pi i}{N}) = \sum_{n=0}^{N-1} \alpha_n e^{nj\frac{2\pi i}{N}}, \tag{3.8}$$

where the Fourier coefficients are given by

$$\alpha_n = \frac{1}{N}\sum_{i=0}^{N-1} \mathbf{z}(\frac{2\pi i}{N}) e^{-nj\frac{2\pi i}{N}}. \tag{3.9}$$

Now consider the problem of approximating a curve segment $\mathbf{z}(t)$ by a truncated (or finite) series of sinusoidal functions in complex form:

$$\mathbf{z}(t) \approx \sum_{k=-p}^{+p} \alpha_k e^{jk\omega_0 t} \tag{3.10}$$

where p, is a small positive integer that defines the maximum order of the harmonic terms used in the approximation. Such a curve is also known as a trigonometric polynomial curve of order p in the field of Computer Aided Geometric Design [67, 68]. The goal is to find the coefficient α_k so that the truncated series best approximates the given curve segment $\mathbf{z}(t)$. In this case, DFT is in general not applicable, since the function $\mathbf{z}(t)$ represents an open curve segment and is therefore not periodic. In this paper, we use a least-squares curve fitting approach to obtain the coefficients α_k. The resulting coefficients are not Fourier coefficients in general. But when the least-squares curve fitting procedure is applied to a periodic function, the resulting coefficients become Fourier coefficients. For the sake of convenience, we use the term Fourier coefficients to describe coefficients α_k associated with sinusoidal function series, regardless they are "proper" Fourier series or not.

We formulate the least squares problem by defining the error function:

$$\Delta = \sum_{i=1}^{n} ||\mathbf{z}(t_i) - \sum_{k=-p}^{+p} \alpha_k e^{jk\omega_0 t_i}||^2 \tag{3.11}$$

in which $||\cdot||$ denotes the magnitude of a vector, $\mathbf{z}(t_i)$ $(i = 1, \ldots, n)$ denote n points that are sampled from the original curve $\mathbf{z}(t)$. Then the error function, Δ, can be rewritten as

$$\Delta = \sum_{i=1}^{n} (X_i - x_i)^2 + \sum_{k=0}^{n} (Y_i - y_i)^2 \tag{3.12}$$

where x_i and y_i are the real and the imaginary part of $\mathbf{z}(t_i)$ respectively, and X_i and

Y_i are given by

$$X_i = \Re\{\sum_{k=-p}^{+p} \alpha_k e^{ik\theta_i}\} = \sum_{k=-p}^{+p} (a_k \cos k\theta_i - b_k \sin k\theta_i) \tag{3.13}$$

$$Y_i = \Im\{\sum_{k=-p}^{+p} \alpha_k e^{ik\theta_i}\} = \sum_{k=-p}^{+p} (b_k \cos k\theta_i + a_k \sin k\theta_i) \tag{3.14}$$

wherein $\alpha_k = a_k + jb_k$ and $\theta_i = \omega_0 t_i$

To minimize the error Δ by selecting the variables a_m and b_m ($m = -p, \cdots, p$), we require:

$$\frac{\partial \Delta}{\partial a_m} = 0, \quad \frac{\partial \Delta}{\partial b_m} = 0 \tag{3.15}$$

for all $m \in [-p, \cdots, p]$. Rewrite the above necessary conditioins in complex form to obtain

$$\frac{\partial \Delta}{\partial a_m} + j\frac{\partial \Delta}{\partial b_m} = 0 \tag{3.16}$$

and we can represent the conditions with a complex matrix equation

$$\Omega \mathbb{X} = \mathbb{Y} \tag{3.17}$$

where

$$\mathbb{X} = \underset{m \rightarrow}{\left[\cdots, \quad \alpha_m, \quad \cdots \right]^T} \tag{3.18}$$

$$\Omega = \begin{bmatrix} & \cdots & \\ \vdots & \sum_{i=0}^{n} e^{j(k-m)\theta_i} & \vdots \\ & \cdots & \\ & k \rightarrow & \end{bmatrix} \quad m \downarrow \tag{3.19}$$

$$\mathbb{Y} = \underset{m \rightarrow}{\left[\cdots, \quad \sum_{i=0}^{n} \mathbf{z}(t_i) e^{-jm\theta_i}, \quad \cdots \right]^T} \tag{3.20}$$

and k and m vary from $-p$ to p, respectively.

To obtain the best approximation of the curve $\mathbf{z}(t)$ with the finite Fourier series, we seek to solve $\mathbb{X}$ from Eq. (3.17). The use of LU decomposition is recommended to avoid potential numerical problems.

Chapter 4

Kinematic Convexity of Planar Displacements Based On an Approximately Bi-Invariant Metric

The chapter deals with the concept of convexity from the viewpoint of computational kinematic geometry. The goal is to develop the concept of kinematic convexity as a basic tool for geometric computations involving planar displacements. Much of the existing work on kinematic geometry deals with rigid-body motions generated by mechanisms and robot manipulators (see, for example, Hunt [3], Bottema and Roth [5], McCarthy [8]). In recent years, researchers have sought to bring together the fields of kinematics and Computer Aided Geometric Design (CAGD) for the development of freeform parametric motions. These parametric motions are typically defined by combining Bézier or B-spline representation in CAGD with representations of rigid-body displacements in kinematics such as quaternions, dual quaternions, Lie groups and Lie algebra (Ge and Ravani [69, 70], Purwar and Ge [71], Park and Ravani [72], Kim and Nam [73], Jüttler and Wagner [74], Zefran and Kumar [75], Srinivasan and Ge [76]. Applications of these freeform motions include motion animation in computer graphics, spatial navigation in virtual reality

as well as path planning in robotics and 5-axis CNC machining. More recently, Zhe and Ge [77], and Purwar et al[49, 50]. have also studied the problem of motion interpolation under kinematic constraints for planar and spherical 2R, 3R open chains, as well as 6R closed chains.

In this chapter, we seek to develop the notion of kinematic convexity as a basic tool for handling geometric computations for applications mentioned above. The scope of this chapter is limited to planar kinematics. It follows our previous work [78] on the kinematic convexity of $SO(3)$ (i.e., special orthogonal group of spherical displacements) and seeks to extend the concept to $SE(2)$ (i.e., special Euclidean group of planar displacements). In difference to $SO(3)$ which has a bi-invariant metric, $SE(2)$ does not permit such a metric [79]. To address the issue of distance metric for planar displacements, we explore the connection between planar quaternions and quaternions and formulate the concept of kinematic convexity in the space of quaternions where a bi-invariant metric exists.

The organization of the chapter is as follows. Section 2 reviews the concept of planar quaternions and introduced dimensionless planar quaternions. Section 3 explores the connection between dimensionless planar quaternions and double quaternions. Section 4.1 studies the notion of kinematic convexity of a set of planar displacements. Section 4.4 presents an example to illustrate the use of kinematic convexity for estimating the "closest distance" from a fixed body to a moving body undergoing a rational Bézier motion.

4.1 Kinematic Convexity of Planar Displacements

In our recent work [78], we presented the notion of kinematic convexity for $SO(3)$. In this chapter, we take advantage of the aforementioned connection between $SE(2)$ and $SO(3)$ and formulate the concept of kinematic convexity for planar displacements in the space of quaternions where a bi-invariant metric exists. This results in the treatment of kinematic convexity of planar displacements with a metric that is approximately bi-invariant.

From the perspective of projective geometry, the quaternion **G** as given by (2.12) can also be viewed as a set of homogeneous coordinates defining a point in projective three-space. If we normalize **G** using (2.13) so that it becomes a unit quaternion, then **G** defines a spherical displacement; if we normalize **G** so that $w = 1$, i.e., so that $G_3^2 + G_4^2 = 1$, then we obtain a planar quaternion **F**. Thus projective concepts such as projective convexity presented in [78] are directly applicable to the image space of planar displacements. It is only when we are concerned with the metric geometry of the image space, we need to consider how to normalize the quaternion components of **G**. To envision the image, **G** is dehomogenized via its projection onto the hyperplane $G_4 = 1$ which results a three dimensional point $(G_1/G_4, G_2/G_4, G_3/G_4)$.

4.1.1 Convex Combination of Two Planar Displacements

We consider the image space defined by planar quaternions (2.4) as a projective three-space P^3. A set in the space is said to be convex if it contains every line-segment whose end points are in the set. The key here is to clarify the meaning of

"line-segment", especially in the case of antipodal points. To this end, we follow Stolfi [80] and Ge et al. [78] and consider the image space as an oriented projective space. Algebraically, this means that use *signed homogeneous coordinates*, i.e., $w\mathbf{F}$ and $\mathbf{F}$ represent the same point (or planar displacement) if $w > 1$ and they represent two oppositely sensed points (or planar displacements) if $w < 1$. This allows an unambiguous definition of a line-segment from $\mathbf{G}_0$ to $\mathbf{G}_1$.

Given two image point pairs $\mathbf{G}_i$ $(i = 0, 1)$, the line-segment joining $\mathbf{G}_0$ and $\mathbf{G}_1$ is given by the following linear combination:

$$\mathbf{G}(\alpha_0, \alpha_1) = \alpha_0\mathbf{G}_0 + \alpha_1\mathbf{G}_1 \tag{4.1}$$

where we restrict the sum $\mathbf{G}(\alpha_0, \alpha_1)$ to be nonzero and we require the real coefficients α_i to be non-negative. The "nonzero sum" restriction implies that $(\alpha_0, \alpha_1) \neq (0, 0)$ when the two points are non-antipodal and that the ratio $\alpha_1 : \alpha_0$ is not inversely proportional to the weight ratio $w_0 : w_1$ associated with $\mathbf{G}_i(= w_iF_i)$ when the two points of $\mathbf{G}_i$ are antipodal.

Now consider a spherical model of P^3 for $\mathbf{G}$. For $\mathbf{G}$, if the two points, $\mathbf{G}_0$ and $\mathbf{G}_1$, are distinct and non-antipodal, then line-segment $\mathbf{G}_0\mathbf{G}_1$ is the shorter of the two great circular arcs connecting them. If the two points are identical, then the line-segment reduces to a single point; if the two points are antipodal, the line-segment reduces to two antipodal points. Each of Figure 4.1 and 4.2 shows the convex hull of two distinct, non-antipodal displacements.

A line-segment in P^3 corresponds to a motion segment, called *convex hull of the two planar displacements*. When the two oriented points are identical, the convex hull reduces to a single displacement; when the two oriented points are antipodal,

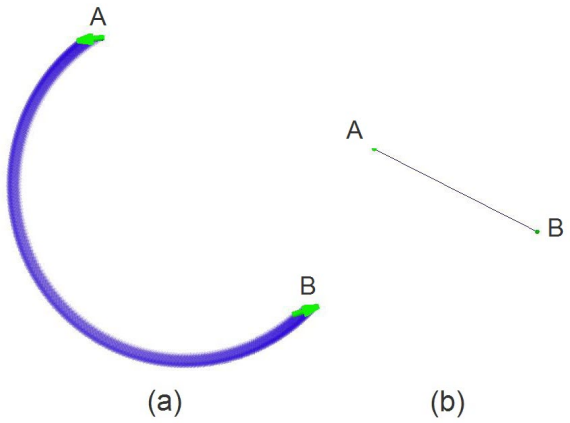


Figure 4.1: (a) Convex hull of two displacements $\mathbf{G}_0$ and $\mathbf{G}_1$ such that relative angle between them is less than but close to π so the path is close to a half turn. (b) Linear interpolations in the space defined by $\mathbf{G}$ (and $\mathbf{H}$), respectively.

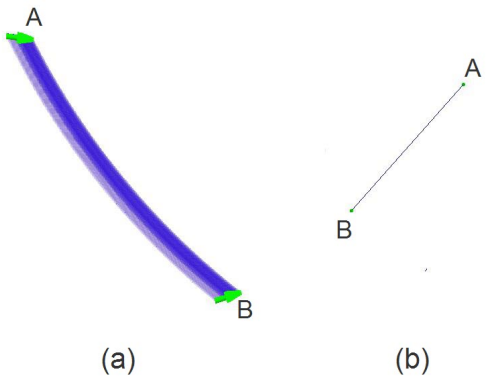


Figure 4.2: (a) Convex hull of two displacements $\mathbf{G}_0$ and $\mathbf{G}_1$, where the relative angle between them is small so the path is close to translation (b) Linear interpolations in the space defined by $\mathbf{G}$ (and $\mathbf{H}$), respectively.

the convex hull consists of two displacements that are geometrically identical but with opposite sense of orientation.

4.1.2 Convex Sets of Planar Displacements

In what follows we exclude geometrically equivalent but oppositely sensed displacements, when defining a set of oriented displacements. This leads to the following definition:

Definition A set Z of planar displacements is convex if it is not empty, and for any pair $\mathbf{G}_0$ and $\mathbf{G}_1 \in Z$, the convex hull of $\mathbf{G}_0$ and $\mathbf{G}_1$) is also in Z.

Let a set of planar displacements be represented by a set of point pairs $\mathbf{G}_i$ ($i = 0, 1, 2, \ldots, n$) in P^3. These points form one simplex S in P^3 with $\mathbf{G}_i$ as their vertices. Similar to the affine definition of convex combination, we define the following as a convex combination of the set of points in P^3:

$$\mathbf{G} = \sum_{i=0}^{n} \alpha_i \mathbf{G}_i \tag{4.2}$$

where $\alpha_i \geq 0$ for all i but $(\alpha_0, \alpha_1, \ldots, \alpha_n) \neq (0, 0, \ldots, 0)$. The point $\mathbf{G}$ lies inside of the simplex S. All convex combinations of the set of points $\mathbf{G}_i$ span a convex set called the convex hull of this point set. Kinematically, every point $\mathbf{G}$ of the convex hull in P^3 defines a planar displacement that belongs to the convex hull of the set of planar displacements represented by $\mathbf{G}_i$ ($i = 0, 1, 2, \ldots, n$). For example, Figure 4.3 illustrates the convex hull of three planar displacements.

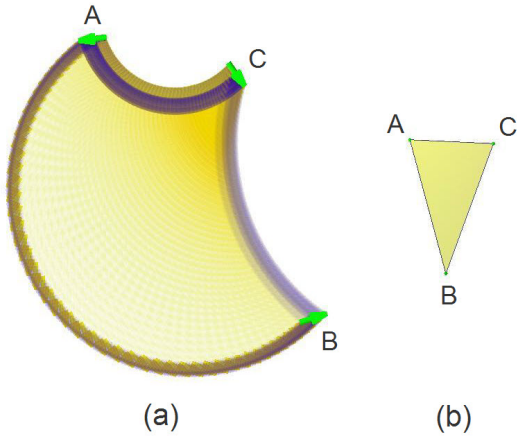


Figure 4.3: Convex hull of three displacements corresponding to a triangle-segment in oriented projective three-space

4.2 A Projective Test for Kinematic Convex Hull

For the projective space P^3 of planar displacements, the convex hulls of both **G** can be formed by the set of all simplices (S_4) of corresponding components of the four displacements from the given set. Any image point inside this simplex is not on the boundary of that hull. Stolfi [80] proposed a simple test for the localization of points relative to the flat spanned by simplices. This section applies this test to the planar displacements.

A given displacement $\mathbf{G}_*$, where $\mathbf{G}_* = (G_1^*, G_2^*, G_3^*, G_4^*)$, can be located with respect to their simplex S_4 by calculating the signs of the coefficients α_i $(i = 0 \ldots 3)$ for linear combination from the following system of linear equations:

$$(\alpha_0\, \alpha_1\, \alpha_2\, \alpha_3) \begin{bmatrix} G_{0,1} & G_{0,2} & G_{0,3} & G_{0,4} \\ G_{1,1} & G_{1,2} & G_{1,3} & G_{1,4} \\ G_{2,1} & G_{2,2} & G_{2,3} & G_{2,4} \\ G_{3,1} & G_{3,2} & G_{3,3} & G_{3,4} \end{bmatrix} = (G_1^*, G_2^*, G_3^*, G_4^*) \qquad (4.3)$$

where $(G_{i,1}, G_{i,2}, G_{i,3}, G_{i,4})$ denote coordinates of the $i^{th}(i = 0 \ldots 3)$ displacement. The sequence of signs of α_i, given by $\rho_0\rho_1\rho_2\rho_3$; $\rho_i = sign(\alpha_i)$ is also called signature of the component $\mathbf{G}$ relative to it simplex. The signature $+ + ++$ implies that the point lies strictly inside the S_4, while the signature $- - --$ implies that the point lies in the antipodal image of the simplex ($\sim S_4$).

Now, we present a rather naïve algorithm that combines the slow algorithm of O'Rourke [81] and the projective test as devised above to obtain the set of displacements that make up the boundary of the convex hull.

Algorithm: Given a displacement set $Z = \{\mathbf{G}_0, \mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_{n-1}\}$, classify those displacements from set Z that lie inside the convex hull of Z and add to a new set Z^+.

```
for each i do
   for each j ≠ i do
      for each k ≠ j ≠ i do
         for each l ≠ k ≠ j ≠ i do
            for each m ≠ l ≠ k ≠ j ≠ i do
               if G_m ∈ S_4(G_i, G_j, G_k, G_l)
               then add G_m to Z^+.
```

Since Z^+ is the set of non-extreme displacements (defined by all positive signature sequence), difference set $Z - Z^+$ gives the desired set of displacements lying on the boundary of the convex hull. A much more practical algorithm for determining convex hull in higher dimensions is a similar to randomized incremental algorithm called "Qhull" (Barber et. al. [82]) that can be used for much faster computation on

moderately sized input sets.

4.3 Projective Representation of Planar Bézier Motions

As an application of the kinematic convexity, we now present a projective representation of planar Bézier motions by recursive application of the linear combination (4.1).

Let $\mathbf{b}_0, \mathbf{b}_1, \cdots, \mathbf{b}_n \in P^3$ denote a set of oriented image point that represent a set of planar displacements. Let $\alpha_i \geq 0$ $(i = 0, 1)$ but $(\alpha_0, \alpha_1) \neq (0, 0)$. Set

$$\mathbf{b}_i^r(\alpha_0, \alpha_1) = \alpha_0 \mathbf{b}_i^{r-1} + \alpha_1 \mathbf{b}_{i+1}^{r-1}; \quad r = 1, \cdots, n, \quad i = 0, \cdots, n-r. \tag{4.4}$$

When $r = 0$, $\mathbf{b}_i^0 = \mathbf{b}_i$. When $r = n$, $\mathbf{b}_0^n(\alpha_0, \alpha_1)$ traces out a Bézier curve in P^3 (denoted by $\mathbf{b}^n$) as (α_0, α_1) varies. Every image point on the Bézier curve corresponds to a planar displacement that belong to a Bézier planar motion. The points $\mathbf{b}_i$ are called *Bézier points* or *control points* of the Bézier curve. The planar displacements that they represent are called *Bézier planar displacements*. It is clear that the Bézier planar motion must lie in the convex hull of the Bézier planar displacements.

Now let us impose an additional requirement that $\alpha_0 + \alpha_1 = 1$. This implies that we treat the Bézier points as points in affine geometry. After letting $\alpha_1 = t$ and $\alpha_0 = 1 - t$, Then Eq.(4.4) becomes the following well-known linear interpolation:

$$\mathbf{b}_i^r(t) = (1-t)\mathbf{b}_i^{r-1}(t) + t\mathbf{b}_{i+1}^{r-1}(t). \tag{4.5}$$

The Bézier curve $\mathbf{b}_0^n(t)$ in this case defines a rational Bézier planar motion of degree $2n$.

4.4 Estimation of Kinematic Separation

The problem that we seek to solve is this: given a rigid body planar Bézier (or, B-spline) motion $\mathcal{M}(t)$ in a plane and a fixed obstacle $\mathcal{P}$ (also in the plane), is there an position of the motion kinematically similar to the obstacle or what is the shortest distance between the trajectory of the moving object and the obstacle? To answer these questions, we first kinematically map $\mathcal{M}(t)$ to $\mathbf{M}(t)$ and $\mathcal{P}$ to $\mathbf{P}$, respectively in the image space of planar displacements. Then, we form the convex hull of $\mathbf{M}(t)$ and check for the location of $\mathbf{P}$ relative to the convex hull by using projective test as given in the section 4.2. Depending on the location of $\mathbf{M}$, either we subdivide the convex hull and repeat the process or stop, as further described below.

Without loss of generality, we can assume that in the image space, the Bézier control points of the motion are unit double quaternion and the curves of $\mathbf{G}$ and $\mathbf{H}$ are cubic. In this case, the two convex hulls of the image curves are a four-dimensional tetrahedron embedded on the surface of a 4-D unit hypersphere. Using Lagrange multiplier method for constrained minimization, the nearest location on the tetrahedron from $\mathbf{P}$ and the shortest distance between the two is calculated. To get a good approximation to the location on the trajectory and the shortest distance, the convex hulls are subdivided and the shortest distance calculations are done on each subdivided segment repeatedly until either a certain number of subdivisions have been performed or the relative difference in minimum distance between two successive iterations falls below a predefined number. As the convex hull is subdivided further, it approaches the swept volume of the moving object.

Consider a cubic Bézier curve in the image space, which corresponds to a de-

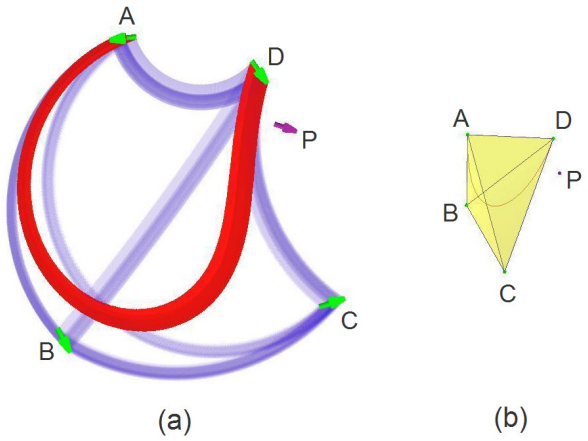


Figure 4.4: 6^{th} degree spherical Bézier motion $\mathcal{M}(t)$, convex hull boundary, obstacle $\mathcal{P}$

gree six motion in $SE(2)$ given by four quaternions $\mathbf{G}; i = 0 \ldots 3$, and another quaternion $\mathbf{P}$, which corresponds to the transformation of an obstacle from canonical coordinate frame to its current location. The generic point $\mathbf{G}$ in the convex hull of $\mathbf{G}_i$ $(i = 0 \ldots 3)$ is given by Eq. (4.2):

$$\mathbf{G} = \alpha_0\mathbf{G}_0 + \alpha_1\mathbf{G}_1 + \alpha_2\mathbf{G}_2 + \alpha_3\mathbf{G}_3, \tag{4.6}$$

where $\alpha_i \geq 0; i = 0 \ldots 3$ and not all $\alpha_i = 0$.

If $\mathbf{G}$ is to be the nearest location from $\mathbf{P}$ the objective is to:

$$maximize\langle \mathbf{G}, \mathbf{P} \rangle^2 \tag{4.7}$$

subject to the constraint $||\mathbf{G}|| = 1$, where $\langle .. \rangle$ denotes the inner product, and

$$\langle \mathbf{G}, \mathbf{P} \rangle = \alpha_0\mathbf{G}_0.\mathbf{P} + \alpha_1\mathbf{G}_1.\mathbf{P} + \alpha_2\mathbf{G}_2.\mathbf{P} + \alpha_3\mathbf{G}_3.\mathbf{P}. \tag{4.8}$$

Using the Lagrange multiplier, we modify the above objective function to La-

grangian function E that incorporates the constraint:

$$E = \alpha_0 \mathbf{G}_0.\mathbf{P} + \alpha_1 \mathbf{G}_1.\mathbf{P} + \alpha_2 \mathbf{G}_2.\mathbf{P} + \alpha_3 \mathbf{G}_3.\mathbf{P} - \lambda(\mathbf{G}^T\mathbf{G} - 1), \tag{4.9}$$

where λ is the Lagrange multiplier.

Optimization of Lagrangian function E require:

$$\frac{\partial E}{\partial \alpha_i} = \mathbf{G}_i.\mathbf{P} - 2\lambda \mathbf{G}.\frac{\partial \mathbf{G}}{\partial \alpha_i} = 0. \tag{4.10}$$

The solution to Eq. (4.10)and the condition that $\mathbf{G}^T\mathbf{G} = 1$ has the same prototype as that of Eq. (4.10), given by the following equations:

$$\alpha_i = c_i/2\lambda, \tag{4.11}$$

where

$$c_0 = \begin{vmatrix} m_0 & f_1 & f_2 & f_3 \\ m_1 & 1.0 & f_4 & f_5 \\ m_2 & f_4 & 1.0 & f_6 \\ m_3 & f_5 & f_6 & 1.0 \end{vmatrix},$$

$$c_1 = \begin{vmatrix} 1.0 & m_0 & f_2 & f_3 \\ f_1 & m_1 & f_4 & f_5 \\ f_2 & m_2 & 1.0 & f_6 \\ f_3 & m_3 & f_6 & 1.0 \end{vmatrix},$$

$$c_2 = \begin{vmatrix} 1.0 & f_1 & m_0 & f_3 \\ f_1 & 1.0 & m_1 & f_5 \\ f_2 & f_4 & m_2 & f_6 \\ f_3 & f_5 & m_3 & 1.0 \end{vmatrix},$$

$$c_3 = \begin{vmatrix} 1.0 & f_1 & f_2 & m_0 \\ f_1 & 1.0 & f_4 & m_1 \\ f_2 & f_4 & 1.0 & m_2 \\ f_3 & f_5 & f_6 & m_3 \end{vmatrix},$$

$$\lambda = \frac{1}{2}\sqrt{(\sum_{i=0}^{m} c_i G_{i,2})^2 + (\sum_{i=0}^{m} c_i G_{i,3})^2},$$

and

$$\begin{array}{lll} m_i = \mathbf{G}_i.\mathbf{P} & (i = 0 \ldots 3), & \\ f_1 = \mathbf{G}_0.\mathbf{G}_1, & f_2 = \mathbf{G}_0.\mathbf{G}_2, & f_3 = \mathbf{G}_0.\mathbf{G}_3, \\ f_4 = \mathbf{G}_1.\mathbf{G}_2, & f_5 = \mathbf{G}_1.\mathbf{G}_3, & f_6 = \mathbf{G}_2.\mathbf{G}_3. \end{array} \tag{4.12}$$

If $\alpha_i \geq 0$ for $i = 0 \ldots 3$, then the optimal $\mathbf{G}$ can be obtained from Eq. (4.6). The approximate shortest distance is given by $\arccos^2(\mathbf{G} \cdot \mathbf{P})$. If all α_i are not non-negative, then the projection point of $\mathbf{G}$ is not inside the convex hull of $\mathbf{G}_i; i = 0 \ldots 3$. In this case, the optimal point may fall on one of the faces, edges, or the vertices of the tetrahedron. So does the nearest position locate on the boundary of the convex hulls. The nearest location method presented above can be generalized to a simplex of any dimension. Expressions for the nearest location to the faces and edges can be obtained in a similar fashion and is not included here due to space limitations.

The problem of calculating the normal distance from a point to the image curve has been effectively solved by Ravani and Roth [83, 53]. The resulting technique has been refined and extended by Larochelle and McCarthy [**?**]. The algorithm proceeds as follows:

1. Consider **P** as a test displacement and locate it relative to the convex hull by solving for coefficients given by Eq. (4.3).

2. If all the coefficients are positive, **P** is inside the convex hull. Set shortest distance to zero and the nearest point to **P** itself. If any of the coefficients are non-positive, use the shortest distance method to find the nearest boundary of

the tetrahedron. It can be one of the four faces or six edges or four vertices. Record the distance and the nearest location.

3. Subdivide the motion recursively, form their convex hull, and repeat from step (1) for each subdivided convex hull unless the stopping condition has been reached, which can be either a preset number of subdivisions or a user-defined bound on the relative reduction of distance in successive iterations.

In Fig. 4.4, we show a degree six planar Bézier motion $\mathcal{M}(t)$ of a gripper and an obstacle $\mathcal{P}$. We implement our subdivision based algorithm for determining the most likely place of collision on the trajectory of the moving object. The results are shown in Fig. 4.5. It can be seen that only after four levels of subdivision, the convex hull approximates the swept volume very closely and the nearest possible colliding location is indistinguishable from the moving object occupying that position at that instant. Once the potential colliding location is found, it is possible to apply geometry-based algorithms to exactly determine the interference points for the given geometric models of moving and stationary object.

4.5 Conclusions

In this chapter, we presented the concept of kinematic convexity for a set of planar displacements. We have shown that planar quaternions for planar displacements can be handled in much the same way as quaternions for spherical displacements. As an application, we presented a novel, simple, and efficient subdivision based algorithm for kinematic separation problem.

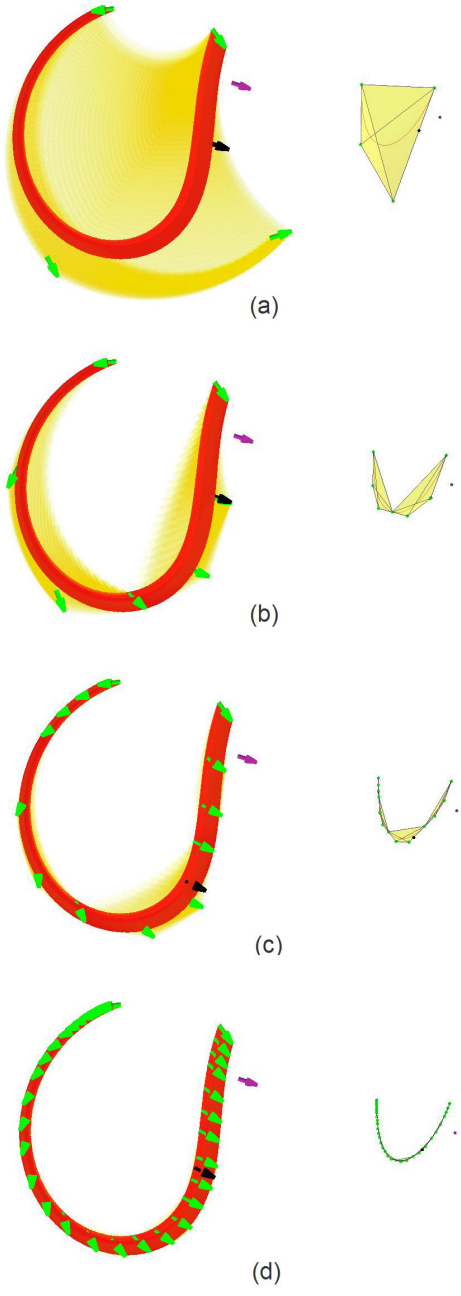


Figure 4.5: Convex hull of the motion $\mathcal{M}(t)$(Fig. 4.4) and the nearest colliding location $\mathcal{G}$ (a) in the beginning (b) after one subdivision (c) after two subdivisions (d) after three subdivisions

Chapter 5

Interactive Dimensional Synthesis and Motion Design of Planar 6R Single-Loop Closed Chains via Constraint Manifold Modification

Typical task-driven problems, also known as motion synthesis problems, seek to synthesize the mechanisms that guide a rigid body to navigate through, or as close as possible to a prescribed motion, which is usually specified as either a set of finite or a continuous time variant sequence of displacements. The problem that we solve here is as follows: given a prescribed planar rational motion, what are the dimensions of the links of a planar 6R closed chain that accomplishes the given task?

Theory of mechanisms synthesis is well-developed (see Sandor and Erdman [7], Suh and Radcliffe [45], and McCarthy [9]), and there have been a lot of academic research efforts in the development of software systems for the synthesis of planar, spherical, and spatial mechanisms (see KINSYN III from Rubel and Kaufamn [22], LINCAGES from Erdman and colleagues [23, 24], Kihonge et al. [25], Spades from Larochelle [26], Sphinx from Larochelle et al. [27], Sphinxpc from Ruth and

McCarthy [28], Osiris from Tse and Larochelle [29], Perez and McCarthy [30], Su and McCarthy [31], Synthetica from Su et al. [32]). In the commercial domain, SyMech [33] and WATT [34] are two well-known software systems for planar mechanisms design. Despite the availability of design packages, their implementation and interface for design of rigid body linkages mostly utilize a black-box like approach, which involves inputting the desired motion, hiding the design theories and algorithms, and outputting a mechanism, which makes the design procedures less intuitive. In this work, we have attempted to give rise to an intuitive design environment wherein the designer is not only able to synthesize a mechanism interactively by simple geometric manipulations in a higher dimensional space, but also learns to understand the relationship between the modifiable geometry and its apparent effect on the dimensions of the links.

In this chapter, we use planar quaternions to represent displacements; see Bottema and Roth [46] and McCarthy [8] for details on quaternion representation. Using the well-known kinematic mapping approach (Ravani and Roth [83, 53]), the given rational motion is transformed into a rational curve in the space of quaternions (also known as the image space of displacements). For details on rational motions, see Ge and Ravani [69, 70], Jüttler and Wagner [74], Wagner [84], Röschel [85], Purwar et.al. [86], and Purwar and Ge [71].

The general approach of the work presented in this paper is closely related to the kinematic mapping approach for dimensional synthesis of planar and spherical mechanisms pioneered by Ravani and Roth [83, 53]. Their work was followed by Bodduluri and McCarthy [87], Bodduluri [88], and Larochelle [89]. Burmester [90]

showed that a four-bar linkage can interpolate at most five given displacements exactly. For more than five displacements, usually approximation is required. Ravani and Roth's approach involved minimizing the distance error between the given displacements and the image curve of the chain. This resulted in approximate motion synthesis. Brunnthaler et.al. [91] 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 [92] used a parameterized constraint manifold and employed nonlinear optimization to give numerical methods for approximate motion synthesis of spherical open and closed chains. Here, we are employing the kinematic mapping for designing planar 6R closed chains that do not have a restriction on the number of interpolating displacements. This is feasible since the planar 6R closed chains have three degrees of freedom, and thus admit an infinite number of solutions.

This chapter seeks to study the dimensional synthesis problem from the perspective of constrained motion interpolation. Jin and Ge [47, 48] and Purwar et al. [49, 50, 51] have studied the problem of motion interpolation under kinematic constraints for planar and spherical 2R, 3R open, and 6R closed chains as well as spatial SS chains. By using quaternions and kinematic mapping approach they transformed the problem of constrained motion interpolation into designing a rational curve constrained to fit the constraint manifold of the chain. Starting with an initial unconstrained curve, they modify the curve using an iterative numerical method until it fits inside the constraint manifold. The current paper investigates the inverse problem, that is, to change the constraint manifold while keeping the

given rational curve fixed for the dimensional synthesis of planar 6R closed chains.

Our design method treats the 6R closed chains as mechanisms assembled using two open chains connected together at the ends. Each open chain imposes kinematic constraints that limit the positions and orientations of the object connected to the end link. In this paper, we use the algebraic form of the constraint manifold (McCarthy [8] and Ge [93]) for the planar 6R closed chains. Thus, the kinematic constraints are transformed into geometric constraints, and the given rational motion is transformed into a rational curve in the image space. This way, our problem reduces to finding the constraint manifold that accommodates the given rational curve. Algebraically, the kinematic constraints are derived in the inequality form, where the limits of the inequalities are functions of link lengths, while the constraint functions themselves incorporate parameters that describe the location of fixed and moving pivots and the location and orientation of the moving frame. However, instead of dealing with the mechanism parameters directly, we formulate our approach in terms of the geometric parameters of the constraint manifold. This, and the decoupled nature of the relationship between various geometric parameters and the mechanism parameters lends intuitiveness to our approach. We provide a design tool with a user-friendly interface (see Figs. 5.3, 5.4), using which the designer can interactively change parameters that govern the size, orientation, and the position of the constraint manifold in the 4D image space (we visualize the 4D space by projecting on a hyperplane) such that the manifold contains the input image curve. This tool also provides a way to change the image curve in case the design requirements can be relaxed. In the end, we design two open chains that simultaneously

satisfy the kinematic constraints and the motion requirements. A visual interpretation of this approach is that we try to find a close fit constraint manifold that will contain the given image curve entirely. The process of designing is fast, intuitive, and especially useful when a numerical optimization based approach would be computationally demanding, and mathematically difficult to formulate. We do not attempt to find an optimal solution, but a good solution that satisfies the input motion requirements. Our simple approach also provides a basis for students and early designers to learn and understand designing of mechanisms by simple geometric manipulations.

The organization of the chapter is as follows. Section 5.1 reviews the kinematic constraints of planar 6R closed chains using quaternions. Section 5.2 presents a method for the interactive dimensional synthesis of planar 6R closed chains, and establishes relationships between mechanism parameters and the manifold size, orientation, and position. Finally, we present an example and visualizations that demonstrate the method before giving concluding remarks.

5.1 Constraint Manifold for Planar Chains

In this section, we review the constraint manifold associated with the kinematic constraints of planar 6R closed chains (see (McCarthy [8] and Ge [93] for details). The kinematic constraints specify the positions and orientations obtainable by a certain link of the chain. Consider a planar 6R closed chain as shown in Fig. 5.1. In the figure, **F** and **M** mark the fixed and the moving frames, respectively. The fixed pivots A_1 and A_2 are located at (x_1, y_1) and (x_2, y_2), respectively, while the

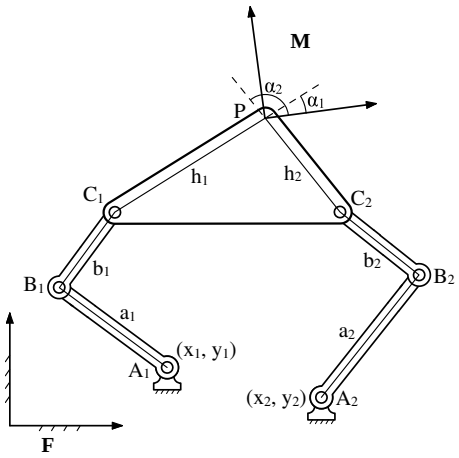


Figure 5.1: A planar 6R closed chain

moving frame is located at a distance of h_1 and h_2 from the two end pivots C_1 and C_2 respectively. The moving frame is assumed to be tilted by angles of α_1 from the line joining the end pivot C_1 and the origin of the moving frame, and α_2 from the line joining the end pivot C_2 and the origin of the moving frame. The length of the links is given by a_1, b_1, a_2, and b_2. A planar 6R closed chain can be seen as two 3R open chains (henceforth called left and right open chains) joined together at the ends. Then, the constraint manifold for the planar 6R closed chain is the intersection of constraint manifold of the two 3R open chains. If the displacement of the moving object attached to the moving frame is represented by a planar quaternion $Z = (Z_1, Z_2, Z_3, Z_4)$, then after eliminating the joint angles from the forward kinematics of each open chain, the algebraic equations for the two manifolds are given by:

Left 3R open chain:

$$\frac{(a_1 - b_1)^2}{4} \leq F_1(Z_1, Z_2, Z_3, Z_4) \leq \frac{(a_1 + b_1)^2}{4}, \tag{5.1}$$

where

$$F_1(Z_1, Z_2, Z_3, Z_4) = \frac{(Z_1 - \sigma_1 Z_3 - \tau_1 Z_4)^2 + (Z_2 - \sigma_2 Z_3 - \tau_2 Z_4)^2}{Z_3^2 + Z_4^2}, \tag{5.2}$$

and

$$\begin{array}{ll} \sigma_1 = (y_1 + h_1 \sin\alpha_1)/2, & \tau_1 = (x_1 + h_1 \cos\alpha_1)/2, \\ \sigma_2 = (-x_1 + h_1 \cos\alpha_1)/2, & \tau_2 = (y_1 - h_1 \sin\alpha_1)/2. \end{array} \tag{5.3}$$

Right 3R open chain:

$$\frac{(a_2 - b_2)^2}{4} \leq F_2(Z_1, Z_2, Z_3, Z_4) \leq \frac{(a_2 + b_2)^2}{4}, \tag{5.4}$$

where

$$F_2(Z_1, Z_2, Z_3, Z_4) = \frac{(Z_1 - \zeta_1 Z_3 - \eta_1 Z_4)^2 + (Z_2 - \zeta_2 Z_3 - \eta_2 Z_4)^2}{Z_3^2 + Z_4^2}, \tag{5.5}$$

and

$$\begin{array}{ll} \zeta_1 = (y_2 + h_2 \sin\alpha_2)/2, & \eta_1 = (x_2 + h_2 \cos\alpha_2)/2, \\ \zeta_2 = (-x_2 + h_2 \cos\alpha_2)/2, & \eta_2 = (y_2 - h_2 \sin\alpha_2)/2. \end{array} \tag{5.6}$$

Equations (5.1) and (5.4) characterize the kinematic constraints of a planar 6R closed chain, and define the constraint manifold for the chain.

To visualize the hypergeometric shape described by Eq.(5.1) or (5.4), we project it on the hyperplane $Z_4 = 1$. Denote $(z_1, z_2, z_3, 1)$ as the projected point of (Z_1, Z_2, Z_3, Z_4), both of which represent the same planar displacement. Then, $F_1(Z_1, Z_2, Z_3, Z_4)$ on $Z_4 = 1$ is given by

$$F_1(z_1, z_2, z_3, 1) = \frac{(z_1 - \sigma_1 z_3 - \tau_1)^2 + (z_2 - \sigma_2 z_3 - \tau_2)^2}{z_3^2 + 1}. \tag{5.7}$$

The volume described by Eq.(5.7) along with the limits of its inequalities creates implicit surfaces of (z_1, z_2, z_3). Setting $F_1(z_1, z_2, z_3, 1) = c$, where c is a constant,

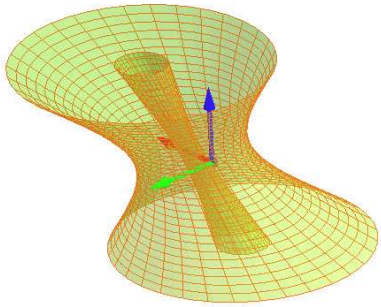


Figure 5.2: Visualization of the constraint manifold of a 3-R open chain as a pair of concentric, co-oriented, and sheared hyperboloid in the hyperplane $Z_4 = 1$; two surfaces indicate the limits of the inequality in Eq. (5.1)

and lies in the range $[\frac{(a_1-b_1)^2}{4}, \frac{(a_1+b_1)^2}{4}]$, we reorganize Eq. (5.1) to obtain:

$$\frac{(z_1 - \sigma_1 z_3 - \tau_1)^2}{c} + \frac{(z_2 - \sigma_2 z_3 - \tau_2)^2}{c} - z_3^2 = 1 \tag{5.8}$$

This is a sheared circular hyperboloid in the space parameterized by (z_1, z_2, z_3); see Fig. 5.2. This hyperboloid has its center at $(\tau_1, \tau_2, 0)$, and the central axis is given by $\frac{z_1-\tau_1}{\sigma_1} = \frac{z_2-\tau_2}{\sigma_2} = \frac{z_3}{1}$. Thus, the hyperboloid orients along the vector $(\sigma_1, \sigma_2, 1)$. It is evident that the center and the orientation are dependent on the location of the fixed pivot, the dimensions of the floating link, and the relative orientation of $\mathbf{M}$ from the floating link. The hyperboloid intersects plane $z_3 = 0$ in a circle, which has a radius, r, equal to $\sqrt{c}$. Thus, c determines the size of the hyperboloid. When the value of $F_1(z_1, z_2, z_3, 1)$ varies from its minimum to maximum, the size of the manifold increases correspondingly, but the center point and the orientation remain unchanged. Table 5.1 summarizes the relationship between geometric features of the hyperboloid and the mechanism parameters. A similar set of relationship exists for the other 3R open chain.

Geometric Features	Constraint Parameters
Center	$(\tau_1, \tau_2, 0)$
Orientation	$(\sigma_1, \sigma_2, 1)$
Size	$\frac{\|(a_1-b_1)\|}{2} \leq r = \sqrt{c} \leq \frac{(a_1+b_1)}{2}$

Table 5.1: Parameters for the Projective Sheared Hyperboloid

Thus, the constraint manifold of the planar 6R closed chains is given by two pairs of concentric and co-oriented sheared hyperboloid, and for a mechanism to pass through a given motion, the image curve would have to be contained within the constraint manifold.

5.2 Interactive Dimensional Synthesis

Our design method treats a 6R closed chain as a mechanism assembled using two independent open chains. The constraint manifolds of both the chains are geometric objects in the image space; the size, shape, and position of which are a function of mechanism parameters. A given rational motion maps to an image curve that needs to be contained inside these manifolds. In this section, we describe the procedure for designing planar mechanisms using our tool. We also describe the user interface with which the designer needs to be familiar. The basic idea is that the designers are provided a set of controls via the graphical user interface (GUI) of the tool that will allow them to interactively manipulate the constraint manifold with the objective to contain the image curve in the manifold. Upon being satisfied visually, the designer will be allowed to instruct the program to check if there are any violations of the

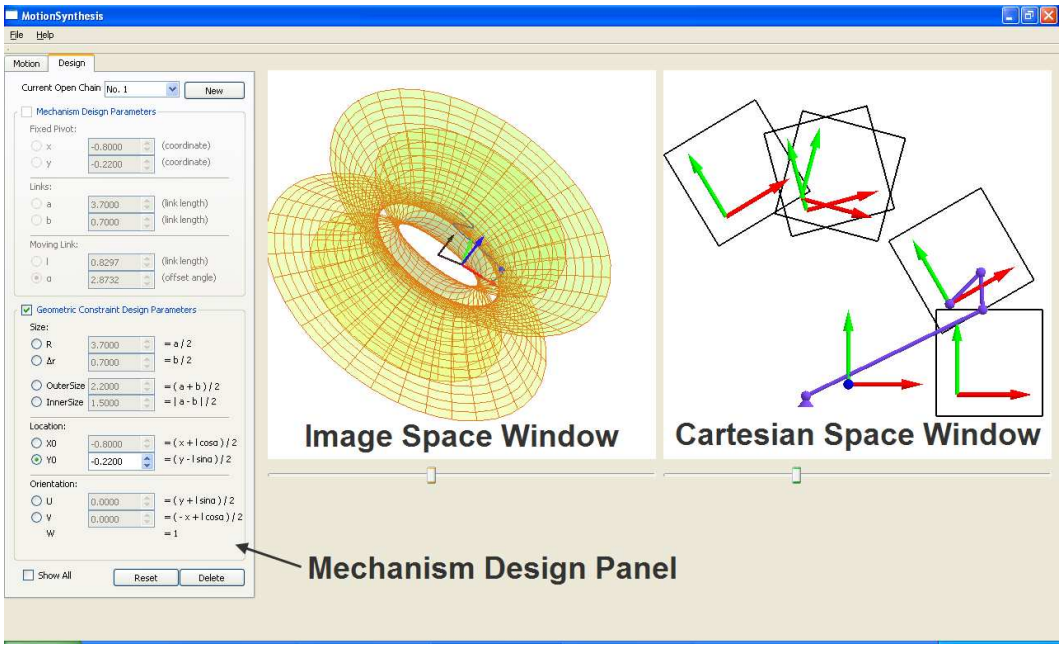


Figure 5.3: A screenshot of the panels and the window space

kinematic constraints. A windows binary of the tool for x86 architecture can be downloaded at `http://cadcam.eng.sunysb.edu/software.html`. We first present the functionalities of the user interface.

User Interface Functionalities

The GUI has four main parts, as shown in Figs. 5.3 and 5.4:

1. The Cartesian Space Window (CSW): This window is used to display the given positions and the animation of the open chains in the Cartesian space.

2. The Image Space Window (ISW): In this window, the constraint manifold as well as the image curve projected on the hyperplane $Z_4 = 1$ are shown.

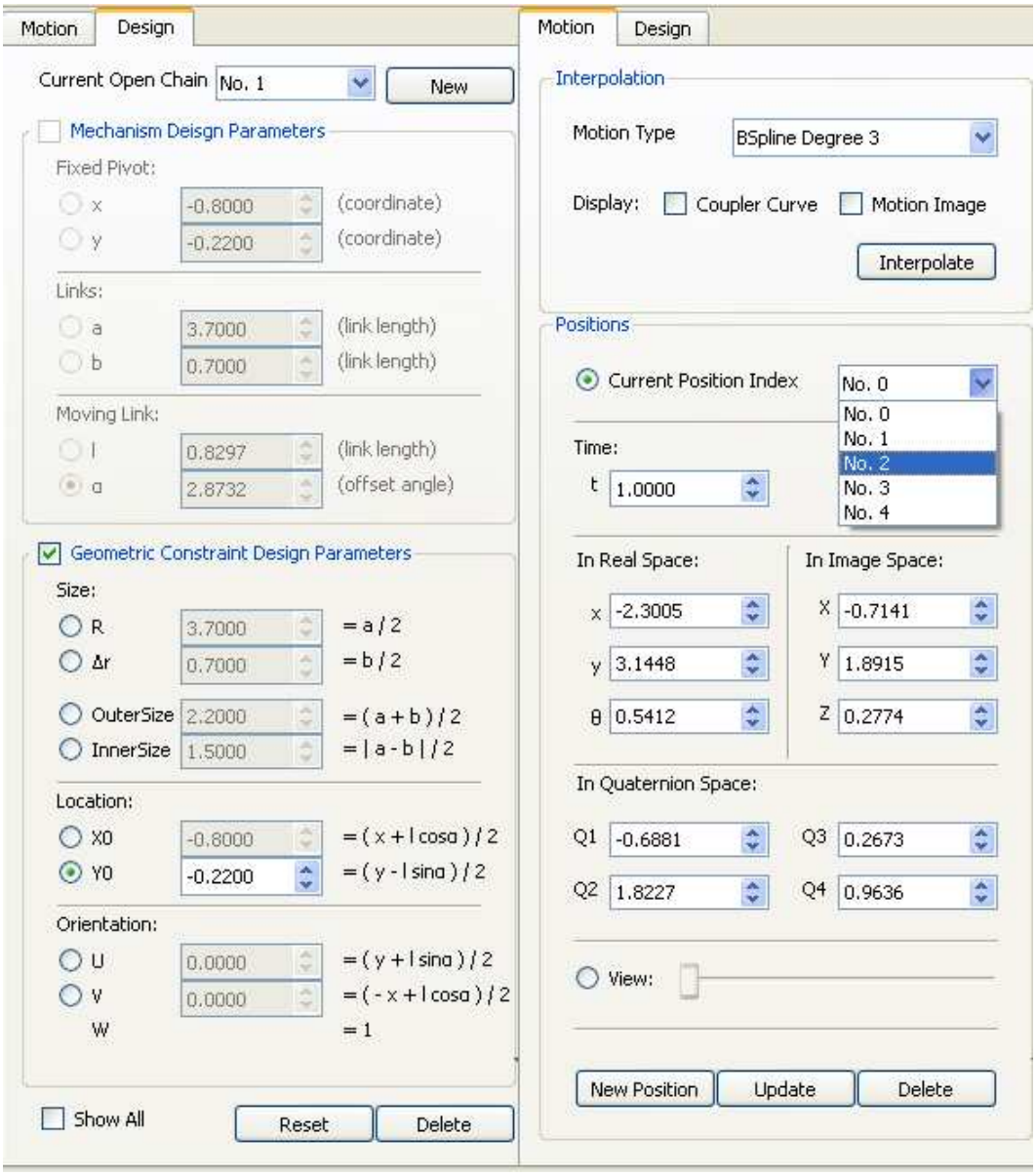


Figure 5.4: A screenshot of the Mechanism and Motion design panels

3. Motion Design Panel (MoDP): This panel supports operations like position insertion, deletion and modification, and comprises functions to animate the motion and to test for constraint violation. The constraint violation test is done using the Eqs.(5.1) and (5.4), and the test results are visualized through the user interface. This operation updates both the Cartesian Space Window and the Image space Window.

4. Mechanism Design Panel (MeDP): There are two ways to edit the mechanism: 1) directly manipulate mechanism parameters in the Cartesian space, like the location, the link lengthes and the relative angle, and as a consequence, constraint manifolds change in the image space, or 2) edit the geometric parameters that change the size, position, and the orientation of the manifolds. Designers may find the latter approach more intuitive. For either approach, there are six design variables. Equations (5.1) – (5.6), and Table 5.1 together describe the relationship between mechanism parameters and constraint manifold parameters. We summarize the relationships for one open chain as follows:

$$
\begin{aligned}
& x_1 = -\sigma_2 + \tau_1, \\
& y_1 = \sigma_1 + \tau_2, \\
& h_1 = \sqrt{\sigma_1^2 + \sigma_2^2 + \tau_1^2 + \tau_2^2}, \\
& \sin\alpha_1 = (\sigma_1 - \tau_2)/\sqrt{\sigma_1^2 + \sigma_2^2 + \tau_1^2 + \tau_2^2}, \\
& \cos\alpha_1 = (\sigma_2 + \tau_1)/\sqrt{\sigma_1^2 + \sigma_2^2 + \tau_1^2 + \tau_2^2}, \\
& \{a_1, b_1\} = \{(r_{\max} + r_{\min}), (r_{\max} - r_{\min})\},
\end{aligned}
\tag{5.9}
$$

where $r_{\max}$, and $r_{\min}$ describe the outermost and the innermost radius, respectively of the constraint manifold associated with an open 3R chain. A similar set of relationships exists for the other open chain.

Design Procedure

Now, we present the design procedure:

1. Use the Motion Design panel to input given positions, associated time parameters, and interpolate them using a NURBS motion.

   The given planar positions can be input with the time parameter t, using either planar quaternion coordinates (Z_1, Z_2, Z_3, Z_4), or Cartesian coordinates (x, y, θ). Once all the given positions are input, a cubic C^2 B-Spline motion (Eq. (2.14)) that interpolates the given positions is generated. Here, the time parameters are input to guarantee the smoothness of the motion. Consequently, the ISW shows the image points of the prescribed positions, and renders a smooth NURBS curve which passes through all the image points; while the CSW shows the given positions and the rational motion.

2. Switch to the Mechanism Design panel. Dimensional synthesis starts with two default 3R open chains. We describe the procedure for one open chain:

   In the CSW, initially, the fixed pivots are located at $(x_1, y_1) = (0, 0)$; the three links have unit length $a_1 = b_1 = h_1 = 1$, and the relative angle of $\mathbf{M}$ to the floating link is $\alpha_1 = 0$. With these initial choices and according to Eq. (5.3) and Table 5.1, the default hyperboloid pair is centered at $(0.5, 0, 0)$, its direction is parallel to vector $(0, 0.5, 1)$, and the inner boundary circle is of radius $r_{\min} = 0$, while the outer one has a radius $r_{\max} = 1$. Thus, in the ISW, a degenerate inner hyperboloid and a finite sized outer hyperboloid appear. At this point, the image curve may not be completely contained between the

two hyperboloids, which means that the constraints are being violated. We note that in general, initial parameters are difficult to select for body guidance problem solved using optimization approach; however, in our case this is not an issue because our approach relies on the interactive geometric manipulation of the constraint manifold. Changing the geometric parameters appropriately grows the inner hyperboloid, while shrinks the outer one. This approach provides the designer visual clues for the next step of parametric changes.

3. Modify the constraint manifold visually using the spinner controls (up and down arrows next to parameters) provided in the MeDP until the curve seems completely contained with a close fit between the pair of hyperboloids. Dragging the slider in either ISW or CSW verifies if the constraints are actually satisfied or not. Using the current value of the mechanism parameters, the program automatically checks the constraint equations as given by (5.1) and (5.4) to see if they are satisfied. When they are satisfied, the program outputs links' length, fixed and moving pivot locations, and the orientation of the moving frame.

4. Repeat steps 2, 3 and 4, and synthesize the other open chain.

5.3 Example

In this section, we present an example that demonstrates how our method interactively completes dimensional synthesis of a planar 6R closed chain using the

constraint manifold modification for a given degree 6 rational motion.

i	x	y	$\theta(^\circ)$	u_i
1	2.0448	-0.1940	0	0.0
2	1.9067	1.5029	30	0.2
3	-0.8894	3.4852	-15	0.5
4	-0.7851	3.2652	15	0.5
5	-2.3005	3.1448	31	1.0

Table 5.2: Cartesian Coordinates of Five Prescribed Positions along with Time Parameter Values

In this example, we use five positions as given in Table 5.2 and shown in the Cartesian Space Window in Fig. 5.3. The positions are given in Cartesian coordinates (x, y, θ), which specify the location and the orientation of the moving frame **M** relative to the fixed frame. Also given are the time parameter values (u_i) associated with each position. We note that time parameters are not required for rigid body guidance problem; we specify them here merely to construct an input B-spline motion. Our method of designing linkage is independent of time parameter values associated with the given positions.

First, the given positions are converted to planar quaternion representation (Z_1, Z_2, Z_3, Z_4) using Eq. (2.1), and then they are interpolated using a cubic B-spline formulation (Eq.(2.14)). This gives a degree 6 B-spline rational motion in the Cartesian space. The corresponding image curve is shown in the image space window of Fig 5.3. The image curve is visualized using Rodrigues parameters (see Bottema and Roth [46]) given by $(Z_1/Z_4, Z_2/Z_4, Z_3/Z_4)$. Hereafter, left and right 3R open chains, and their constraint manifolds are initialized, one at a time. However, navigating through the motion, it is found that the constraints are violated – this shows up as the image

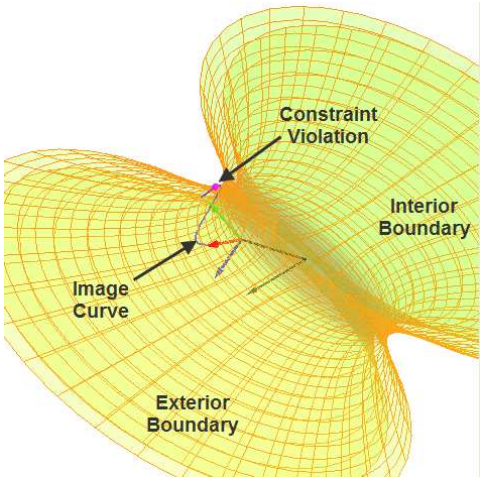


Figure 5.5: Constraint manifold of the left 3R Open Chain and the image curve; in this figure, the image curve is not completely contained inside the manifold. A point of violation is shown.

curve being outside the manifold; see Figs. 5.5 and 5.6. The designer next modifies the constraint manifolds interactively by varying its various geometric parameters, such as the size, position, and orientation with the objective to contain the image curve inside the manifolds. The process is intuitive and fast. Once the synthesis of two individual open chains is completed (see Figs. 5.7, 5.8), the assembly of them yields a 6R closed chain that interpolates the given five positions with a smooth motion. Table 5.3 and 5.4 list the design results, and the final mechanism is shown in Fig. 5.9.

5.4 Conclusions

In this chapter, we presented a simple and intuitive approach to dimensional synthesis of planar 6R closed chains. The approach is based on interactive manipulation

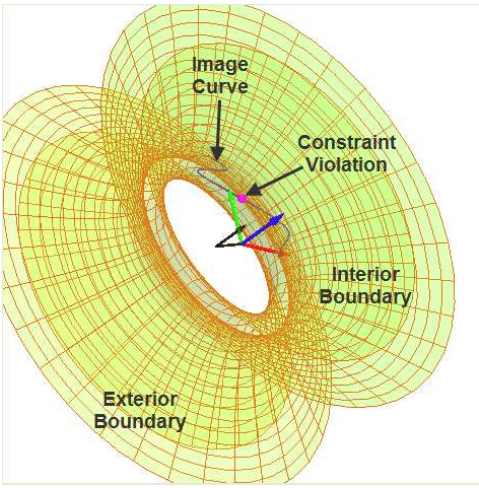


Figure 5.6: Constraint manifold of the right 3R Open Chain and the image curve; in this figure, the image curve is not completely contained inside the manifold. A point of violation is shown.

	x_1	y_1	a_1	b_1	h_1	$\alpha_1(^\circ)$
Left Chain	-1.6	1.7	5.5	0.3	2.9682	122.6
	x_2	y_2	a_2	b_2	h_2	$\alpha_2(^\circ)$
Right Chain	-0.8	-0.22	3.7	0.7	0.8297	164.6

Table 5.3: Synthesis Parameters of the 6R Planar Closed Chain

of constraint manifold associated with the mechanisms in a 3D environment. The approach is general, and can be extended to other spherical and spatial mechanisms for which constraint manifold are characterized by algebraic equations.

	σ_1	σ_2	τ_1	τ_2	$(a_1 - b_1)$	$(a_1 + b_1)$
Left Chain	1.6	0	-1.6	-0.9	2.6	2.9
	ζ_1	ζ_2	η_1	η_2	$(a_2 - b_2)$	$(a_2 + b_2)$
Right Chain	0	0	-0.8	-0.22	1.5	2.2

Table 5.4: Synthesis Parameters of the Constraint Manifolds

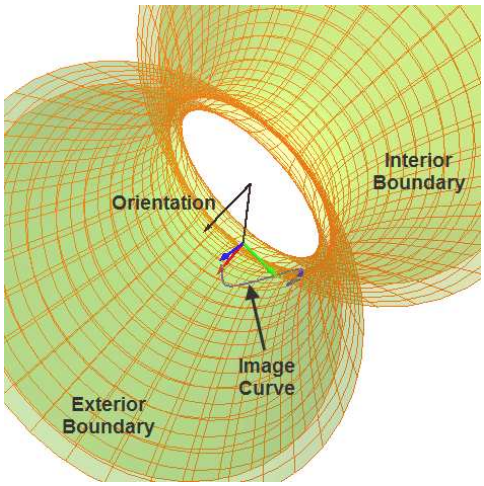


Figure 5.7: Constraint manifold of the left 3R Open Chain and the image curve; in this figure, the image curve is completely contained inside the manifold, thus implying that the constraints are not violated.

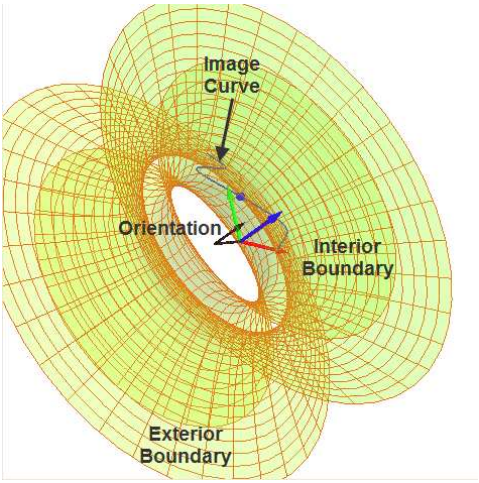


Figure 5.8: Constraint manifold of the right 3R Open Chain and the image curve; in this figure, the image curve is completely contained inside the manifold, thus implying that the constraints are not violated.

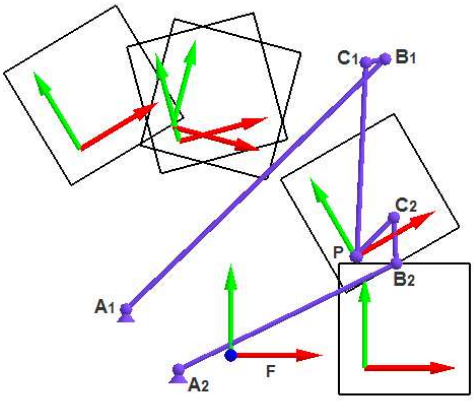


Figure 5.9: The final assembled mechanism

Chapter 6

A Fourier Descriptor Based Method for Four-Bar Linkage Synthesis for Generation of Open and Closed Paths

This chapter studies the problem of dimensional synthesis of planar four-bar linkages for path generation using Fourier series. Central to this problem is the formulation of an error function that captures the deviation of the generated path from the desired path in terms of nine independent variables associated with the design of a four-bar linkage. When an error function attempts to use all nine variables simultaneously and to compare the shape, size, location, and orientation of the two curves all at once, as is the case for the so-called *Structural Error Function*, it makes the resulting optimization routine far less efficient and sometimes even intractable [94].

In the field of *Computational Shape Analysis* [52], it is routine to process and simplify shapes before comparisons are made. The simplified representation of shapes is called *shape descriptor* or *shape signature*. The use of shape descriptors often simplifies the process of not only shape comparison but also shape storage and classification. There are many shape descriptors proposed for various appli-

cation domains. They include boundary scalar transform such as the centroid-to-boundary distance function, global scalar transform such as moment-based methods, and global space domain methods such as the media axis transform. Hoeltzel and Chieng [95] was one of the early adopters of computational shape analysis techniques for comparing the coupler curves for the dimensional synthesis of four-bar linkages. They used moment invariants to represent and compare coupler curves.

Of our interest is a Fourier transform based method for characterizing the shape function. Freudenstein [96] was the first to explore the use of Fourier transform for four-bar linkage analysis and synthesis. This work was followed by Funabashi [97], Farhang et al. [98, 99], Chu and Cao [100, 101, 102, 103], McGarva [104, 105], and Nie and Krovi [106]. As shown by Ullah and Kota [94], when reformulated using the Fourier descriptors, the error function that captures the deviation between the synthesized and desired path, leads naturally to a reduced search space that decouples the comparison of shapes from their size, location, and orientation. In addition, they observed that, for the case of four-bar coupler curves, the use of a small set of Fourier coefficients associated with the low-harmonic components is sufficient for shape comparison. This has been independently observed by cognitive scientists for human judgement of perceptual shape similarity[107]. The combined effect of decoupled search space with small set of Fourier coefficients leads to drastic improvement in optimization routine for four-bar linkage synthesis.

Recently, we found that when the path can be approximated by the first and second harmonics of Fourier series, the search space can be reduced even further and thus leads to an even more efficient method for four-bar linkage synthesis [108].

The Fourier descriptor based approach in its current form, however, is not without limitations. The representation is global in nature and requires a period function, which means a closed curve. In many applications for the path synthesis problem, however, only a segment of the specified path needs to be followed either by a non-Grashof or Grashof linkage. The purpose of the current paper is to extend the Fourier descriptor based method for four-bar linkage synthesis from closed curve to open curves and from Grashof linkage to non-Grashof four-bar linkages.

This work is motivated not only by the desire to extend and refine the Fourier based method for synthesizing four-bar linkages but also, more importantly, by the need to provide efficient tools for generating mechanically realizable geometric constraints. These tools will play an important role in a new constraint based paradigm for task centered design of mechanisms that bridges the gap between type synthesis and dimensional synthesis [109].

The organization of this chapter is as follows.Section 10.1 converts the loop closure equations of a four-bar linkage into a periodic function in complex form and then study the Fourier series approximation of the loop-closure equation in a form that is readily applicable to Fourier series based dimensional synthesis. Section 6.2 presents a general method for kinematic approximation of a curve segment using the coupler curve of a four-bar linkage. Both the curve segment and the loop closure equations are approximated using finite Fourier series. Section 6.3 presents numerical examples to demonstrate the validity of the proposed method and the developed algorithm.

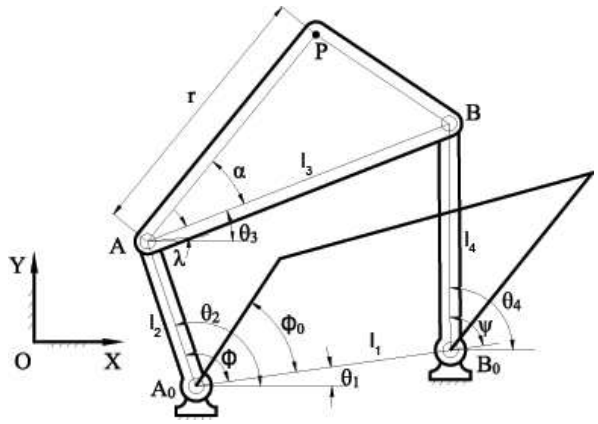


Figure 6.1: A crank-rocker mechanism

6.1 Fourier Approximation of Loop Closure Equation

Consider a planar four-bar mechanism shown in Figure 8.1 with XOY being the fixed coordinate frame. The fixed pivot A_0 is located at point (x_0, y_0). A_0B_0 is the ground link, and A_0A is the input link. The ith link has a length l_i and a position angle θ_i 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. In particular, let ϕ_0 denote the initial value of ϕ, which is the initial input angle. A point P on the coupler link AB is defined by the distance r and the angle α. Let the angular velocity of the input link be ω, then we have

$$\phi = \omega t + \phi_0. \tag{6.1}$$

The loop closure equation for the linkage in complex number form is given by

$$l_2 e^{j\phi} + l_3 e^{j\lambda} - l_1 = l_4 e^{j\psi}, \tag{6.2}$$

where ϕ, λ, and ψ are the input angle, coupler angle, and output angle relative to the fixed link A_0B_0. Here we seek to eliminate ψ and solve for $e^{j\lambda}$ as a function of the input angle ϕ. It is clear from (6.2) that its solution depends on link ratios $l_2/l_1, l_3/l_1, l_4/l_1$. The conjugate of (6.2) is given by

$$l_2 e^{-j\phi} + l_3 e^{-j\lambda} - l_1 = l_4 e^{-j\psi}. \tag{6.3}$$

Multiplying both sides of Eqs. (6.2) and (6.3) to eliminate ψ and then multiplying both sides of the resulting equation with $e^{j\lambda}$, we obtain, after some algebra:

$$A(\phi)e^{2j\lambda} + B(\phi)e^{j\lambda} + C(\phi) = 0 \tag{6.4}$$

where

$$A(\phi) = l_3(l_2 e^{-j\phi} - l_1), \tag{6.5}$$

$$B(\phi) = l_1^2 + l_2^2 + l_3^2 - l_4^2 - 2l_1 l_2 \cos\phi, \tag{6.6}$$

$$C(\phi) = l_3(l_2 e^{j\phi} - l_1). \tag{6.7}$$

The solution to the quadric equation (6.4) can be obtained as

$$e^{j\lambda} = \frac{-B(\phi) \pm \sqrt{\Delta_1(\phi)\Delta_2(\phi)}}{2A(\phi)}, \tag{6.8}$$

where

$$\Delta_1(\phi) = l_1^2 + l_2^2 + (l_3 + l_4)^2 - 2l_1 l_2 \cos\phi, \tag{6.9}$$

$$\Delta_2(\phi) = l_1^2 + l_2^2 + (l_3 - l_4)^2 - 2l_1 l_2 \cos\phi. \tag{6.10}$$

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

Eq.(6.8) yields a meaningful solution for λ when the input angle ϕ satisfies the following well-known feasibility condition:

$$\Delta_1(\phi)\Delta_2(\phi) \leq 0 \tag{6.11}$$

If the above inequality is true for all ϕ in the range $[0, 2\pi]$, then the input link is a crank; otherwise, it is a rocker.

From a purely mathematical point of view, $e^{j\lambda}$ as defined by (6.8) is always a periodic function if we disregard the constraint (6.11) so that ϕ can take the full range of $[0, 2\pi]$. This allows us to obtain the Fourier series representation of $e^{j\lambda}$ as

$$e^{j\lambda} = \sum_{k=-\infty}^{\infty} C_k e^{jk\phi} \tag{6.12}$$

where C_k are the proper Fourier coefficients and they depend not on the absolute size of the linkage but only on their relative ratios: $\frac{l_2}{l_1}, \frac{l_3}{l_1}, \frac{l_4}{l_1}$. For the sake of efficiency, only a small group of C_k associated with low harmonics components will be used for approximating $e^{j\lambda}$. When the input link is not a crank, or the path to be approximated is only a portion of the complete coupler curve, the corresponding input angle ϕ rocks in a range that is less than 2π. In this case, better approximation can be achieved if C_k is obtained from the least squares method for open curves as discussed in Section 3.3.

Now, we seek to obtain the Fourier representation of the coupler curve of the four bar mechanism. Let $\mathbf{A}_0 = x_0 + jy_0$ be the complex number specifying the fixed pivot A_0 and let $\mathbf{z} = re^{j\alpha}$ represent the position of P with respect to the coupler link AB. The position of the coupler point P can be represented as

$$\mathbf{P} = \mathbf{A}_0 + l_2 e^{j\theta_2} + \mathbf{z}e^{j\theta_3} = \mathbf{A}_0 + l_2 e^{j\theta_1} e^{j\phi} + \mathbf{z}e^{j\theta_1} e^{j\lambda}. \tag{6.13}$$

Substituting Eq. (6.12) into (6.13) and in view of Eq. (6.1), we obtain

$$\mathbf{P} = \sum_{k=-\infty}^{\infty} P_k e^{jk\omega t}, \tag{6.14}$$

where

$$\begin{cases} P_0 = \mathbf{z}e^{j\theta_1} C_0 + \mathbf{A}_0 \\ P_1 = \mathbf{z}e^{j\theta_1} C_1 e^{j\phi_0} + l_2 e^{j(\theta_1+\phi_0)} \\ P_k = \mathbf{z}e^{j\theta_1} C_k e^{jk\phi_0}|_{k\neq 0,1} \end{cases} \tag{6.15}$$

By inspecting Eq.(6.15), it can be concluded that $\mathbf{A}_0$, l_2 and $\mathbf{z}e^{i\theta_1}$ contribute to the harmonic components of the coupler curve $\mathbf{P}$ in different ways. First, $\mathbf{A}_0$ defines the position of the coupler curve. The variation of the position of $\mathbf{A}_0 = x_0 + jy_0$ results in a translation of the entire coupler curve. Secondly, l_2 is associated with the first harmonic term. The complex number term $\mathbf{z}e^{j\theta_1} = re^{j(\alpha+\theta)}$ acts as a constant multiple of the coefficients, C_k, associated with the loop closure equation as given by (6.12).

6.2 Fourier Based Synthesis Method

The interest of synthesize a crank-rocker linkage remains in finding the ten variables $\{l_1, l_2, l_3, l_4, x_0, y_0, \theta_1, r, \alpha, \phi_0\}$ to satisfy Eq (6.15). In this section, we seek to match or approximate as closely as possible a task trajectory with the coupler curve of a four-bar linkage. The task curve is time-prescribed, i.e., it is defined as a function of the input angle, $\phi - \phi_0 = \omega t$, with ϕ_0 as its initial angle. The task curves can be either closed or open. We approximate both the given task curve and the coupler curve using truncated Fourier series in a least squares formulation. The objective is to find the optimal values of the relevant parameters such that the curve

and the path match closely in the spectrum domain and at the same time consider the loop closure constraint. This Fourier representation allows the decoupling of the design variables and thus greatly reduces the dimensions of the search space.

6.2.1 Decoupling of Design Variables

Because both the task curve and the coupler curve are parameterized in term of the input angle, they share the same fundamental frequency and can be approximated with the same group of basis functions $\{e^{-jp\omega t}, \cdots, e^{jp\omega t}\}$. The finite Fourier series that approximates the task curve **T** is given by

$$\mathbf{T} \approx \sum_{k=-p}^{+p} T_k e^{jk\omega t} \tag{6.16}$$

For either the closed or the open task curve, T_k can be computed by the least squares curve fitting procedure as outlined from Eq.(3.10) to (3.20). In addition, the coupler path **P** is also approximated by a finite Fourier series:

$$\mathbf{P} \approx \sum_{k=-p}^{+p} P_k e^{jk\omega t}, \tag{6.17}$$

where P_k are given by Eq. (6.15).

The task curve **T** and the coupler curve **P** matches perfectly if

$$T_k = P_k \tag{6.18}$$

for all $k \in [-p, p]$. In view of Eq. (6.15), this leads to

$$\text{for } k = 0, \qquad T_0 = C_0 r e^{j(\alpha+\theta_1)} + x_0 + jy_0, \tag{6.19}$$

$$\text{for } k = 1, \qquad T_1 = C_1 r e^{j(\alpha+\theta_1)} e^{j\phi_0} + l_2 e^{j\theta_1} e^{j\phi_0}, \tag{6.20}$$

$$\text{for } k \neq 0, 1, \qquad T_k = C_k r e^{j(\alpha+\theta_1)} e^{jk\phi_0}. \tag{6.21}$$

In general, the problem of path synthesis for a four-bar mechanism involves nine design variables $\{l_1, l_2, l_3, l_4, x_0, y_0, \theta_1, r, \alpha\}$ as well as the initial angle ϕ_0 of the input link. In this paper, we use the following set of ten parameters

$$\mathbf{S} = \{l_2, \frac{l_2}{l_1}, \frac{l_3}{l_1}, \frac{l_4}{l_1}, x_0, y_0, \theta_1, \mathbb{C}, \mathbb{S}, \phi_0\} \tag{6.22}$$

where

$$\mathbb{C} = r\cos(\alpha + \theta_1), \;\; \mathbb{S} = r\sin(\alpha + \theta_1), \tag{6.23}$$

for developing the synthesis procedure. Furthermore, we can separate the set $\mathbf{S}$ into two subsets of design variables:

$$\mathbf{S}_1 = \{\frac{l_2}{l_1}, \frac{l_3}{l_1}, \frac{l_4}{l_1}, \phi_0, \mathbb{C}, \mathbb{S}\}, \quad \mathbf{S}_2 = \{l_2, x_0, y_0, \theta_1\}. \tag{6.24}$$

This is because Eq. (6.21) includes exclusively the design variables contained in $\mathbf{S}_1$. In addition, it is straightforward to express the four design variables in $\mathbf{S}_2$ in terms of the six design variables in $\mathbf{S}_1$ by solving Eq. (6.19) and (6.20), i.e.,

$$x_0 + iy_0 = T_0 - C_0(\mathbb{C} + i\mathbb{S}), \tag{6.25}$$

$$l_2 e^{i\theta_1} = T_1 e^{-i\phi_0} - C_1(\mathbb{C} + i\mathbb{S}). \tag{6.26}$$

Note that C_0, C_1 depend only on link ratios and the initial angle ϕ_0. Thus, the problem of path synthesis is simplified to that of seeking the optimal values for $\mathbf{S}_1$ such that Eq. (6.21) is satisfied to the extent possible for all $k \in [-p, -1] \bigcup [2, p]$.

To find a least squares solution to Eq. (6.21), we define the following error

function:

$$\begin{aligned} I &= \sum_{k\neq 0,1} |C_k r e^{i(\alpha+\theta_1+k\phi_0)} - T_k|^2 \\ &= \sum_{k\neq 0,1} [(A_k \cdot \mathbb{C} - B_k \cdot \mathbb{S} - T_k^x)^2 !' !' + (A_k \cdot \mathbb{S} + B_k \cdot \mathbb{C} - T_k^y)^2] \end{aligned} \quad (6.27)$$

where $T_k = T_k^x + iT_k^y$ and $C_k e^{ik\phi_0} = A_k + iB_k$.

The variables, $\mathbb{C}$ and $\mathbb{S}$, can be used in replacement of $\alpha + \theta_1$ and r and they appear explicitly in Eq.(6.27). To minimize the error function I, they must satisfy

$$\frac{\partial I}{\partial \mathbb{C}} = 0, \quad \frac{\partial I}{\partial \mathbb{S}} = 0, \quad (6.28)$$

which leads to

$$\mathbb{C} + j\mathbb{S} = \frac{\sum_{k\neq 0,1} T_k C_k^* e^{-jk\phi_0}}{\sum_{k\neq 0,1} |C_k|^2} \quad (6.29)$$

where C_k^* is the conjugate of C_k and depends only on the ratios $l_2/l_1, l_3/l_1, l_4/l_1$ and the initial angle ϕ_0. Thus the least squares solution to Eq. (6.21) may be obtained by minimizing the error function I involving the following four design variables:

$$\overline{\mathbf{S}}_1 = \{\frac{l_2}{l_1}, \frac{l_3}{l_1}, \frac{l_4}{l_1}, \phi_0\}. \quad (6.30)$$

In summary, the use of Fourier representation enables the decoupling of design variables for the problem of path synthesis and this leads the reduction of search space from ten design variables to four design variables.

6.2.2 Feasibility Test

In section 10.1, the loop closure equation (6.2) has been reduced to (6.8), which yields $e^{i\lambda}$, as well as the feasibility condition (6.11), which indicates whether a

four-bar linkage can be assembled for a given input angle ϕ. Thus, in addition to (6.8), it is required that the feasibility condition (6.11) be satisfied during the path synthesis process. Let the given range of ωt of the target path be $\Delta\phi$ and ϕ_0 be the initial angle. It follows from (6.9), (6.10), and (6.11) that the set of four design parameters $\overline{\mathbf{S}}_1$ must satisfy

$$[1 + \frac{l_2^2}{l_1^2} + \frac{(l_3 + l_4)^2}{l_1^2} - \frac{2l_2}{l_1}\cos\phi][1 + \frac{l_2^2}{l_1^2} + \frac{(l_3 + l_4)^2}{l_1^2} - \frac{2l_2}{l_1}\cos\phi] \leq 0 \quad (6.31)$$

for all $\phi = [\phi_0, \phi_0 + \Delta\phi]$.

6.2.3 Numerical Algorithm

Now we present a numerical algorithm for path synthesis based on the aforementioned formulation using finite Fourier series. The task curve is specified as a function of the input angle $\phi - \phi_0$. As alluded to earlier, the goal is to find a set of design variables $(\frac{l_2}{l_1}, \frac{l_3}{l_1}, \frac{l_4}{l_1}, \phi_0, \mathbb{C}, \mathbb{S}, l_2, x_0, y_0, \theta_1)$ such that the error between the task curve and the coupler curve of a four-bar linkage as defined by (6.27) is minimized. The numerical algorithm outlined below is based on the reduction of the search space from ten design variables to four design variables, $\overline{\mathbf{S}}_1 = (\frac{l_2}{l_1}, \frac{l_3}{l_1}, \frac{l_4}{l_1}, \phi_0)$.

a) The task curve **T** is approximated with a user selected finite Fourier series using (6.16), which include the basis functions to the p^{th} order. Typically, for a smooth curve, a low harmonic curve is sufficient to encode its shape information, and thus p can be small [67, 68]. After that, the least squares curve fitting procedure is applied to **T** to obtain its coefficients.

b) In this paper, we carry out a numerical search for $\overline{\mathbf{S}}_1$ within a reasonable range

as described below. The range of ϕ_0 is allowed to be $[0, 2\pi]$. 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 search $\frac{l_2}{l_1}, \frac{l_3}{l_1}, \frac{l_4}{l_1}$ in $[\frac{1}{K_{max}}, K_{max}]$, and we are only interested in the mechanisms that satisfy the link conditions for link ratios:

$$\frac{l_m}{l_n} \in [\frac{1}{K_{max}}, K_{max}], \quad \text{for all } m, n \in \{1, 2, 3, 4\}. \tag{6.32}$$

c) Examine the feasibility constraint for current value of $\overline{\mathbf{S}}_1$ using (6.31). If the feasibility condition is satisfied, the subsequent steps **(d)** and **(e)** will be processed; otherwise, they will be skipped.

d) For each set of values for $\overline{\mathbf{S}}_1$, use Eq.(6.8), to obtain $e^{j\lambda}$ as a function of ϕ, and then use the least squares procedure outlined in Chapter 3 to obtain the Fourier coefficients, C_k.

e) Compute $\mathbb{C} + j\mathbb{S}$ using Eq. (6.29) and $\mathbf{S}_2$ using (6.25) and (6.26). Evaluate the error function I using (6.27) and store the value of I.

f) Search for the minimum value of the error function I. The corresponding values of the design variables $\mathbf{S}$ yield the optimal design for a four-bar linkage such that its coupler curve best approximates a given task curve $\mathbf{T}$.

The Pseudo Code

- Specify the basis functions $\{e^{-jp\omega t}, \cdots, e^{jp\omega t}\}$;
- Fit the task path $\mathbf{T}$ with a finite Fourier series;
- Specify the maximum link ratio K_{max};

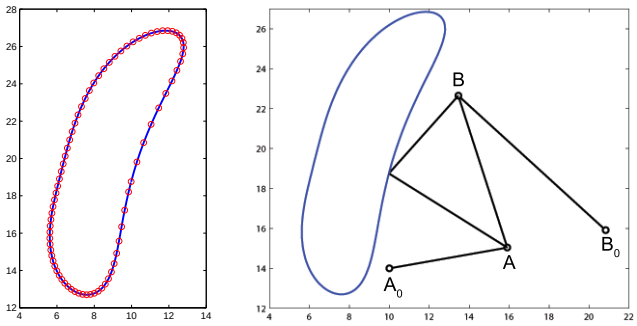
For each $\phi_0 \in [0, 2\pi]$, and each $\frac{l_2}{l_1}, \frac{l_3}{l_1}, \frac{l_4}{l_1} \in [\frac{1}{K_{max}}, K_{max}]$ **do**
If the link ratio condition is satisfied **do**
If feasibility constraint is satisfied **do**
For each $+$ or $-$ configuration **do**
• Reconstruct $e^{j\lambda}$;
• Compute the Fourier coefficients , C_k's, of $e^{j\lambda}$;
• Compute $\mathbb{C} + j\mathbb{S}$ and $\mathbf{S}_2$;
• Evaluate the error function I
If the value of I is the minimum, **do**
• Update $\mathbf{S}$ with current values;
End if
End for
End if
End if
End for
• Recover $\{l_1, l_2, l_3, l_4, x_0, y_0, \theta_1, r, \alpha, \phi_0\}$¡¡from $\mathbf{S}$;

Essential to the algorithm is the search for the optimal value of $\overline{\mathbf{S}}_1$ in a four-dimensional space. For the sake of simplicity, one may adopt the direct search method to obtain the best $\overline{\mathbf{S}}_1$. This is of course not efficient but an optimal solution can always be found. The efficiency of the algorithm will be improved if a more intelligent search method such as the simulated annealing search [110, 111] is used in replacement of the direct search method.

6.3 Examples

In this section, we present four examples for four-bar linkage synthesis using the Fourier descriptor based algorithm. In the first two examples, the given paths are coupler curves generated from known four-bar linkages. The goal is to verify the validity of the algorithm presented in this paper. In the last two examples, the paths are defined using finite Fourier series whose coefficients are selected without

connection to a four-bar linkage. All length units are in inches. All these examples are implemented in MATLAB on a Dell laptop running Window XP with Intel Core Duo CPU T2250 at 1.73GHz and 1 GB of RAM.



(a) (b)

Figure 6.2: Closed path generation. (a) The small red circles trace the given closed path and the solid curve in blue is the synthesized one. (b) The synthesized four-bar linkage.

6.3.1 Generation of a Closed Path

First presented is an example of synthesizing a crank-rocker mechanism for the generation of a closed path. In order to illustrate the validity of the proposed algorithm, we use the known dimensions of a crank-rocker mechanism to generate a closed coupler curve as shown in Figure 6.2(a) as a red dotted curve. We then use the generated closed curve as the input to our algorithm to see if we can recover the original link dimensions. To find a global optimum for all choices of the initial angle ϕ_0, we evenly sample ϕ_0 over $[0, 2\pi]$, apply the direct search method to obtain the best choice for $\{\frac{l_2}{l_1}, \frac{l_3}{l_1}, \frac{l_4}{l_1}\}$, and store the resulting minimum structural error. The results show excellent match as indicated in Table 6.1-6.2 and Figure 6.2. The

error I vs initial angle ϕ_0 as shown in Figure 6.3 are obtained using the simulated annealing algorithm.

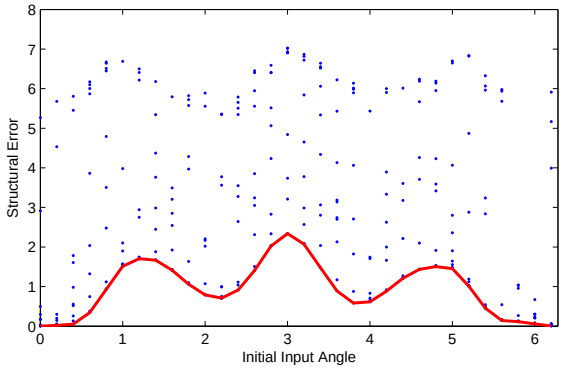


Figure 6.3: The distribution of structural errors I over initial input angle ϕ_0, computed by the simulated annealing method on $\{\frac{l_2}{l_1}, \frac{l_3}{l_1}, \frac{l_4}{l_1}\}$.

	Task Path	$\delta_1 = 0.01$	$\delta_2 = 0.02$
C_{-2}	$-0.7574 + 0.0774i$	$-0.7574 + 0.0770i$	$-0.7762 + 0.0809i$
C_{-1}	$-2.3581 + 0.9571i$	$-2.3585 + 0.9557i$	$-2.3545 + 0.9514i$
C_0	$8.5305 + 19.6441i$	$8.5296 + 19.6417i$	$8.5289 + 19.6412i$
C_1	$4.6530 - 1.3332i$	$4.6516 - 1.3359i$	$4.6507 - 1.3361i$
C_2	$0.5513 - 0.4648i$	$0.5516 - 0.4645i$	$0.5470 - 0.4667i$
I		1.5868×10^{-5}	4.0411×10^{-4}

Table 6.1: Fourier Coefficients of the given path and two synthesized coupler curves with stepsize $\delta_1 = 0.01$ and $\delta_2 = 0.02$

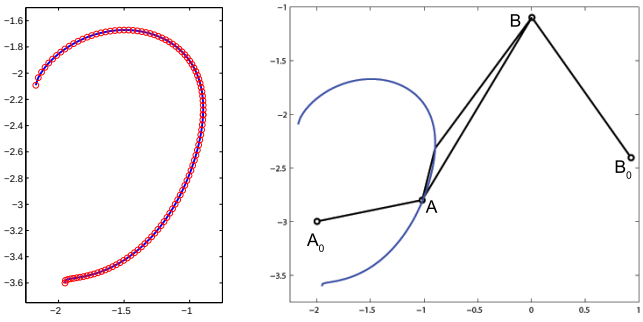
6.3.2 Generation of an Open Path

Presented next is an example of synthesizing a four-bar mechanism that follows an open path. In order to demonstrate the validity of our algorithm, the open path, see Figure 6.4(a), is generated from a known four-bar mechanism; the input angle ϕ, evenly sampled for interval 0.01 from its whole range that preserves the loop closure

	Task Path	$\delta_1 = 0.01$	$\delta_2 = 0.02$
x_0	10	10.0022	10.0334
y_0	14	13.9988	14.0131
θ_1	0.1745	0.1743	0.1802
l_1	11	10.9943	10.8390
l_2	6	5.9969	5.9716
l_3	8	7.9958	7.9621
l_4	10	9.9996	9.9527
r	7	6.9996	7.0004
α	0.6981	0.6990	0.6867
ϕ_0		0.2991	0.3991

Table 6.2: The design parameters of the given four-bar linkage and those of two synthesized for two stepsizes. All angles are in rad.

condition and the associated coupler points are used as input to test our algorithm. Again, the results indicate excellent match between the original mechanism and those that have been synthesized. Please refer to Table 6.3-6.4 and Figure 6.4(a, b) for details. Similar to the former example, the structural error I versus the initial angle ϕ_0 is plotted in Figure 6.5.



(a) (b)

Figure 6.4: Open path generation. (a) The small red circles trace the given open path and the solid curve in blue is the synthesized one. (b) The synthesized four-bar linkage.

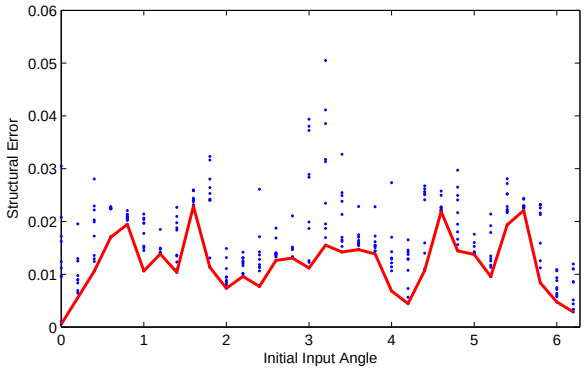


Figure 6.5: The distribution of structural errors I over initial input angle ϕ_0, computed using the simulated annealing method on $\{\frac{l_2}{l_1}, \frac{l_3}{l_1}, \frac{l_4}{l_1}\}$.

	Task Path	$\delta_1 = 0.01$	$\delta_2 = 0.02$
C_{-2}	$0.0061 + 0.0137i$	$0.0178 - 0.0118i$	$0.0211 - 0.0134i$
C_{-1}	$-0.0590 + 0.1365i$	$-0.0762 + 0.1721i$	$-0.0798 + 0.1757i$
C_0	$-1.6911 - 2.6475i$	$-1.6716 - 2.6864i$	$-1.6676 - 2.6910i$
C_1	$0.8290 + 0.2092i$	$0.8113 + 0.2432i$	$0.8082 + 0.2476i$
C_2	$0.0202 - 0.0364i$	$0.0326 - 0.0600i$	$0.0350 - 0.0631i$
I		1.2340×10^{-7}	9.6908×10^{-7}

Table 6.3: Fourier Coefficients of the given open path and two synthesized coupler curves with stepsize $\delta_1 = 0.01$ and $\delta_2 = 0.02$

6.3.3 Generation of Low Harmonic Curve Segments

Now consider the problem of synthesizing a four-bar linkage for generating an arbitrarily selected open path using the proposed algorithm that includes the use of the simulated annealing algorithm instead of direct search on $\{\frac{l_2}{l_1}, \frac{l_3}{l_1}, \frac{l_4}{l_1}\}$.

Simulated annealing is a probabilistic search algorithm for global optimization. Instead of the best solution, it seeks an acceptable approximation to the optimal result with satisfactory efficiency. The results inherit a degree of randomness. This fact is demonstrated by the following two examples. The first one, as shown in

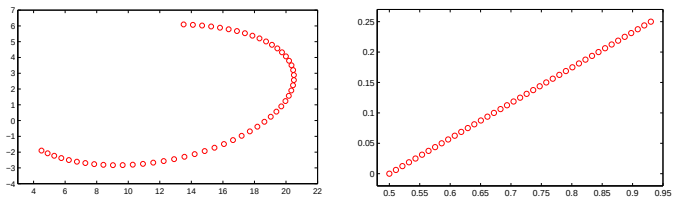
	Task Path	$\delta_1 = 0.01$	$\delta_2 = 0.02$
x_0	-2	-1.9982	-1.9977
y_0	-3	-2.9971	-2.9990
θ_1	0.2	0.1997	0.2015
l_1	3	2.9856	2.9581
l_2	1	0.9998	1.0006
l_3	2	1.9853	1.9868
l_4	1.6	1.6011	1.5705
r	0.5	0.4970	0.4976
α	0.3	0.2963	0.3042
ϕ_0		0	0.0030

Table 6.4: The design parameters of the given four-bar linkage and those of two synthesized for two stepsizes. All angles are in rad.

Figure 6.6(a), is a 2-harmonic curve segment given by the equation below:

$$\begin{aligned} \mathbf{T} = \; & (10.0 + 2.0i) + (2.0 + 1.4i)e^{-i\omega t} + (6.0 - 3.0i)e^{i\omega t} \\ & + (0.2 + 0.4i)e^{-2i\omega t} + (0.9 - 0.5i)e^{2i\omega t} \end{aligned} \tag{6.33}$$

where ωt varies in the range $[-0.5\pi, 0.5\pi]$. We use the simulated annealing algorithm to search on $\{\frac{l_2}{l_1}, \frac{l_3}{l_1}, \frac{l_4}{l_1}, \phi_0\}$ with the maximum link ratio being 5. We obtain satisfactory results when the simulated annealing algorithm runs up to 20 times, and the average running time for each search is 24.9375 second. Figure 6.7 shows several design instances of small structural error based on our search.



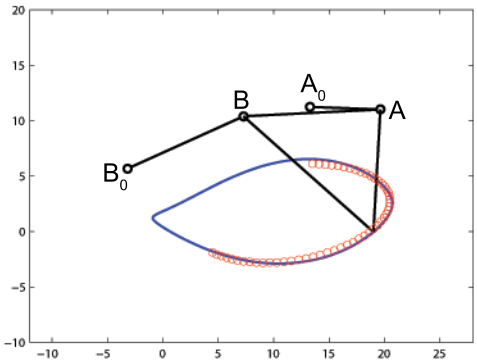
(a) (b)

Figure 6.6: (a) A 2-harmonic curve segment. (b) A low-harmonic straight-line path.

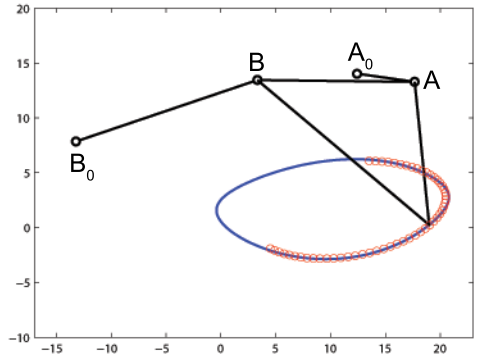
The second one, as shown in Figure 6.6(b), is a straight-line path from $(0.5, 0)$ to $(0.93, 0.25)$ and the corresponding input angle ϕ varies from ϕ_0 to $\phi_0 + 0.75\pi$. Simulated annealing search is also used with the maximum link ratio $K_{max} = 5$. The straight-line path is approximated by a 3-harmonic curve, a 4-harmonic curve, and a 5-harmonic curve, respectively, and the resulting three optimal mechanisms as illustrated in Figure 6.8-(a), (b) and (c), respectively.

6.4 Conclusions

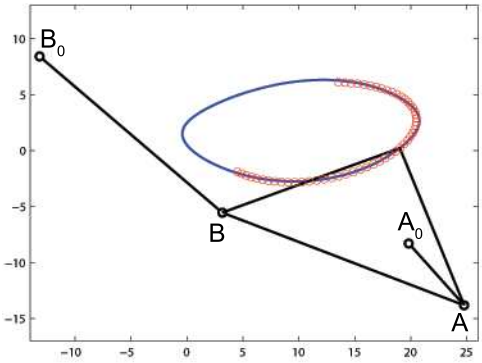
In this chapter, we have extended the Fourier descriptor based method for kinematic approximation using a least squares formulation. This leads to a general method for the dimensional synthesis of four-bar linkage for generating not only desired closed curves but also desired open curve segments. The method applies not only to Grashof linkages but also non-Grashof linkages. In addition, the method can be used for approximating a curve segment using only a portion of the coupler curve instead of the complete coupler curve as required in the original Fourier based method. The proposed algorithm can be used for generating mechanically realizable constraints in a constraint based paradigm for task centered mechanism design.



(a)

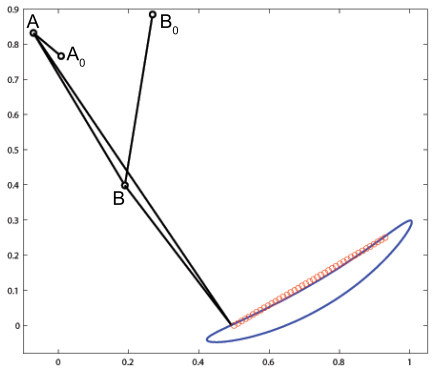


(b)

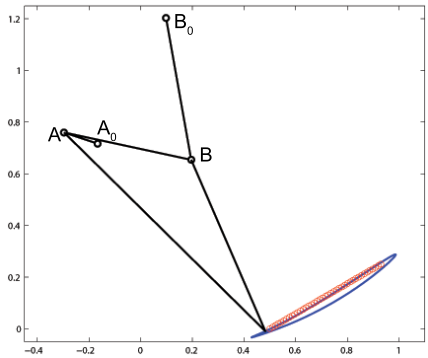


(c)

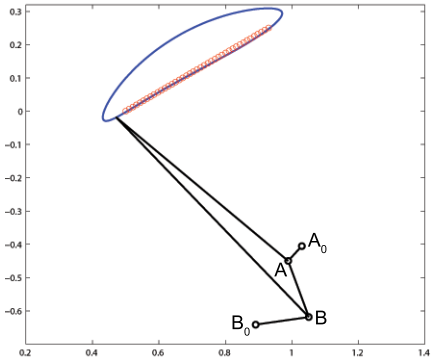
Figure 6.7: The synthesized mechanisms for the generation of the low-harmonic curve segment: The red bubbles represent the given path, and the solid curves in blue are the synthesized curves. The highest frequency employed in approximation is 2ω. The simulated annealing algorithm leads to multiple solutions with various structural errors.
(a): The structural error I is 0.0677. The running time is 22.0938 sec.
(b): The structural error I is 0.0049. The running time is 21.8281 sec.
(c): The structural error I is 0.0016. The running time is 28.5000 sec.



(a)



(b)



(c)

Figure 6.8: The synthesized mechanisms for the generation of a low-harmonic straight-line path. The red bubbles represent the given path, and the solid curves in blue are the synthesized curves.
(a): The highest frequency used for approximation is 3ω, the structural error I is 0.0017, and the running time is 52.7813 sec.
(b): The highest frequency used for approximation is 4ω, the structural error I is 0.0016, and the running time is 58.2656 sec.
(c): The highest frequency used for approximation is 5ω, the structural error I is 0.0015, and the running time is 178.75 sec.

Chapter 7

Kinematic Acquisition of Geometric Constraints for Task Centered Mechanism Design: Point Constraints

Since the early days of Industrial Revolution, machine theorists and kinematicians have sought to develop a theory to analyze and synthesize mechanisms so that engineers could approach the problem of mechanism design in a rational way. Franz Reuleaux, the author of " Theoretische Kinematik" [1], laid the foundation for modern kinematics by determining the basic mechanical building blocks and developing a system for classifying mechanism types. There is an excellent digital library at Cornell University that documents the work done by Reuleaux called Kinematic Models for Design Digital Library. Another giant in mechanism classification is Ivan Artobolevsky who published in 1975 an encyclopedia of known mechanisms titled "Mechanisms in Modern Engineering Design" [2]. More recent comprehensive treatment of kinematics in English language include Hunt [112], Phillips [4], Bottema and Roth [113], Erdman and Sandor [114], McCarthy [8, 9]. Erdman and Sadler [10] summarizes the development of modern kinematics in the forty years

preceding 1992.

Much of the existing theories for mechanism design follow the paradigm of solving the problem in two major steps, type synthesis and 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. As type synthesis constitutes the genesis of innovation and creativity in the design of mechanism-based products and systems, it is the most critical part of the mechanism design process. A recent review of the existing methods for type synthesis of can be found in Mruthyunjaya [21]. Inspite of all the advances in this area, type synthesis remains to be the most elusive part of the mechanism design process.

The two-step based paradigm for mechanism design, which has been around for over a century, is derived from the classical viewpoint that a kinematic mechanism is a collection of kinematic links connected with kinematic pairs (or joints). Links, joints and the pattern of their interconnections are viewed as a natural descriptor for the characteristics of a mechanism and provide an effective means for mechanism classification and enumeration. However, these basic mechanical elements are in general not natural descriptor for the characteristics of the output motion of a mechanism. Detailed kinematic analysis is in general required in order to understand the motion or functional characteristics of a mechanism. Since the important

work by Freudenstein and Maki [35], there have been numerous attempt to separate structural and functional (or task) considerations in mechanism design. For example, Chiou and Kota [36] identified a finite set of kinematic building blocks that provide simple kinematic functions and devised a matrix representation scheme for automatic generation of conceptual designs. Yan [37] presented a comprehensive approach to creative design of mechanisms. Inspite of all the advances in this area, including the application of artificial intelligence (Hoeltzel and Chieng [38], Yang et al. [39]), this "mechanism-centric" approach to mechanism design makes type synthesis a very challenging task, even for those who have been well trained with mechanism theory.

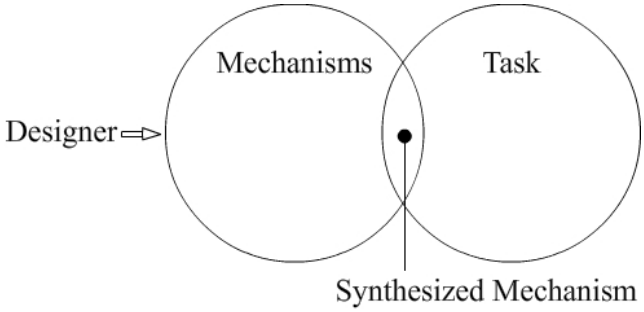
This chapter advocates a task-centered approach to mechanism design using a constraint based paradigm (Figures 7.1). In contrast to the prevailing mechanism-centric approach, a designer would focus on the *analysis* of point (or line) trajectories associated with a given motion task. The goal of the trajectory analysis is obtain a trajectory that can be construed as a geometric condition or constraint that best constrains the given motion. Typically this is done in a constraint identification and acquisition process, i.e., by comparing various trajectories of a specified motion with known constraints from a library of mechanically realizable constraints. The resulting constraints are feasible constraints that best approximate the given motion in some measure, say, in a least squares sense.

A motion task can be given in various ways. It can be defined parametrically or discretely in terms of an ordered sequence of displacements or in geometric means. The resulting feasible constraints can be used directly for the simultaneous type and

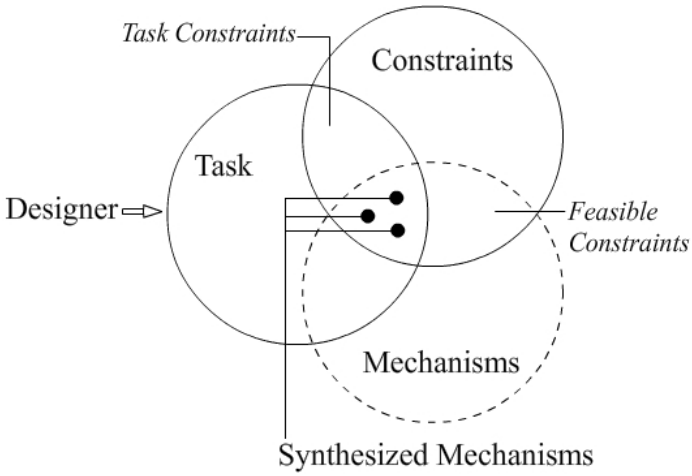
dimensional synthesis of a physical device such as mechanical linkage that generates the specified motion task. This effectively reduces the problem of mechanism synthesis to that of constraint identification and acquisition and thus bridges the gap between type and dimensional synthesis. Furthermore, as a mechanism is defined by a combination of geometric constraints, this constraint based approach reduces the complexity in type synthesis significantly. Only combinations of the feasible constraints are considered for type synthesis.

The focus of this chapter is on the introducing the framework of this constraint based approach and proposes one simple solution to the constraint identification and acquisition problem by directly comparing corresponding points associated with a point trajectory and a given constraint. More sophisticated and indirect methods may be developed for the solution to the problem. In the field of *Computational Shape Analysis* [52], it is routine to process and simplify shapes before comparisons are made. The simplified representation of shapes is called *shape descriptor* or *shape signature*. The use of shape descriptors often simplifies the process of not only shape comparison but also shape storage and classification. We will explore these methods in the future for better solutions to the problem of constraint identification.

The organization of the chapter is as follows. Section 7.1 provides a review of kinematics of planar motion from the viewpoint of motion task specification and recast representations of planar motions in explicit and implicit forms. Section 7.2 discusses how effectively a motion task is captured by an explicit or implicit form of a motion and highlights the need for conversion from one motion form to another.



(a) **Mechanism-centric design process**.



(b) **Task-centered design process**.

Figure 7.1: Mechanism-centric vs. task-centered design.

Section 7.3 deals with how to represent a geometric constraint so that it is invariant with respect to coordinate transformation and scaling. Section 9.5 provides a list of simple geometric constraints that can be readily realized mechanically and how they may be used for task-centered mechanism design. Section 7.5 presents a least-square method for identifying a geometric constraint from an explicitly given motion. Section 7.6 presents several examples to illustrate the effectiveness of the advocated approach.

7.1 Explicit and Implicit Forms of Motion

Consider the planar displacement of a rigid body as shown in Figure 7.2. Attach a moving frame **M** to the moving body with $\mathbf{d} = (d_x, d_y)$ being its origin; attach a fixed frame **F** to the fixed plane with **O** being the origin of the fixed frame. Let α denote the angle of rotation of **M** relative to **F**. Instead of Cartesian coordinates (v_x, v_y), homogeneous coordinates (v_1, v_2, v_3) are used to represent a moving point **v**, where $v_x = v_1/v_3$ and $v_y = v_2/v_3$. The homogeneous coordinates for the same point, when measured with respect to **F**, are denoted by (V_1, V_2, V_3). The corresponding Cartesian coordinates are (V_x, V_y), where $V_x = V_1/V_3$ and $V_y = V_2/V_3$. Note that we use the boldface letter **v** (or **V**) to represent a point as well as its vector representations in terms of Cartesian or homogeneous coordinates. In this way, a planar displacement can be represented by homogeneous transform:

$$\mathbf{V} = [H(d_x, d_y, \alpha)]\mathbf{v} = \begin{bmatrix} \cos\alpha & -\sin\alpha & d_x \\ \sin\alpha & \cos\alpha & d_y \\ 0 & 0 & 1 \end{bmatrix} \mathbf{v} \tag{7.1}$$

where $\mathbf{V} = (V_1, V_2, V_3)$ and $\mathbf{v} = (v_1, v_2, v_3)$.

A planar displacement as described by three parameters (d_x, d_y, α) has three degrees of freedom. When these parameters are given as explicit functions of a single parameter t, such as the time, we obtain a continuous set of planar displacements belonging to a *parametrically defined* planar motion of one degree of freedom; when these parameters are given discretely at a knot sequence $t = [t_0, t_1, \ldots, t_{n-1}]$ (with $t_{i-1} < t_i$ and $i = 1, \ldots, n-1$), we obtain a discrete set of planar displacements that can be used to define a planar motion discretely. In both cases, we consider the motion is *explicitly defined*, as one can easily generate discrete displacements from

a parametric motion or interpolate/approximate a discrete set of displacements to obtain a parametric motion that best approximates the given set of displacements.

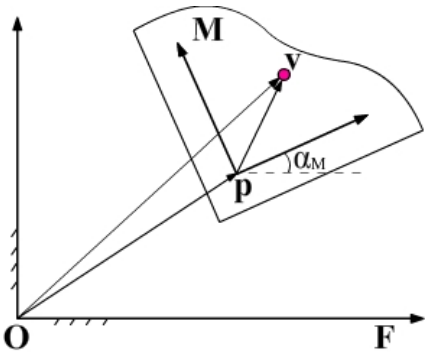


Figure 7.2: A planar displacement.

Alternatively, a planar motion can be defined in geometric means, which results in a *geometrically defined motion*. A planar motion is specified if all planar displacements satisfy two geometric conditions called *geometric constraints*. As points and lines are the basic elements of plane geometry [**?**], one can impose the geometric constraints on a planar motion in terms of geometric constraints on a pair of points, or a pair of lines, or a combination of a point and a line. When a specific point $\mathbf{v}$ of the moving plane $\mathbf{M}$ is required to stay on a given curve, $f(\mathbf{V}) = 0$, of $\mathbf{F}$, a point constraint, $f([H(d_x, d_y, \alpha)]\mathbf{v}) = 0$, is imposed. The imposition of one such constraint removes one degree of freedom and results in a two-degrees-of-freedom motion; the imposition of two such constraints removes two degrees of freedom and results in a one-degree-of-freedom motion. In this paper, we consider such a geometrically defined motion as the *implicit representation* of a motion. For exam-

ple, when $f(\mathbf{V}) = 0$ defines an algebraic curve, the resulting motion is an algebraic motion, for any of its points traces out an algebraic curve [113]. In the special case when two of its moving points are required to stay on two separate circles, the moving plane **M** follows the motion of the coupler link of a four-bar linkage.

7.2 What Captures a Motion Task?

In specifying a motion task, it is common to use either a discretely specified motion such as finite displacements or a continuous motion in parametric form such as a rational motion in Bézier or B-spline form [69, 115, 71]. Figure 7.3 shows an example of a discrete set of planar displacements that are given explicitly. Thus from the perspective of the end users, the explicit form of a motion captures the task requirements.

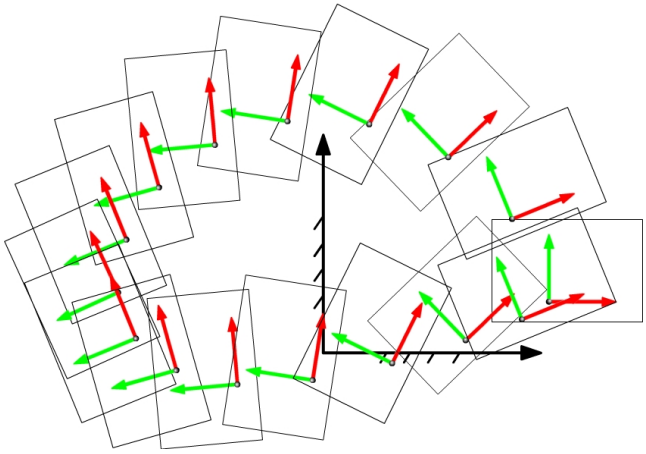


Figure 7.3: A discrete set of planar displacements.

The goal of mechanism synthesis is to come up with a mechanism type as well as the associated dimensions so that the output motion of the end-effector of the

mechanism best approximates the given set of displacements. The main function of the mechanism is to impose appropriate geometric constraints to the end-effector so that it will follow the explicitly given motion path. Thus, from the perspective of the mechanism designer, it is the constraints associated with the motion that is supposed to capture a specific motion task. Existing works on mechanism synthesis, however, focus mostly on the mechanisms themselves, either on the classification and enumeration of various types of mechanisms or on the kinematic geometry of known types of mechanisms. It is this mechanism-centric approach that makes the process of type synthesis very challenging and often elusive, even for those who are well trained in mechanism theory.

In this chapter we propose a new paradigm for mechanism synthesis that is centered on the motion task itself. Instead of trying to search and match directly a mechanism to a given motion task, we seek to uncover the hidden geometric constraints associated with a specified motion task and then use the resulting geometric constraints as a basis for mechanism synthesis. Central to this paper is the development of a general method for converting motions from their explicit form to the implicit form by recognizing and identifying the underlying geometric constraints. This problem is here referred to as *kinematic acquisition of geometric constraints*:

Given: a motion in explicit form

Find: a set of "simple geometric constraints" that best fit the given motion.

Here in this paper, "simple geometric constraints" are defined as those that can be readily generated by simple kinematic chains. In general, the notion of "simple

constraints" can be defined in the context of specific applications to be considered, with or without regard to those that can be mechanically realized.

The inverse to the constraint acquisition problem is to find an explicit representation of a motion from a set of constraints that a motion has to satisfy. When the constraints are directly generated by a mechanism, this reduces to the classical problem of mechanism analysis.

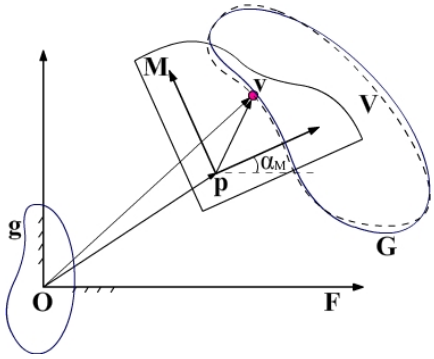


Figure 7.4: The solid curve in blue is the standard geometric constraint **g**, the solid curve in blue **G** is transformed from **g**, and the dash curve in black **V** is the trajectory of moving point **v**.

7.3 Description of Geometric Constraints

In this paper, we restrict ourselves to point-geometric constraints, i.e., a specific point on the moving body is required to stay on a given curve. In order to focus on the shape of the curve, we seek to develop a representation of a curve that is invariant with respect to rigid-body transformation in the plane as well as scaling.

A linear transformation in the plane that preserves the shape but not the scale

of a curve is called an *equiform displacement* [113]. Let $\mathbf{g} = (g_x, g_y, 1)$ and $\mathbf{G} = (G_x, G_y, 1)$ denote a specified curve and its new configuration after an equiform displacement (in Figure 7.2):

$$\mathbf{G} = [E]\mathbf{g} = \begin{bmatrix} a & -b & c_x \\ b & a & c_y \\ 0 & 0 & 1 \end{bmatrix} \mathbf{g} \tag{7.2}$$

where $a = \lambda \cos\xi$ and $b = \lambda \sin\xi$ with ξ being the angle of rotation and λ being the scaling factor. Thus, an equiform displacement has four degrees of freedom and are defined by the parameters (a, b, c_x, c_y).

The main task in kinematic acquisition is to identify a point $\mathbf{v}$ in the moving body such that its trajectory $\mathbf{V}$ best approximates one of the constraint curves in a library of "standard constraint curves". Another factor that affects this constraint matching process is the parameterizations of the constraint curve and the trajectory $\mathbf{V}$. Ideally, an arc-length parametrization for each of the curve can be carried out to remove the influence of parametrization in the shape matching process. In this paper, we take a computationally less expensive approach of normalizing the parameterizations of these closed curves so that they are in the range of $[0, 1]$. Since the periods of these curves are made equal to one, we have

$$\mathbf{g}(t) = \mathbf{g}(t-1), \quad \mathbf{G}(t) = \mathbf{G}(t-1), \quad \mathbf{V}(t) = \mathbf{V}(t-1) \tag{7.3}$$

where $t \in [0, 1]$. Another factor that affects the constraint matching process is the choice of initial point on each of these curves. To address this issue, we provide a shift factor t_Δ so that the initial point on the constraint curve can vary in order to find a better match. Incorporating the shift factor, we can define the normalized

parameters, t_G and t_V, for the constraint curve $\mathbf{G}(t_G)$ and a point trajectory of the given motion, $\mathbf{V}(t_V)$ as

$$t_V = t, \qquad t_G = t + t_\Delta \tag{7.4}$$

wherein $t \in [0, 1]$ and the shift parameter t_Δ indicates how much the motion-shape correspondence has to be shifted. $\mathbf{V}(t_V)$ on $[0, 1]$ match $\mathbf{G}(t_G)$ on $[t_\Delta, 1 + t_\Delta]$.

7.4 Simple Geometric Constraints

This section presents a set of simple geometric constraints that can be readily provided using simple mechanisms. As mentioned in the proceeding section, all parameters are normalized to the range $[0, 1]$. All constraints are represented by closed curves and special attention has been paid to their orientations as they play a significant role in shape matching process. Specifically, the same set of points are viewed as two distinct constraints if they are traced in two opposite orientations, clockwise and counterclockwise.

A circular constraint is represented by a unit circle centered at the origin of the coordinate system and with the initial point locating at $(1, 0)$, Figure 7.5.

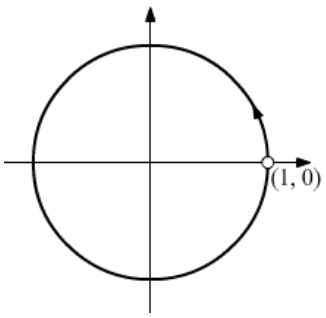


Figure 7.5: The circular constraint traced in counterclockwise direction.

A circular-arc constraint is a segment of the circumference of a unit circle. It starts from point $(1, 0)$ and extends counterclockwise, so that the central angle is always positive. The central angle remains a variable in the process of constraint identification. To make the circular arc closed, we represent it as a forward-and-backward plotted curve as shown in Figure 7.6.

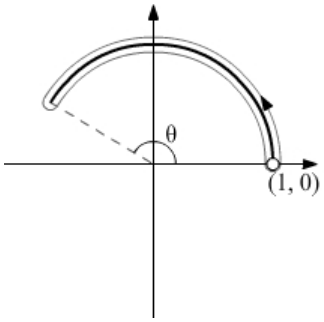


Figure 7.6: The counterclockwise traced circular arc with center angle θ.

A line segment along the x-axis starts at $(1, 0)$ and ends at $(-1, 0)$. It is also represented as double-traced closed curve, Figure 7.7.

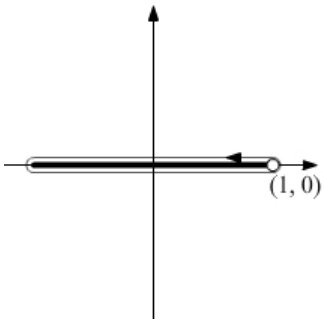


Figure 7.7: A line-segment is represented as a closed curve traced in counterclockwise direction.

An ellipse is located at $(0, 0)$. Its major axis lies on x-axis, and minor axis on

y-axis. The length of semi-major axis is 1. The length of semi-minor axis is freely adjustable and ranges from 0 to 1, Figure 9.6.

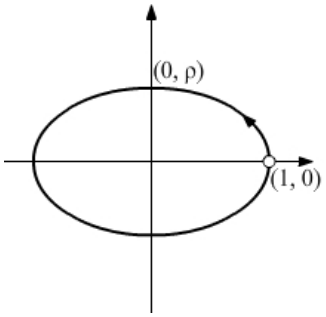


Figure 7.8: An counterclockwise traced ellipse with semi-minor axis of ρ.

In addition to the aforementioned simple curves, the coupler curve of a four-bar linkage can also be considered as defining a geometric constraint.

7.5 Least-Square Approximation for Constraint Identification

This section presents an algorithm for identifying a given constraint from point trajectories of a specified motion task. This work may be considered as an extension of an earlier work on kinematic approximation of circles and lines [116].

We assume that the motion task has been given explicitly as an ordered sequence of N discrete positions and a set of N ordered points belong to a given constraint curve has also been given. Consequently, instead of using continuous parameter t_V and t_G as given by Eq. (10.14), we use indexes i ($i = 1, \cdots, N$) to represent the sequence of points and positions, where the shift parameter t_Δ becomes an integer n, which means that the starting point on the geometric constraint point set is shifted

by n points. Thus the squared distance between the ith point on the trajectory $\mathbf{V}$ and the $(i+n)$th point on the geometric constraint is given by

$$\delta_{i,n}^2 = (\mathbf{V}_i - \mathbf{G}_{i+n})^T(\mathbf{V}_i - \mathbf{G}_{i+n}) \tag{7.5}$$

where $|\cdot|$ denotes the magnitude of a vector.

In viewof Eq. (7.1) and (7.2), we have

$$\mathbf{V}_i = [H_i]\mathbf{v}, \;\; \mathbf{G}_{i+n} = [E]\mathbf{g}_{i+n} \tag{7.6}$$

where $[H_i]$ is obtained from the 3×3 matrix in (7.1) by replacing α, d_x, d_y with $\alpha_i, d_{x,i}, d_{y,i}$.

Substituting (10.16) into (10.15), after some algebra, we obtain

$$\delta_{i,n}^2 = (\mathbf{A}_{i,n}^T\mathbf{X} - d_{x,i})^2 + (\mathbf{B}_{i,n}^T\mathbf{X} - d_{y,i})^2 \tag{7.7}$$

where

$$\begin{array}{rcl} \mathbf{A}_{i,n}^T & = & [\;\; g_{x,i+n}, \;\; -g_{y,i+n}, \;\; -\cos\alpha_i, \;\;\; \sin\alpha_i, \;\; 1, \;\; 0 \;\;] \\ \mathbf{B}_{i,n}^T & = & [\;\; g_{y,i+n}, \;\;\; g_{x,i+n}, \;\; -\sin\alpha_i, \;\; -\cos\alpha_i, \;\; 0, \;\; 1 \;\;] \end{array} \tag{7.8}$$

are determined from the specified motion and the given geometric constraint, and

$$\mathbf{X} = [a, \quad b, \quad v_x, \quad v_y, \quad c_x, \quad c_y]^T \tag{7.9}$$

are the design variables associated with the equiform displacement and choice of the point $\mathbf{v} = (x_v, y_v)$ on the moving body. The sum of squares of the deviations is thus given by (Eq. (7.7)):

$$S = \frac{1}{N}\sum_{i=1}^{N}\delta_{i,n}^2 = \frac{1}{N}\sum_{i=1}^{N}\{(\mathbf{A}_{i,n}^T\mathbf{X} - d_{x,i})^2 + (\mathbf{B}_{i,n}^T\mathbf{X} - d_{y,i})^2\}. \tag{7.10}$$

The goal for constraint acquisition is to find a point $\mathbf{v} = (v_x, v_y)$ on the moving body such that the trajectory of **v** best approximates the given constraint curve after an appropriate equiform displacement, (a, b, c_x, c_y).

The least squares solution to (10.20) can be symbolic given by the following

$$\left[\sum_{i=1}^{N}\{\mathbf{A}_{i,n}\mathbf{A}_{i,n}^T + \mathbf{B}_{i,n}\mathbf{B}_{i,n}^T\}\right]\mathbf{X} = \sum_{i=1}^{N}\{\mathbf{A}_{i,n}d_{x,i} + \mathbf{B}_{i,n}d_{y,i}\}. \tag{7.11}$$

This system of six linear equations can either be solved using the Grammer's rule or a Gaussian elimination based method such as the LU decomposition [**?**, **?**].

Alternatively, we may rewrite (7.11) as a system of three linear equations in complex numbers:

$$\left\{\sum_{i=1}^{N}\begin{bmatrix} \overline{\mathbb{G}}_{i+ni+n} & -\overline{\mathbb{G}}_{i+n}\mathbb{E}_i & \overline{\mathbb{G}}_{i+n} \\ -\overline{\mathbb{E}}_i\mathbb{G}_{i+n} & 1 & -\overline{\mathbb{E}}_i \\ \mathbb{G}_{i+n} & -\mathbb{E}_i & 1 \end{bmatrix}\right\}\begin{bmatrix} a+jb \\ v_x+jv_y \\ c_x+jc_y \end{bmatrix} = \sum_{i=1}^{N}\begin{bmatrix} \overline{\mathbb{G}}_{i+n}\mathbb{D}_i \\ -\overline{\mathbb{E}}_{i+n}\mathbb{D}_i \\ \mathbb{D}_i \end{bmatrix} \tag{7.12}$$

where $_{i+n} = g_{x,i+n} + jg_{y,i+n}$, $\mathbb{E}_i = \cos\alpha^i + j\sin\alpha^i$, $\mathbb{D}_i = d_{x,i} + jd_{y,i}$ with j being the imaginary unit. In the above equation, we have used the symbol $\overline{\mathbb{G}}$ to denote the conjugate of a complex number $\mathbb{G}$.

Up to this point, it has been assumed that the shift parameter n is known, and the least squares solution can be obtained by solving a linear system of equations. However, in general, the desired value of n is unknown, and as aforementioned, the order of the positions sampled from the motion and that of the points on the constraint curve are not automatically and optimally correlated. We now introduce a numerical algorithm for shape matching which combines least squares optimization with a direct search dealing with the shift parameter n, as stated below:

Objective: Given a discrete motion M with N sampled positions and a geometric constraint curve $\mathbf{g}$ of N sampled points, find a point on the moving body $\mathbf{v} = (v_x, v_y)$, such that after the constraint curve $\mathbf{g}$ is transformed into to a new configuration $\mathbf{G}$ via an equiform displacement, (a, b, c_x, c_y), the error between the trajectory $\mathbf{V}$ and the constraint curve $\mathbf{G}$ as defined by Eq. (10.20) is at minimum.

Algorithm

For each $n \in [1, \cdots, N]$ **do**

For each $i \in [1, \cdots, N]$ **do**

- Initialize values of $\mathbb{G}_{i+n}$, $\mathbb{R}_i$ and $\mathbb{T}_i$;
- Substitute $\mathbb{G}_{i+n}$, $\mathbb{R}_i$ and $\mathbb{T}_i$ into Eq. (7.12);

End for

- Solve Eq. (7.12) for a, b, c_x, c_y, v_x, and v_y;
- Compute the error using Eq. (10.20);

If the total error is currently at minimum **do**

- Update final solution to $\{a, b, c_x, c_y, v_x, v_y, n\}$;

End if

End for

7.6 From Geometric Constraints to Mechanisms

We now consider how to mechanically realize the geometric constraints using simple kinematic chains. Figure 7.9 enumerates some 1-dof mechanisms and classify them based on the geometric constraints they provide. For example, while a circle or a circular arc can be easily realized with a revolute joint, one can also generate a circle or a circular arc using a special four-bar linkage. A comprehensive treatise on the generation of planar algebraic curves using planar mechanisms can be found in [117].

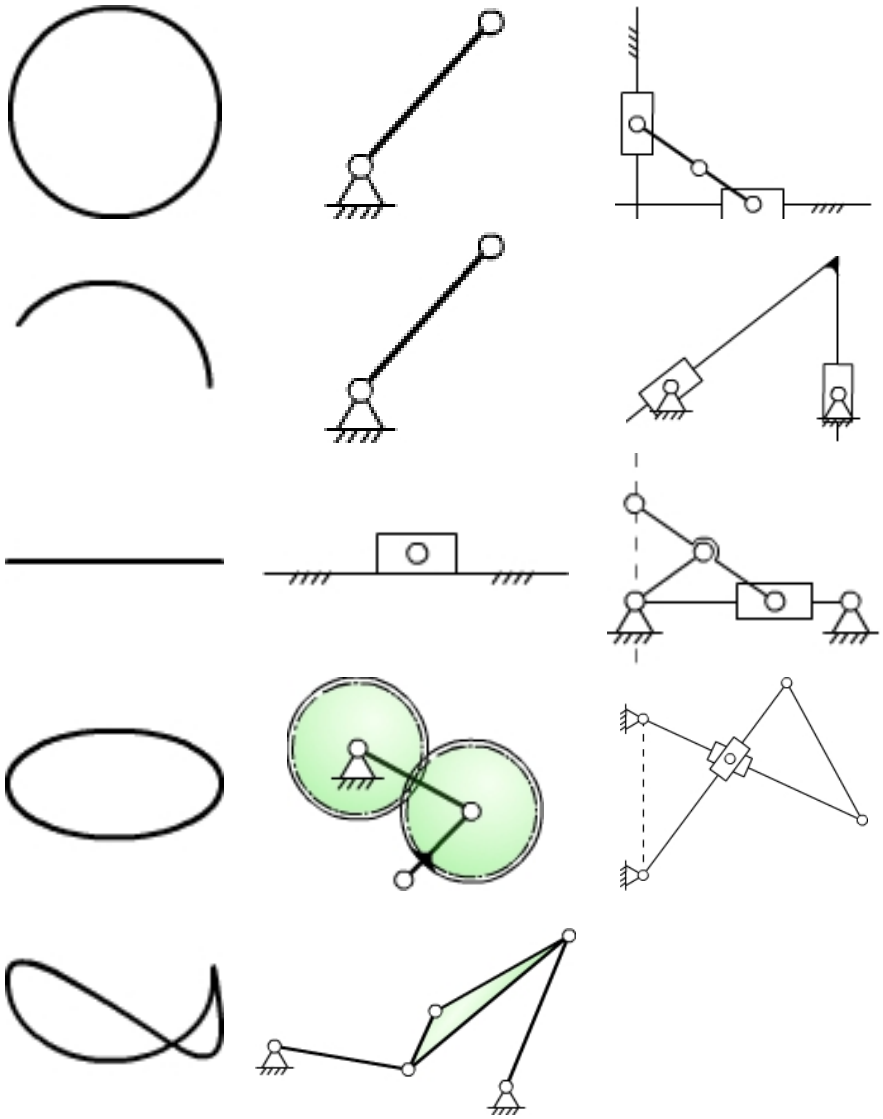


Figure 7.9: The column on the left lists the simple geometric constraints including circle, arc, line, ellipse and a four-bar coupler curve. The column in the middle lists the corresponding 1-dof mechanisms that provide the constraints. As one constraint may be generated by multiple mechanisms, the column on the right list additional mechanisms for the same constraint.

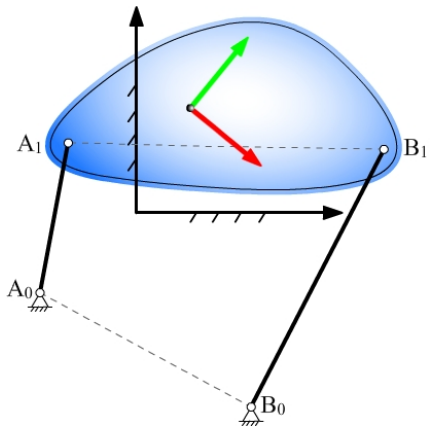


Figure 7.10: A given four-bar mechanism with $A_0B_0 = 3.8$, $A_0A_1 = 2.4$, $A_1B_1 = 5$ and $B_0B_1 = 4.6$. As crank A_0A_1 rotates for $360°$, the coupler link A_1B_1 undergoes a periodic closed motion.

In what follows we first present an example to illustrate the effectiveness of the constraint recognition scheme. We then show that the result leads naturally to task-centered design of mechanisms for the specified task. In this example, we specify 100 positions that approximate the coupler motion of a known crank-rocker mechanism as shown in Figure 7.10. The goal is to investigate if we can retrieve the original constraints associated with the crank-rocker mechanism and if we can find alternative constraints for the same motion. To this end, four geometric constraints are used for constraint recognition and retrieval: (I) a circle, (II) an arc, (III) an ellipse and (IV) the coupler curve of a four-bar motion. All these curves are parameterized in term of the input angle of the crank A_0A_1. These geometric constraints are supposed to match, approximately, some paths on the specified motion. As presented in Table 7.1 and Figure 7.11, the circular constraint (I) has the least error and thus achieves the best fit.

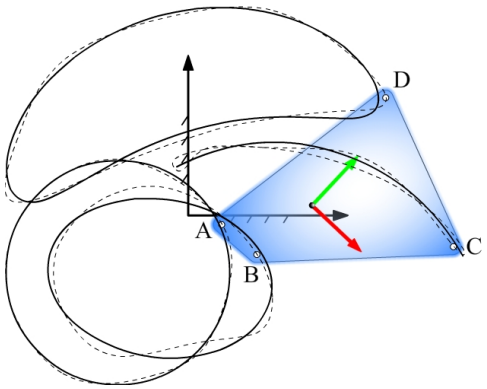


Figure 7.11: The rigid body in blue follows the motion of the coupler of a four-bar mechanism approximately. The trajectories (dash curves) of four points A, B, C and D on this rigid body are identified to optimally match the geometric constraints circle, arc, ellipse and a four-bar coupler curve (solid curves), respectively.

If all errors in Table 7.1 are within acceptable range, then any two of the four constraints from I through IV may be used to constrain the specified motion and thus lead to different mechanisms for the same specified motion. For example, a circle and a circular arc may be used to constrain the motion and this leads naturally to a crank-rocker mechanism. However, we may use one revolute-jointed link as shown in the middle column of Figure 7.9 and replace the other link with a special four-bar shown in the left column in Figure 7.9. In this case, we obtain a six-bar linkage with two sliding joints for the generation of the same set of two constraints, i.e., a circle and a circular arc. If one replaces the circle constraint with a four-bar coupler curve D and keeps the circular arc constraint, we obtain another six-bar linkage with six revolute joints for the same motion.

If a motion task is arbitrarily given, then it may not be possible to constrain it fully with simple geometric constraints as shown in Figure 7.9. To illustrate this

	I	II	III	IV
λ	2.3971	4.5837	2.4068	0.9948
$\xi(^\circ)$	-53.9353	28.1608	171.3758	0.1315
c_x	-1.5146	1.8172	-0.6827	-0.3484
c_y	-1.2352	-3.0513	-1.3393	2.5372
v_x	-1.1421	2.8075	-0.1947	-0.5480
v_y	-1.6271	1.4162	-1.5914	2.7983
n	7	91	44	0
s	0.0028	0.0201	0.0283	0.0172

Table 7.1: Results of Constraint Identification

point, we now consider a set of 16 positions shown in Figure 7.3, which are generated such that one of its moving points lies on a circle. To obtain more positions for the constraint identification scheme, we interpolate these positions with a uniform rational cubic C^2 B-spline motion[115, 71]. We then use all five constraints as shown in Figure 7.9 for constraint identification. The results are shown in Figure 7.12 and 7.13. Clearly, with the exception of the circle constraint, all other constraints are poorly matched. In this case, only the circle constraint can be used to constrain the motion. In order to obtain another constraint for the motion, a more complex kinematic chain than those listed in Figure 7.9 is needed. This task, however, is left for future research.

7.7 Conclusions

In this chapter, we have studied a new type of motion analysis problem that seeks to extract geometric constraints from a specified motion task. The resulting constraints are then used to identify and synthesize simple mechanisms for constraint genera-

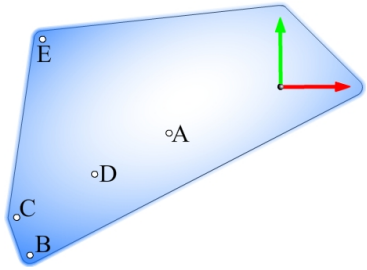


Figure 7.12: A, B, C, D and E are the five points whose trajectories are identified to match the geometric constraints.

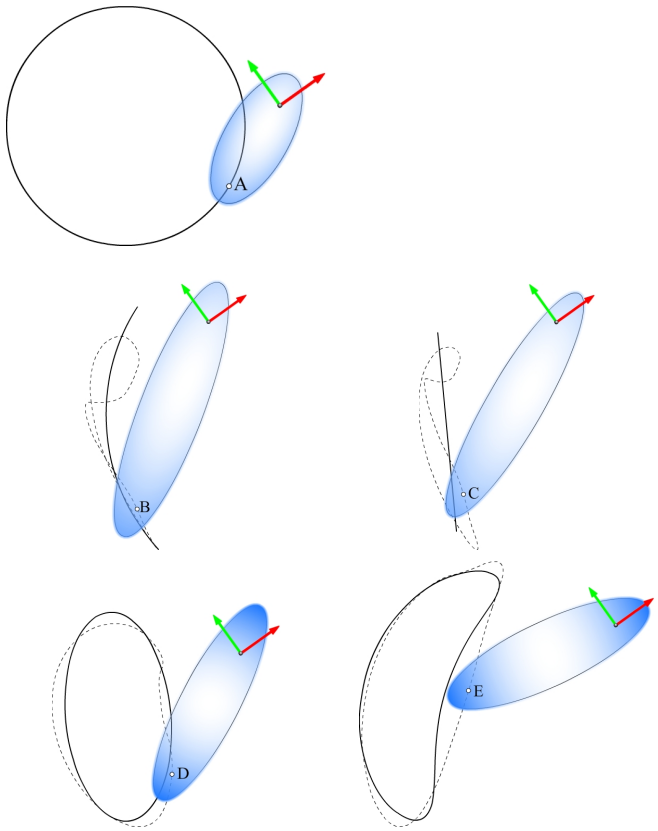


Figure 7.13: The trajectories and the geometric constraints for point A, B, C, D and E are illustrated.

tion. This task-centered and constraint based paradigm to mechanism design greatly reduces the complexity of the type synthesis problem. The current and following two chapters focus on kinematic acquisition of point-geometric constraints using a direct comparison of corresponding points in a least squares procedure. Chapter 10 will employ more sophisticated methods from computational shape analysis and extend the results to line-geometric constraints in planar mechanism design.

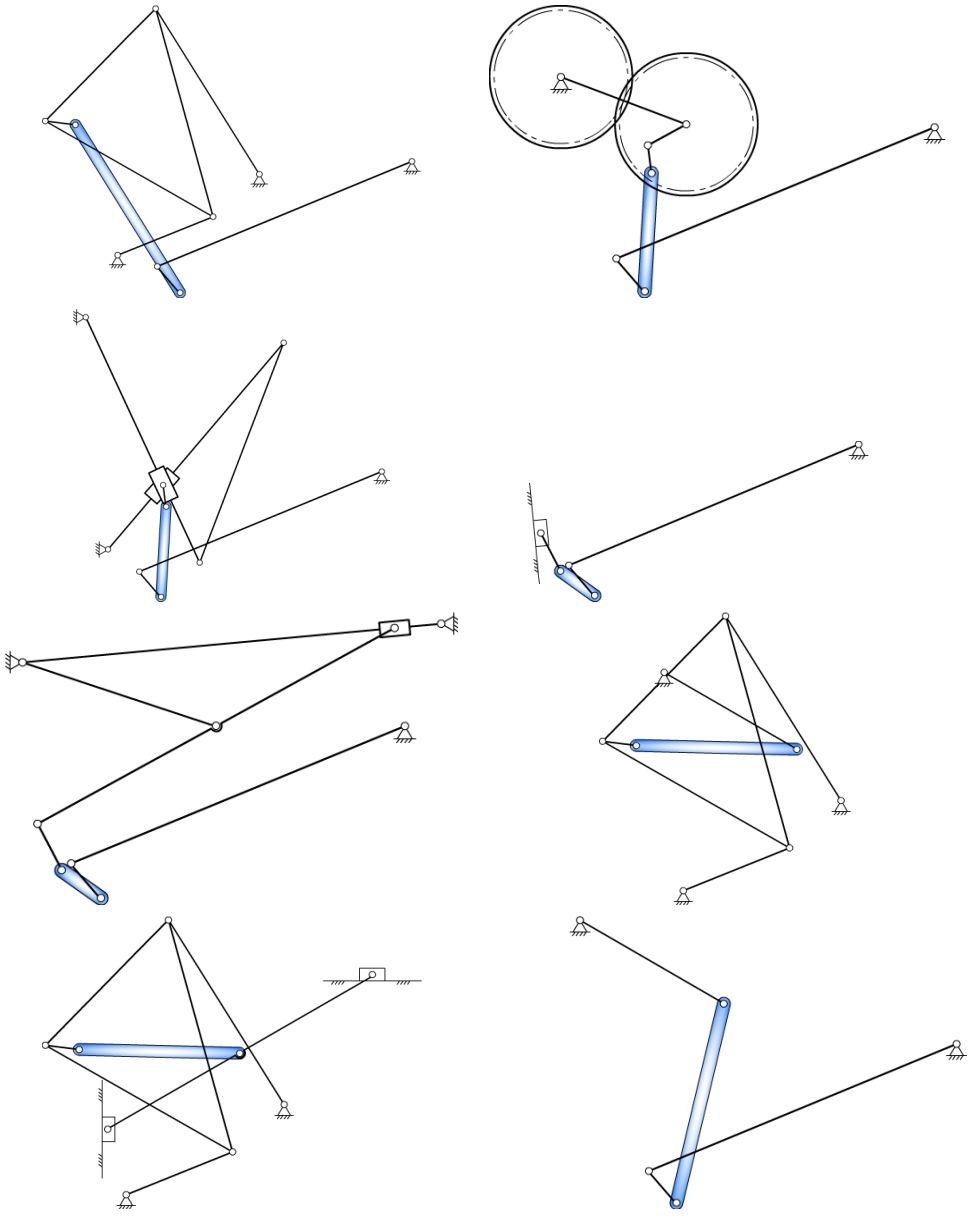


Figure 7.14: 8 synthesized mechanisms

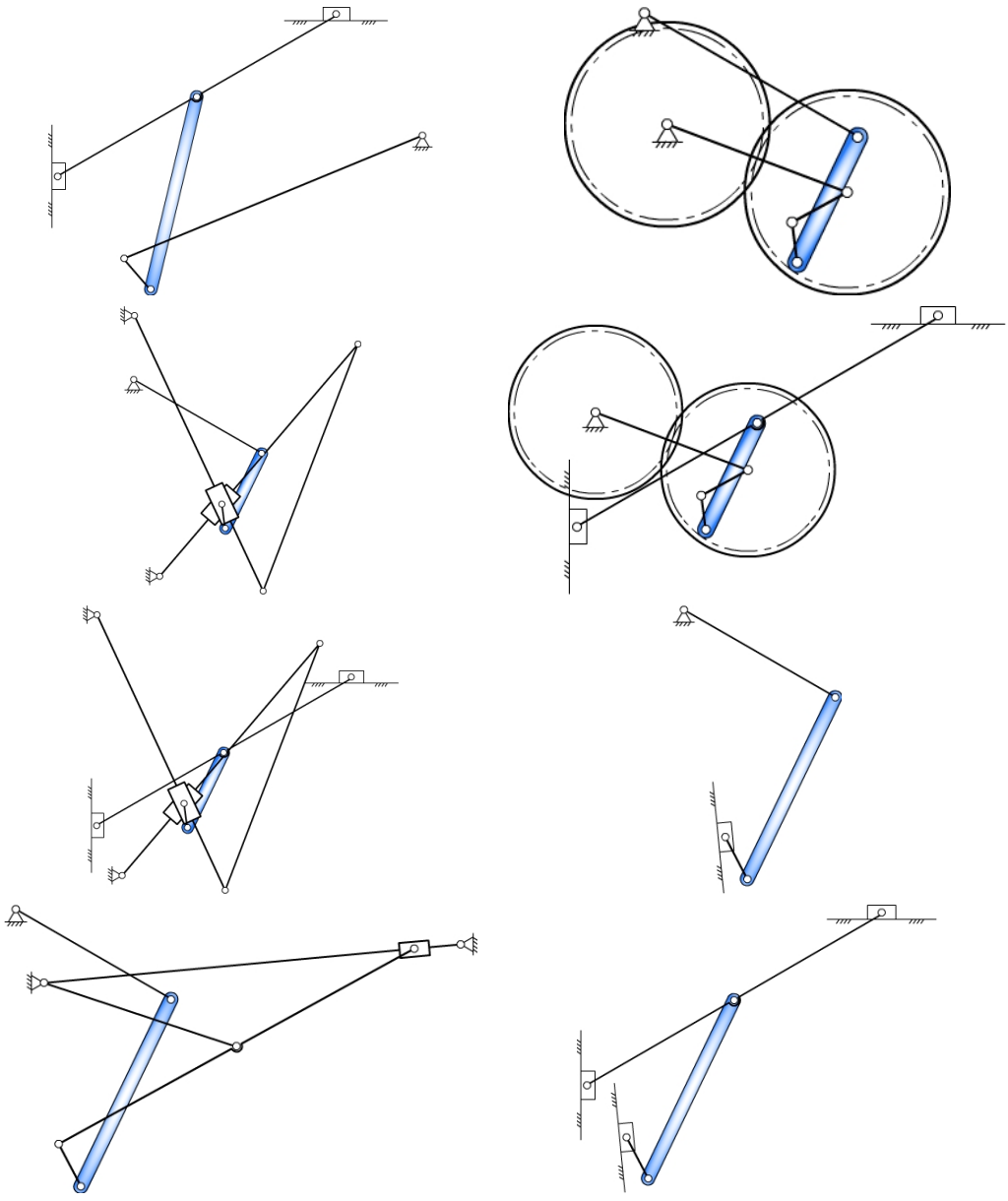


Figure 7.15: 8 synthesized mechanisms

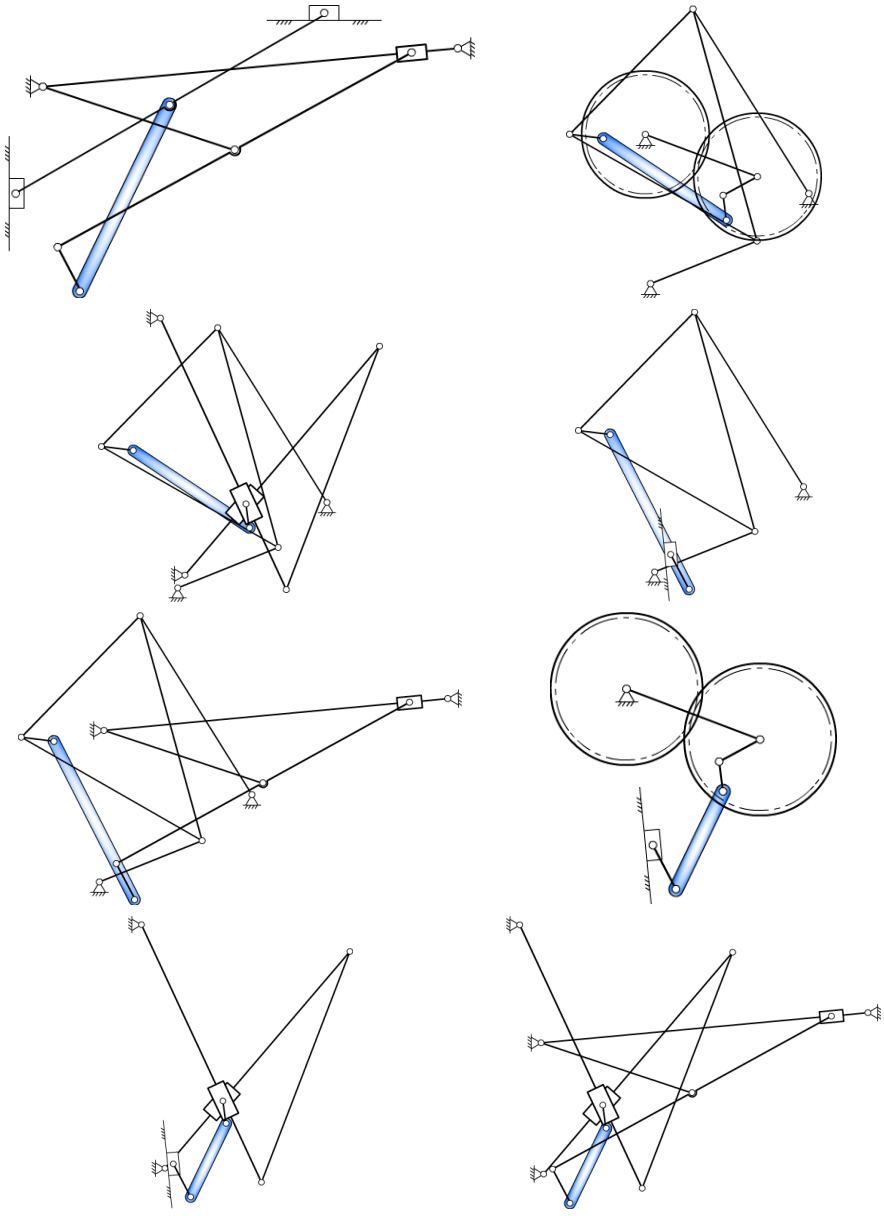


Figure 7.16: 8 synthesized mechanisms

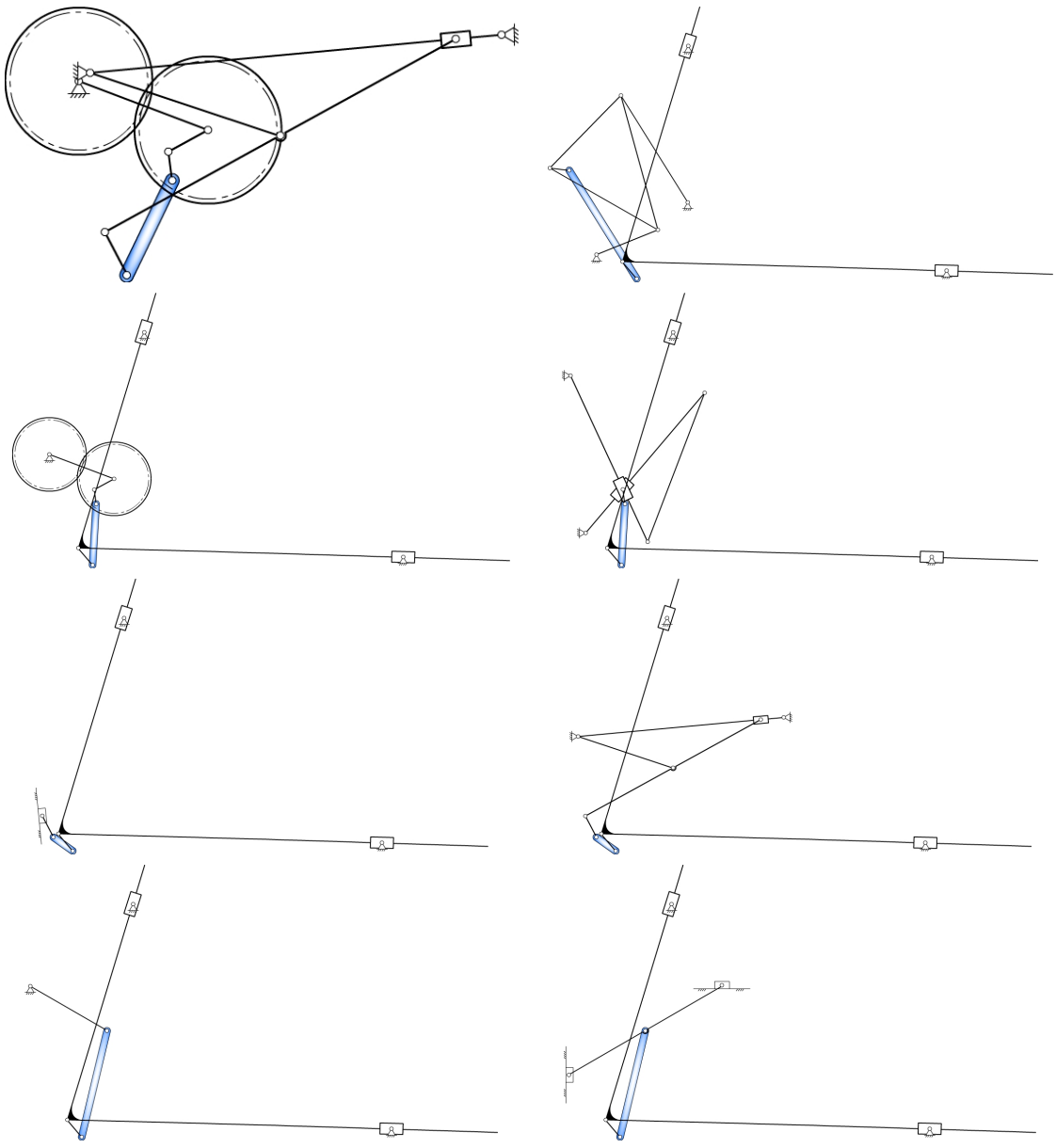


Figure 7.17: 8 synthesized mechanisms

Chapter 8

On Representation and Identification of Geometric Constraints Using Fourier Descriptors

This chapter seeks to apply the concept of Fourier Descriptors (FD) to the problem of estimating or extracting geometric constraints from a given planar motion task. The problem, referred to as *kinematic acquisition of geometric constraints*, was recently introduced in the framework of task centered mechanism design [118].

Two representations of a motion are discussed in [118], one is explicit and the other implicit. A motion that is given directly in terms of the displacements belonging to it is said to be in *explicit form*, and a motion that is defined indirectly in terms of geometric constraints is said to be in *implicit form*. The former can be prescribed either as a continuous motion in parametric form or as a discrete motion via an ordered sequence of displacements; the latter can be imposed either geometrically or algebraically. For an example, as shown in Figure 8.1 the motion of the coupler link AB of a planar four-bar linkage is constrained geometrically such that two of the points on the coupler link stay on two separate circles. Alternatively, in algebraic form, one may express that the coordinates of each of the two points satisfy the

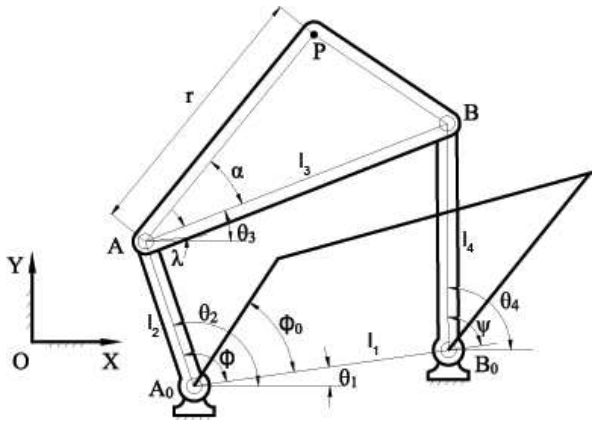


Figure 8.1: A crank-rocker mechanism

algebraic equation of each of the two circles. The coupler motion defined in this way is said to be in its implicit form. On the other hand, the motion of the same coupler can also be represented directly as a parameterized set of displacements of the coupler link relative to a fixed reference frame, i.e., in terms of the coordinates of the point A as well as the angle θ_3 that the coupler link makes from the horizontal axis.

When geometric constraints of a planar motion are completely specified, the objective for converting the motion from the implicit form to its explicit form, called *explicitation*, is to determine a set of displacements, velocities and accelerations, that satisfy the specified constraints. In addition, when the given constraints happen to define a kinematic mechanism, the conversion problem can be made equivalent to kinematic analysis of the resulting mechanism. When a motion is explicitly given, the objective for converting the motion from the explicit form to its implicit form, called *implicitation*, is to determine the geometric constraints that define the motion indirectly. When the resulting geometric constraints can be generated mechanically

with a mechanism, the conversion problem can be made equivalent to mechanism synthesis. The analysis and synthesis of planar mechanisms are highly developed subjects, especially when it comes to the analysis and synthesis of four-bar linkages.

This chapter studies the problem of representing simple geometric constraints that can be generated mechanically using Fourier descriptors and demonstrate how they can be used for estimating geometric constraints from an explicitly given motion. The focus is on how a motion can be constrained geometrically with various constraints instead of matching a given motion to a particular mechanism directly.

8.1 Fourier Descriptors of Closed Curves

The problem of computing the Fourier descriptors directly from the point coordinates of a closed curve has been studied in [119, 120]. A planar curve C in complex form is given by

$$\mathbf{z}(t) = x(t) + iy(t) \tag{8.1}$$

where $x(t), y(t)$ are the Cartesian coordinates of a point of C for the parameter value t. When t varies in $[0, T]$, where T is the period, the function $\mathbf{z}(t)$ traces out a closed curve. Fourier coefficients associated with $\mathbf{z}(t)$ can be obtained from either (3.2) or (3.9).

Freudenstein [96] was the first to explore the use of Fourier transform for four-bar linkage analysis and synthesis. This work was followed by Funabashi [97], Farhang et al. [98, 99], Chu and Cao [100], and McGarva [104, 105]. Instead of applying Fourier transform directly to a closed curve, Ullah and Kota [94] obtained Fourier descriptors from the angular function resulting from the tangents of a closed

coupler curve of a four-bar linkage. They went on to show that this leads naturally to a Fourier representation of the coupler curve that is invariant with respect to scaling, rotation and translation. In addition, they observed that, for the case of four-bar coupler curves, the use of a small set of Fourier coefficients associated with the low-harmonic components is sufficient for shape comparison. Recently, Wu et al. [108] showed that the search space can be reduced even further and thus leads to an even more efficient method for four-bar linkage synthesis for closed path generation. This work has been further refined and extended to both open and closed path generation [121].

8.2 Fourier Representations of Low-Harmonic Geometric Constraints

As alluded to earlier, the Fourier representation of a closed curve allows for the level-of-detail interpretation in the frequency domain. Low frequency descriptors represent general features of a shape, while higher frequency descriptors add the finer details. We now consider simple geometric shapes such as an ellipse, a circle and a line-segment and see how they can be represented by one- or two-harmonic components, i.e., a low-harmonic Fourier series consisting of $\{v_{-2}, v_{-1}, v_0, v_1, v_2\}$ associated with frequencies $\{-2\omega, -\omega, 0, \omega, 2\omega\}$.

8.2.1 Ellipse

Consider an ellipse shown in Figure 2 centering at (x_0, y_0), whose major axis makes an angle γ from the X-axis. Let $v_0 = x_0 + iy_0$ denote the complex number rep-

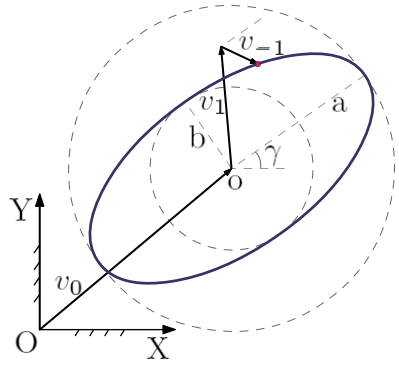


Figure 8.2: An ellipse and its Fourier representation in terms of its 1-harmonic components v_{-1} and v_1.

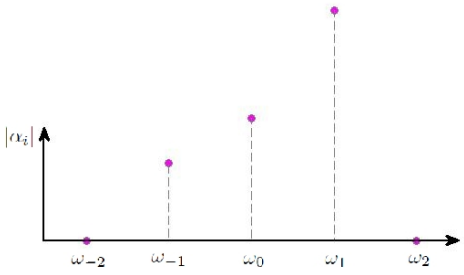


Figure 8.3: The amplitude spectrum of an ellipse: amplitudes of higher harmonics are zero.

resentation of the center of the ellipse. A point on the ellipse with the parametric angle $\theta = \omega t + \theta_0$, where ω is the fundamental frequency and θ_0 is the initial angle, is given by

$$\mathbf{z}(\theta) = v_0 + (a\cos\theta + ib\sin\theta)e^{i\gamma} \tag{8.2}$$

where a, b are the major and minor semi-axes of the ellipse. This leads to the 1-harmonic representation of the ellipse:

$$\mathbf{z}(t) = v_0 + v_1 e^{i\omega t} + v_{-1}e^{-i\omega t} \tag{8.3}$$

where $v_1 = r_1 e^{i\phi_1}$ and $v_{-1} = r_{-1} e^{i\phi_{-1}}$ and

$$r_1 = \frac{a+b}{2}, \;\; r_{-1} = \frac{a-b}{2}, \tag{8.4}$$

$$\phi_1 = \theta_0 + \gamma, \;\; \phi_{-1} = -\theta_0 + \gamma. \tag{8.5}$$

It is clear from the above that v_0 represents the translation that leaves the shape, size, and orientation of the ellipse unchanged. The amplitudes of v_1, v_{-1}, i.e., the Fourier descriptors r_1, r_{-1}, determine a, b and thereby the shape and size of the ellipse. In particular, we have

$$\frac{r_{-1}}{r_1} = \frac{s-1}{s+1} < 1, \;\; s = \frac{a}{b}, \tag{8.6}$$

i.e., the ratio of the Fourier descriptors is related to the aspect ratio of the ellipse. The phases of v_1, v_{-1} determine the orientation of the ellipse as well as the starting point. So if we are only concerned with the shape and size of the ellipse, all we need are the Fourier descriptors r_1 and r_{-1} (Figure 3).

Note 1: It can be seen that as t increases from 0, Eq. (8.2) generates points on the ellipse in counterclockwise sequence starting from the point $f(0) = v_0 + ae^{i\gamma}$ (see Figure 2).

Note 2 If $\theta = -\omega t + \theta_0$ is used instead in (8.2), i.e., when the points are generated in clockwise direction, then the roles of v_1 and v_{-1} are switched. In this case, we have $r_{-1}/r_1 > 1$.

8.2.2 Circle

When $a = b$, the ellipse becomes a circle (see Figure 4). It follows that $r_1 = a$ and $r_{-1} = 0$ and that

$$\mathbf{z}(t) = v_0 + v_1 e^{i\omega t}. \tag{8.7}$$

Figure 5 and 6 show the amplitude spectrums of a circle.

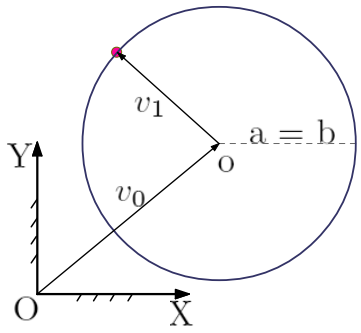


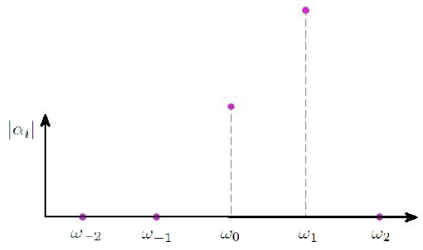
Figure 8.4: A circle whose points are ordered in counterclockwise direction.

8.2.3 Curves Defined By Two- or More Harmonic Components

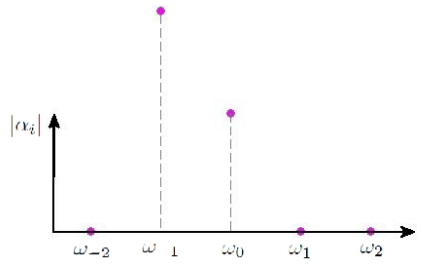
It is easy to show that the following Fourier components associated with the second-order harmonics $e^{\pm i2\omega t}$:

$$\mathbf{z}_2(t) = v_2 e^{i2\omega t} + v_{-2} e^{-i2\omega t} \tag{8.8}$$

represents an ellipse in the same way as the first-order harmonics (8.3) does but with doubled frequency. In other words, as t varies in $[0, T]$, $\mathbf{z}_2(t)$ traces out an ellipse twice. It follows that the Fourier transform of a periodic complex function $\mathbf{z}(t)$ as given by (3.1) results in a sum of ellipses of increasing frequency. While



(a)



(b)

Figure 8.5: (a) The amplitude spectrum of a counterclockwise oriented circle: $v_{-1} = 0$; (b) The amplitude spectrum of a clockwise oriented circle: $v_1 = 0$

the low frequency components defines the rough shape of the curve for the entire period, the high frequency components provide finer details.

8.3 Straight Line Segments

Here we consider a straight line segment as a degenerated form of a closed curve, that shrinks to overlap. A straight line-segment can be represented by a collapsed ellipse when $b = 0$ in equations from (8.2) to (8.5). In this case, the equation of a line-segment with length $2a$ is given by

$$\mathbf{z}(\theta) = v_0 + a \cos\theta e^{i\gamma} \tag{8.9}$$

which is exactly the projection of the ellipse (8.2) onto its major axis. Alternatively, we have $r_1 = r_{-1} = a/2$ when $b = 0$, i.e., the two Fourier descriptors associated with the first-order harmonics are of equal value.

It is possible to define a line-segment using more than one-harmonic components. Consider the following 2-harmonic curve:

$$\mathbf{z}(t) = v_0 + v_1 e^{i\omega t} + v_{-1} e^{-i\omega t} + v_2 e^{i2\omega t} + v_{-2} e^{-i2\omega t}. \tag{8.10}$$

where $v_j = r_j e^{i\phi_j}$ with $j = -2, -1, 0, 1, 2$. The ellipse associated with the first order harmonic term collapses into a line-segment when $r_1 = r_{-1}$. The angle that the resulting first-order line-segment makes with the horizontal axis is γ which can be obtained from (8.5) as

$$\gamma = \frac{\phi_1 + \phi_{-1}}{2}. \tag{8.11}$$

Similarly, the ellipse associated with the second order harmonic term represents a line-segment when $r_2 = r_{-2}$. In order to align the two line-segments, we must have

$$\gamma = \frac{\phi_2 + \phi_{-2}}{2}. \tag{8.12}$$

Thus we have the following conditions for a 2-harmonic line-segment (see Figure 6, 7, 8):

$$\begin{cases} r_{-1} = r_1, \;\; r_{-2} = r_2, \;\; \text{and} \\ \frac{1}{2}(\phi_{-1} + \phi_1) = \frac{1}{2}(\phi_{-2} + \phi_2) = \gamma \end{cases} \tag{8.13}$$

8.4 Geometrically Constrained Motions

A planar motion of a rigid body has three degrees of freedom. If a point (B_x, B_y) on the moving body is required to stay on a curve $f_1(x, y) = 0$, then we obtain a two-degrees-of-freedom (dof) motion; if another point (D_x, D_y) of the moving body is

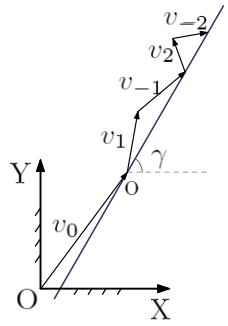


Figure 8.6: This figures shows how a line segment is constructed using one- and two-harmonic components.

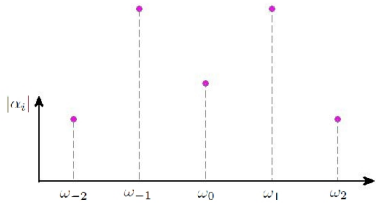


Figure 8.7: The amplitude spectrum of a overlapped line segment.

also required to stay on a different curve $f_2(x, y) = 0$, then we obtain an one-dof motion. Each of these curves can be a straight line-segment, a circle, an ellipse or a more complex curve such as the coupler curve of a four-bar linkage. Consider a slider-crank mechanism shown in Figure 9. The moving body is constrained such that the point B lies on a circle and the point D lies on a line-segment. The point, C, is a general point on the coupler and it traces out a more complex curve.

In this paper, we are interested in geometric constraints that can be generated mechanically. Table 8.1 lists the geometric constraints and their corresponding 1-dof generators.

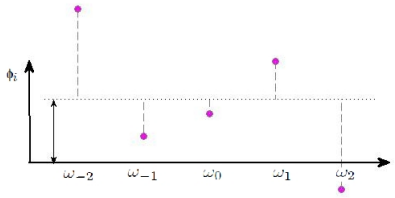


Figure 8.8: The phase spectrum of a line segment: ϕ_i is the phase of the $i$the harmonic component.

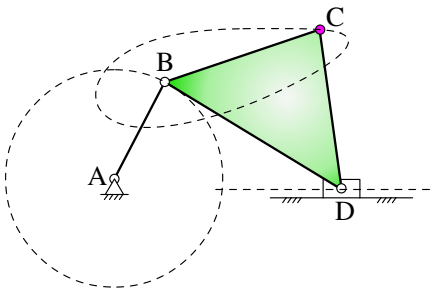


Figure 8.9: A slider-crank mechanism

8.5 Trajectory Analysis and Constraint Identification

For an explicitly given motion, whether continuously as a function of time or discretely as an ordered sequence of displacements, the goal is to identify a point on the moving body such that its trajectory best matches with one of the predefined low-harmonic geometric constraints such as a line-segment, a circle, an ellipse or even a four-bar coupler curve. This done by analyzing the trajectory of all points within a feasible zone using Fourier transform and then search in the frequency domain to see if any of the trajectory matches a specified low-harmonic constraint. In this paper, a direct search method is used to compute the error between a given

Geometric Constraints	Mechanism Linkages
Ellipse	Geared-2R, $\cdots$
Circle	1R, $\cdots$
Straight Line	1P, $\cdots$
Complex Shape	Crank-Rocker, $\cdots$

Table 8.1: Linkage Classification by Geometric Constraints

constraint and a selected trajectory. The error is defined in terms of squares of their respective Fourier descriptors. All errors can be plotted into a colored map. To set up an error map, we select an designer interested region from the moving rigid body. Secondly, we generate a grid of points on that region. Thirdly, we generate a trajectory of every sampled point. Thereafter, we compute the shape errors between each of the trajectory and the given constraint curve. In the end, a linear color interpolation is applied to the designer interested region in order to visualize the scaled relative shape errors. This color map is called the *error map* for one give motion and one type of constraint curve.

Synthesis Procedures

1. The closed task motion is explicitly represented in any format, like rational B-Spline motion, or matrix expressed motion, or any other designer favorite way.

2. The designer specifies an interested observing region, where is the desirable area to amount the mechanical linkage to.

3. Sample the designer interested region, and compute their trajectory. Dense

sampling is recommended for large area but should not exceed the computational capability.

4. Query every type of geometric constraints, and set up the their error maps. These error maps depend on the input motion, so they are built dynamically, and serve as evaluation guidance to match trajectories and geometric constraints.

5. The designer looks up the error maps, navigates points of the rigid body, and examines the shape, the amplitude and the phase spectrums of the trajectories. With the comprehensive references, the designer picks the desired trajectory, the trace to constrain one dyad.

6. Hereafter, it is moved to the step of synthesis. The geometric constraint query is requested. The mechanism database delivers the dyad types that are compatible with the selected trajectory, under the criterion of the geometric constraint.

7. Finally, the designer select the dyad type, and continues the dimensional synthesis using the existing methods.

8. Repeat the steps 5-7 till the terminal of the design.

8.6 EXAMPLES

In this section, we present one simple example that demonstrates how our method provides multiple synthesis solutions for a given B-Spline motion.

For geometric constraints, this example concentrates on four types of shapes, including ellipse, circle and straight line, and exclusively assuming the rest shapes belong be the complex group. Concisely, we assign only one representative mechanism linkage to each geometric constraint, as shown in Table 8.1.

The motion of the coupler link is presented as a uniform ratiopnal cubic C^2 B-spline motion. Table 8.2 gives the Cartesian parameters for the 16 control positions. Without loss of generality, the interval of the knot sequence is set to be equal. The motion is shown in Figure 10. We select the rectangular observing region, the

i	X	Y	α
1	3.4788	0.7772	0
2	1.7164	2.5350	1.2258
3	-0.2225	2.1049	2.4045
4	-0.6083	0.8492	-2.7924
5	-0.0166	0.4004	-1.8403
6	0.3063	0.7591	-1.0589
7	0.0508	1.1131	-0.4783
8	-0.3638	0.9936	-0.1207
9	-0.5035	0.4008	0
10	-0.2198	-0.5303	-0.1207
11	0.3170	-1.7028	-0.4783
12	0.6541	-2.9200	-1.0589
13	0.3599	-3.5819	-1.8403
14	-0.2605	-2.8299	-2.7924
15	0.0437	-0.7110	2.4045
16	1.8604	1.0111	1.2258

Table 8.2: 16 Control Positions

shaded rectangle in Figure 10, identified by the diagonal ending at points $(-6, -3)$ and $(6, 3)$. 100 by 100 Points are sampled and their trajectories are calculated.

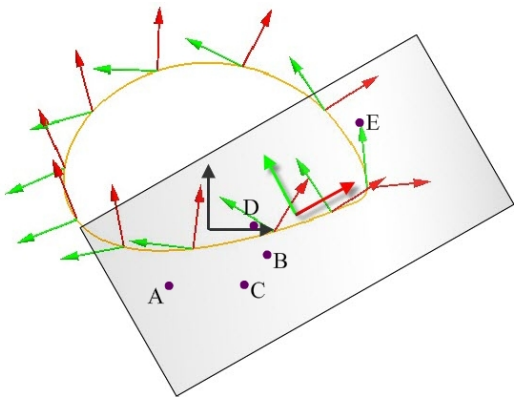


Figure 8.10: The given motion: The black frame is fixed, and the shaded frame is the moving frame. The dark square is selected for observation. Point A, B, C, D and E are the positions to attach mechanism.

We seek to extract from sampled trajectories to obtain a simple geometric constraint such as an ellipse, a circle, or a straight line or low-harmonic coupler curve of a four-bar linkage, respectively, and thereafter, render the corresponding error maps. As shown in Figure 11 through 14, the error maps provides a visualization for the quality of the constraint matching. The color bar, existing along with every error map, is the reference of error scale. The area of the error maps all corresponds to the shaded region in Figigure 10. The number next to the color bar is the greatest error of that map. Obviously, the smaller the error is, the more precisely the geometric constraint matches the trace of that point, and more reliable the synthesis will be. Moreover, the error maps intrinsically appear to be the gradient fields of errors, and it is the prominent aspect of the error maps that they also imply the better searching area, if the currently selected region dose not cover any reasonable candidate geometric constraints. For example, if we initially observe only the right-upper

area instead of the entire current region, then there is no pleasant point satisfying a circle, but fortunately, the colorful gradient field clearly advises the search of the left-lower area for a better circular point constraint. Simultaneously, the plots

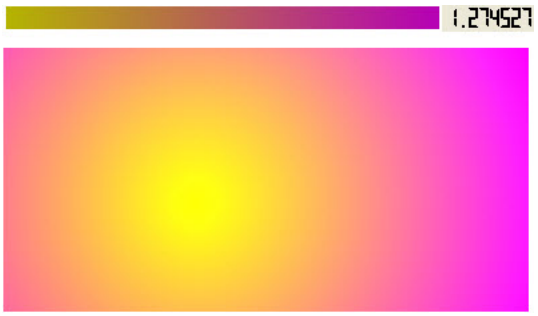


Figure 8.11: The error map for the point constraint of an ellipse: Yellow points have smaller error than the pink ones for approximation of an ellipse.

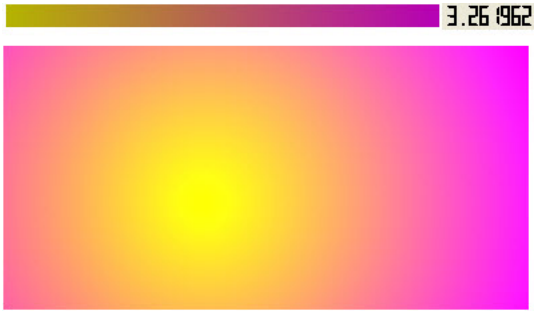


Figure 8.12: The error map for the point constraint of a circle: yellow points have smaller error than the pink ones for approximation of a circle.

of amplitude and phase spectrum expose the designers to more information of the trajectories, although this requires the designers to possess basic knowledge about the spectral characteristics of the geometric constraints. The observation of both the error maps and the spectrums of point trajectories suggests that 1, the point A is more suitable for a straight line segment, 2, the point B creates satisfactory circles

Figure 8.13: The error map for the point constraint of a line segment: yellow points have smaller error than the pink ones for approximation of a a line segment.

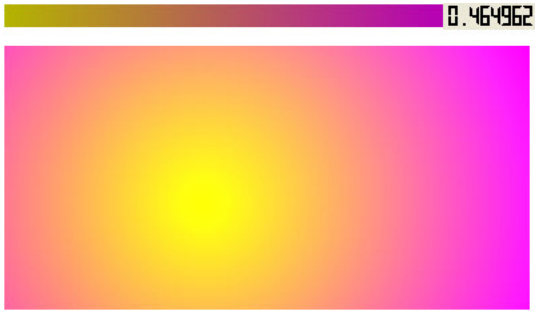


Figure 8.14: The error map for the point constraint of a general curve dominated by the five lowest harmonics: for this map, the trajectory of each point is approximated using the five lowest-frequency harmonics. Yellow means a better approximation.

3, the point C generates an ellipse-alike trace, and 4, both points D and E plot the body shape of a dolphin.

In view of the geometric constraints, a prismatic joint is attached to A, a crank is recommended for B, a geared 2-R matches C , and the design experience assigns the crank-rocker mechanisms for the body shape of a dolphin by D and E. Ultimately, traditional methods of dimensional synthesis are implemented to figure out the best size for each dyad. As what immediately precedes, diverse design solutions yield by the combination of different dyads comprise, for examples, a crank-slider,

a geared 4R-1P, a Stephenson III six-bar, a geared seven-bar, and a eight-bar mechanism.

Conclusion

In this chapter, we studied the problem of how to represent and identify simple constraint curves using Fourier descriptors. This is used to analyze and identify geometric constraints from an explicitly given motion. A direct search method is presented for the constraint identification process. The results provide a basis for task centered design of mechanisms.

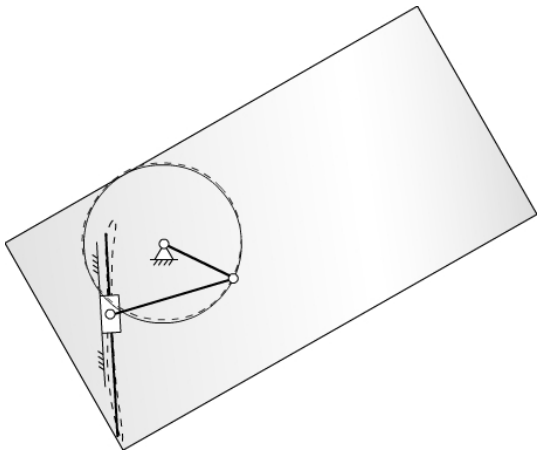


Figure 8.15: This mechanism is combined by a slider and a crank.

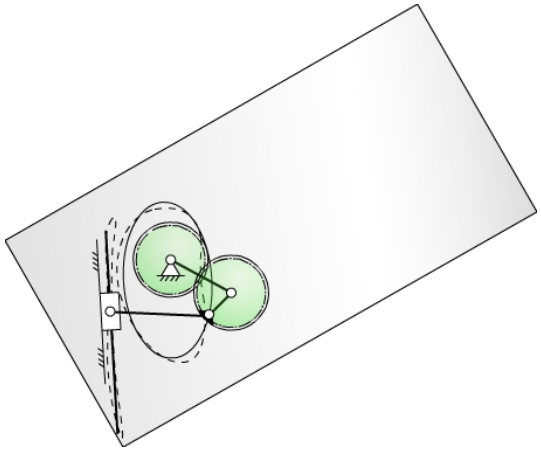


Figure 8.16: This mechanism is combined by a slider and a geared 2R.

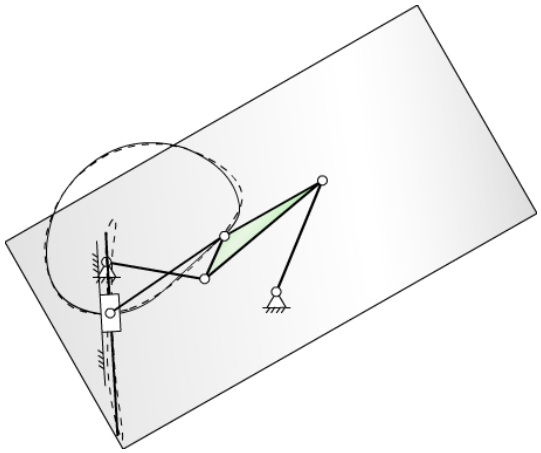


Figure 8.17: This mechanism is combined by a slider and a crank-rocker mechanism.

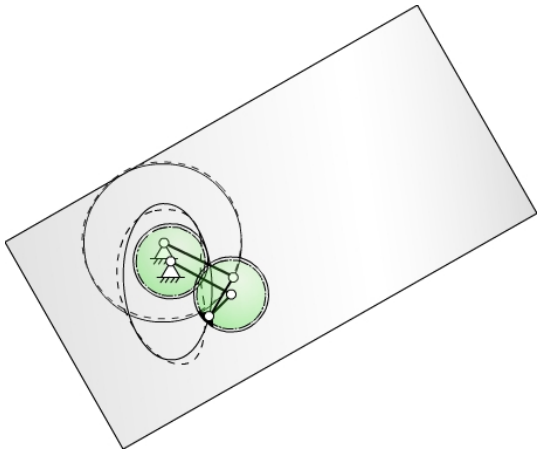


Figure 8.18: This mechanism is combined by a crank and a geared 2R.

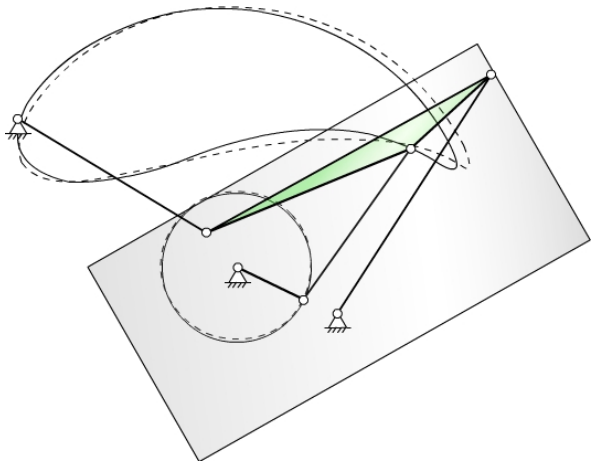


Figure 8.19: This mechanism is combined by a crank and a crank-rocker mechanism.

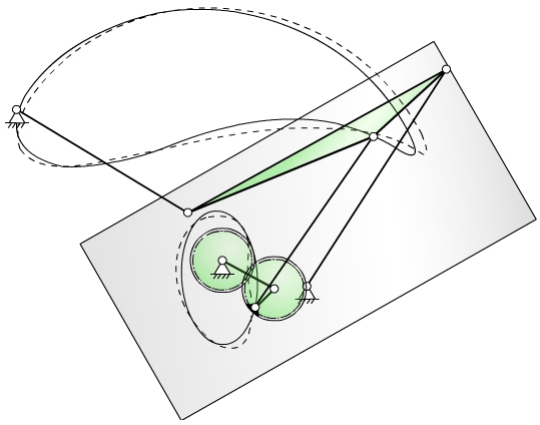


Figure 8.20: This mechanism is combined by a geared 2R and a crank-rocker mechanism

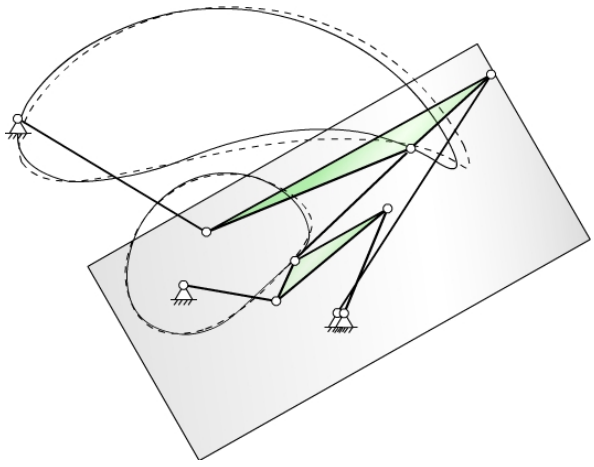


Figure 8.21: This mechanism is combined by two crank-rocker mechanisms.

Chapter 9

Fourier Descriptors with Different Shape Signatures: a Comparative Study for Shape Based Retrieval of Kinematic Constraints

This chapter deals with the problem of analyzing geometric constraints associated with a given motion from the perspective of computational shape analysis. The goal is to develop computational methods for retrieving or extracting geometric constraints from a specified motion. This problem is studied as a shape retrieval problem, which involves three main issues: shape representation, shape similarity measure and shape indexing. Among them, shape representation is the most important issue.

The use of shape descriptors often simplifies the process of not only shape comparison but also shape storage and classification. There are many shape descriptors proposed for various application domains. They include boundary scalar transform such as the centroid-to-boundary distance function, global scalar transform such as moment-based methods, and global space domain methods such as the media axis transform. Hoeltzel and Chieng [95] was one of the early adopters of computational

shape analysis techniques for comparing the coupler curves for the dimensional synthesis of four-bar linkages. They used moment invariants to represent and compare coupler curves.

Of our interest is a Fourier transform based method for characterizing the shape function. Freudenstein [96] was the first to explore the use of Fourier transform for four-bar linkage analysis and synthesis. This work was followed by Funabashi [97], Farhang et al. [98, 99], Chu and Cao [100], and McGarva [104, 105]. As shown by Ullah and Kota [94], when reformulated using the Fourier descriptors, the error function that captures the deviation between the synthesized and desired path, leads naturally to a reduced search space that decouples the comparison of shapes from their size, location, and orientation. In addition, they observed that, for the case of four-bar coupler curves, the use of a small set of Fourier coefficients associated with the low-harmonic components is sufficient for shape comparison. This has been independently observed by cognitive scientists for human judgement of perceptual shape similarity[107]. The combined effect of decoupled search space with small set of Fourier coefficients leads to drastic improvement in optimization routine for four-bar linkage synthesis.

In this chapter, we focus on a comparative study of Fourier descriptors derived from various shape signatures of planar closed curves. We will conduct the comparison not only using commonly used criteria for shape representation and identification but also in the context of shape based retrieval of kinematic constraints for task centered mechanism design.

9.1 Shape signatures

In general, a shape signature is any 1-D function representing 2-D areas or boundaries. Four shape signatures are considered in this paper, these are central distance, complex coordinates (position function), curvature and cumulative angular function. The reason for choosing these four shape signatures for test and comparison is because they are mostly used in recent FD implementations and have been shown practical for general shape representation. In the following, we assume the shape boundary coordinates $(x(t), y(t)), t = 0, 1, \cdots, L-1$, have been extracted in the preprocessing stage.

9.2 Explicit Representations of Closed Curves

In numerical implementation, a continuous closed curve, see Fig 9.1, is typically extracted as a polygon consisting of a sequence of sample points $\mathbf{P}_i$ coordinated at (x_i, y_i), $i = 0, 1, \cdots, N-1$, and the arc length s_i between two adjacent points $\mathbf{P}_i$ and $\mathbf{P}_{i+1}$ is approximated by their distance.

$$s_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{9.1}$$

As the number of sample points increase, the polygon and the arc length converge to the continuous curve and the actual arc length, respectively.

Various representations of a curve exist in the literature. However, representations based on different curve features have significantly distinct impact on the result of shape comparison. For the purpose to develop this paper, we briefly review four types of shape signatures that are mostly used to derive Fourier descriptors, in-

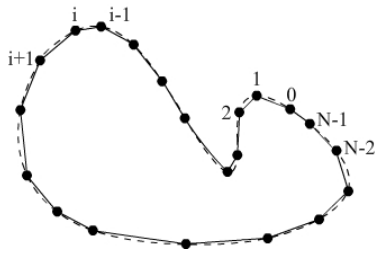


Figure 9.1: The continuous curve in dash is numerically approximated by a polygon in solid which connects a set of sample points of the continuous curve.

cluding complex coordinate, centroid distance, curvature signature and cumulative angle.

9.2.1 Complex Coordinates

Complex coordinates also known as position function represent a point as a complex number on the complex plane. In order to make this representation invariant to translation, the position function is defined as a displacement of the sample points from centroid, such that

$$z_i = (x_i - x_c) + j(y_i - y_c) \tag{9.2}$$

where the definition of centroid (x_c, y_c) is given by the following equation

$$x_c = \frac{\sum_{i=0}^{N-1} x_i s_i}{\sum_{i=0}^{N-1} s_i}, \quad y_c = \frac{\sum_{i=0}^{N-1} y_i s_i}{\sum_{i=0}^{N-1} s_i}. \tag{9.3}$$

by considering the account of the nonuniform sampling interval.

9.2.2 Centroid Distance

The centroid distance denotes the distance of a sample point to the centroid, which is eventually the magnitude of position function defined in Eq. (9.2).

$$r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \tag{9.4}$$

Centroid distance is also invariant to translation.

9.2.3 Cumulative Angle

A curve can be represented by using its tangent angle at each point. Denotes the tangent angle at $\mathbf{P}_i$ with respect to horizontal axis as θ_i. To remove the effect of orientation, the cumulative angle at $\mathbf{P}_i$ is defined as the net angular bend of the starting point from the initial.

$$\varphi_i = \theta_i - \theta_0 \tag{9.5}$$

The cumulative angle is discontinuous at 2π, since as tracing a closed curve from the initial point to the end counterclockwise, it increase from 0 to 2π. To deal with this discontinuity, a linear increasing angle is subtracted from the cumulative angle.

$$\psi_i = \varphi_i - \frac{2\pi \cdot \sum_{k=0}^{i-1} s_k}{\sum_{k=0}^{N-1} s_k} \tag{9.6}$$

This angular function is periodic and invariant to both translation and rotation.

9.2.4 Curvature Signature

Curvature, an intrinsic property of a curve, measures the degree by which a curve deviates from being straight. In differential geometry, curvature is defined as the

first derivative of the curve tangent with respect to arc length. The curvature at $\mathbf{P}_i$ can be approximated as

$$k_i = \frac{\theta_{i+1} - \theta_i}{s_i} \tag{9.7}$$

which means the change of tangent angle normalized by the associated arc length as a point moves from $\mathbf{P}$ to $\mathbf{P}_{i+1}$. Curvature is invariant to both translation and rotation.

9.3 Shape Comparison with Fourier Descriptors

The concept of Fourier Descriptors (FD) has been widely used in the field of Computational Shape Analysis [52]. It is an extension of the concept of Fourier transform and uses the Fourier coefficients in the frequency domain for shape representation. The great advantage of the Fourier representation is the level-of-detail interpretation in the frequency domain. Low frequency descriptors (coefficients) represent general features of a shape, while higher frequency descriptors add the finer details. As the result, the dimensions of the Fourier descriptors used for indexing shapes are significantly reduced. In addition, FD features can be made normalized to translation, rotation and scale. Another interesting feature of the Fourier descriptor is that it has been shown to be consistent with human judgement of perceptual shape similarity [122, 107].

9.3.1 Fourier Transform on Curve Signatures

Fourier transform on different curve signatures gives the rise to different FDs. By replacing $\mathbf{z}(t)$ in Eq. (3.7) with the signature function of a curve,

$$\mathbf{z}(t) \in \{z_i, r_i, \psi_i, k_i\} \tag{9.8}$$

the corresponding $|\alpha_n|$ are usually called Fourier Descriptors of that curve, denoted as FD_n, $n = 0, 1, \cdots, N-1$.

The FDs of a curve derived based on curve signatures are subjective to the parametrization, the size and the initial point of the curve.

Fourier transform is an operation involving the parametrization of the periodic function. FDs on the same set of samples of the same curve yet associated with different parametrization are not identical. Intrinsic parametrization is referred to handle the issue of parametrization. In this paper, we adopt the cumulative arc length to be the parameter. Therefore, one curve has unique FDs. In this case, t_i stands for the length of the curve starting from point $\mathbf{P}_0$ and ending at $\mathbf{P}_i$, and Δt_i denotes that of the curve connecting $\mathbf{P}_i$ and $\mathbf{P}_{i+1}$, and T is the perimeter:

$$t_i = \sum_{k=0}^{i-1} s_k, \quad \Delta t_i = s_i, \quad T = \sum_{k=0}^{N-1} s_k \tag{9.9}$$

Except cumulative angle, $\{\psi_i\}$, the other three curve signatures, $\{z_i, r_i, k_i\}$, are proportional to the curve size. So the FDs derived based on the curve signatures are also scaled by the size of the curve, simply because Fourier transform is a linear operation on signature function $\mathbf{z}(t)$. To compare two curves, one must be resized so as to attain the closest matching with the other one.

Now we consider the impact of initial point on different FDs. FDs derived from the position function, the centroid distance or the curvature signature are invariant to the choice of initial point. For the same samples of a given curve, if these three signature functions are constructed by starting from different initial points, their parametrization undergoes a constant shift. The constant parametrization shift changes the phase spectrum of the Fourier transform but not amplitude spectrum, and thus, has no impact on the derived FDs. However, the cumulative angle is exception. If the construction of the cumulative angle starts from different initial points, constant shift exists not only in parametrization but also in the signature value, simply because the cumulative angle is a relative value of the observation angle to the initial angle θ_0, starting and ending with value 0. While the parametrization shift does not triggers the change of FDs, the constant shift of the signature value brings FD_0 to a different value. In general, except that FD_0 for the cumulative angle depends on the choice of initial point, FDs derived based on any signature function are invariant to the choice of initial point. To label a curve with FDs, $\{FD_n, n = 0, 1, \cdots, N-1\}$ can be used for the position function, the centroid distance, and the curvature signature, yet $\{FD_n, n = 1, \cdots, N-1\}$ for cumulative angular function.

Freudenstein [96] was the first to explore the use of Fourier transform for four-bar linkage analysis and synthesis. This work was followed by Funabashi [97], Farhang et al. [98, 99], Chu and Cao [100], and McGarva [104, 105]. Instead of applying Fourier transform directly to a closed curve, Ullah and Kota [94] obtained Fourier descriptors from the angular function resulting from the tangents of a closed

coupler curve of a four-bar linkage. They went on to show that this leads naturally to a Fourier representation of the coupler curve that is invariant with respect to scaling, rotation and translation. In addition, they observed that, for the case of four-bar coupler curves, the use of a small set of Fourier coefficients associated with the low-harmonic components is sufficient for shape comparison. Recently, Wu et al. [108] showed that the search space can be reduced even further and thus leads to an even more efficient method for four-bar linkage synthesis for closed path generation. This work has been further refined and extended to both open and closed path generation [109].

9.3.2 Shape Similarity

To measure the similarity of two curves, curve C_a and curve C_b in spectrum domain of one specific signature function, we resort to the closeness of their FDs. As aforementioned, we scale one curve to eliminate the impact of their size difference. If C_a is the standard shape we are interested, and C_b is one candidate curve to be considered whether it is similar to C_a, we resize curve C_b and preserve the size of curve C_a as we compare them. By fixing the size of the standard shape, it is allowed to sort the similarity of the candidate curves with respect to the standard curve by relying on their FD differences.

Let FD_i^a denote the ith component of the FDs of curve C_a, FD_i^b label that of curve C_b, and C_b be scaled by λ times for the purpose to eliminate the impact of size on shape comparison. Then the structure error defined as the Euclidean distance of

their FDs is given as below

$$\delta_{a,b} = \sqrt{\sum_i (FD_i^a - \lambda \cdot FD_i^b)^2} \tag{9.10}$$

FDs derived based on the position function, the centroid distance or the curvature signature are invariant the choice of initial point, so that we take all FD terms into account and we have $i = 0, 1, \cdots, N-1$. λ for these FDs are determined based on the criterion to minimize the structural error $\delta_{a,b}$, and as a result, when two curves are of exactly the same shape but two sizes, the computed structural error after resizing goes to zero. In this way, we have

$$\lambda = \frac{\sum_i FD_i^a \cdot FD_i^b}{\sum_i (FD_i^b)^2} \tag{9.11}$$

The choice of the initial point affect the constant term of the FDs derived based on cumulative angle, so that the constant component is not considered and we have $i = 1, \cdots, N-1$. However, the cumulative angle is invariant to the size of the curves, and hence, resizing is not needed, $\lambda = 1$ for FDs derived based on cumulative angle.

Since Fourier transform essentially project a vector onto a set of orthonormal basis, FDs of a curve are the projections of that curve. When two curves have similar shapes and they are represented with the same type of signature, they are expected to possess close FDs as projected onto the same basis. Henceforth, the small structure error indicated the reasonable similarity between the two curves.

9.4 Geometrically Constrained Motion

A planar motion of a rigid body has three degrees of freedom. If a point (B_x, B_y) on the moving body is required to stay on a curve $f_1(x, y) = 0$, then we obtain a two-degrees-of-freedom (dof) motion; if another point (D_x, D_y) of the moving body is also required to stay on a different curve $f_2(x, y) = 0$, then we obtain an one-dof motion. Each of these curves can be a straight line-segment, a circle, an ellipse or a more complex curve such as the coupler curve of a four-bar linkage. Consider a slider-crank mechanism shown in Figure 9.2. The moving body is constrained such that the point B lies on a circle and the point D lies on a line-segment. The point, C, is a general point on the coupler and it traces out a more complex curve.

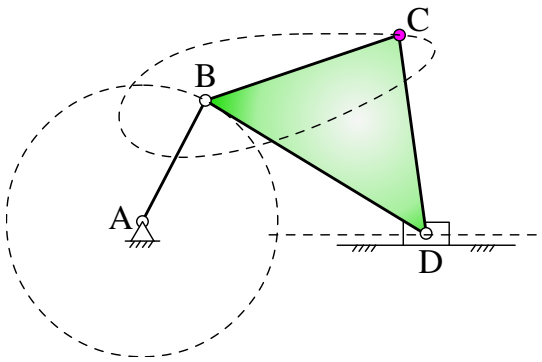


Figure 9.2: A slider-crank mechanism.

In this paper, we are interested in the primitive geometric constraints that can be generated mechanically.

9.5 Simple Geometric Constraints

This section presents a set of simple geometric constraints that can be readily provided using simple mechanisms. As mentioned in the proceeding section, all parameters are normalized on $[0, 1]$. All constraints are represented by closed curves and special attention has been paid to their orientations as they play a significant role in shape matching process. Specifically, the same set of points are viewed as two distinct constraints if they are traced in two opposite orientations, clockwise and counterclockwise.

A circular constraint is represented by a unit circle centered at the origin of the coordinate system and with the initial point locating at $(1, 0)$, Figure 7.5.

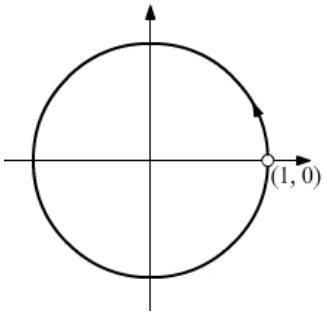


Figure 9.3: The circular constraint traced in counterclockwise direction.

A circular-arc constraint is a segment of the circumference of a unit circle. It starts from point $(1, 0)$ and extends counterclockwise, so that the central angle is always positive. The central angle remains a variable in the process of constraint identification. To make the circular arc closed, we represent it as a forward-and-backward plotted curve as shown in Figure 7.6.

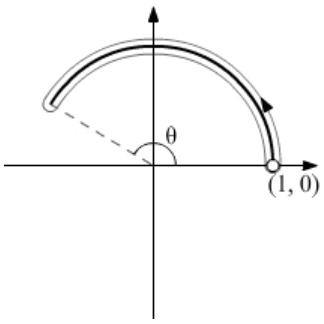


Figure 9.4: The counterclockwise traced circular arc with center angle θ.

A line segment along the x-axis starts at $(1, 0)$ and ends at $(-1, 0)$. It is also represented as double-traced closed curve, Figure 7.7.

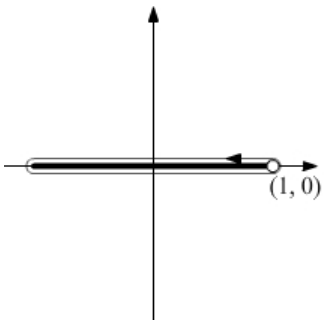


Figure 9.5: A line-segment is represented as a closed curve traced in counterclockwise direction.

An ellipse is located at $(0, 0)$. Its major axis lies on x-axis, and minor axis on y-axis. The length of semi-major axis is 1. The length of semi-minor axis is freely adjustable and ranges from 0 to 1, Figure 9.6.

In addition to the aforementioned simple curves, the coupler curve of a four-bar linkage can also be considered as defining a geometric constraint.

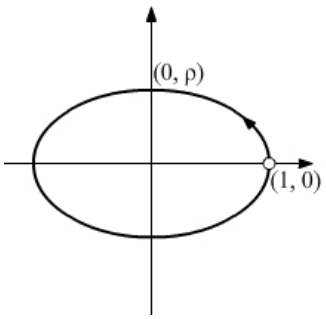


Figure 9.6: An counterclockwise traced ellipse with semi-minor axis of ρ.

9.6 Trajectory Analysis and Constraint Identification

For an explicitly given motion, whether continuously as a function of time or discretely as an ordered sequence of displacements, the goal is to identify a point on the moving body such that its trajectory best matches with one of the predefined geometric constraints such as a circle, a circular arc, a line-segment, an ellipse or even a four-bar coupler curve. This done by analyzing the trajectory of all points within a feasible zone using Fourier transform derived from different curve signatures, and then search in the frequency domain to see if any of the trajectory matches a specified geometric constraint registered by the same type of Fourier descriptor. In this paper, a direct search method is used to compute the error between a given constraint and a selected trajectory. The error is defined in terms of squares of their respective Fourier descriptors, as shown in Eq. (9.10). All errors can be plotted into a colored map. To set up an error map, we select an designer interested region from the moving rigid body. Secondly, we generate a grid of points on that region. Thirdly, we generate a trajectory of every sampled point. Thereafter, we compute

the shape errors between each of the trajectory and the given constraint curve. In the end, a linear color interpolation is applied to the designer interested region in order to visualize the scaled relative shape errors. This color map is called the *error map* for one give motion and one type of constraint curve.

The structural error of two curves are subjective to the choice of the Fourier descriptor. It is interesting to compare the effect of different Fourier descriptors on results of the geometric constraint retrieval for the same given motion. The contour line on the error map that joints points of equal structural error shows the distribution of the structural errors or the similarity between the trajectories to the desired geometric constraint. In this paper, we compare the four types of Fourier descriptors by applying them to identify the four geometric constraints described in Section 9.5.

Constraint Retrieval Procedures

1. The closed task motion is explicitly represented in any format, like rational B-Spline motion, or matrix expressed motion, or any other designer favorite way.

2. Specify an interested observing region, where is the desirable area to impose the geometric constraints.

3. Select the type of Fourier descriptor. Different FDs lead to various retrieval results.

4. Sample the designer interested region, and compute their trajectory. Dense

sampling is recommended for large area but should not exceed the computational capability.

5. Query every type of geometric constraints, and set up the their error maps. These error maps depend on the input motion, so they are built dynamically, and serve as evaluation guidance to match trajectories and geometric constraints.

6. The designer looks up the error maps and error contour, navigates points of the rigid body, and examines the shape, the signature plotting of the trajectories. With the comprehensive references, the designer picks the desired trajectory, the trace to impose the geometric constraint.

7. Repeat the steps 3 to 6 to identify more constraints and compare the effect of using different Fourier descriptors.

9.7 Case Study

In this section, we present a simple example to illustrate and compare the effect of using different Fourier Descriptors to identify geometric constraints for a B-Spline motion.

The example rigid-body motion is presented as a uniform ratiopnal cubic C^2 B-spline periodic motion. Table 9.1 gives the Cartesian parameters for the 16 positions to interpolate. Without loss of generality, the interval of the knot sequence is set to be equal. The motion is show in Figure 9.7. We select the rectangular observing region, the shaded rectangle in Figure 9.7, identified by the diagonal ending at

points $(-6, -3)$ and $(-6, -3)$, 100 by 100 Points are sampled and their trajectories are calculated. All sample traces are closed continuous curves due to the form of the given motion, which allows their representations in terms of signature functions as well as the corresponding Fourier descriptors.

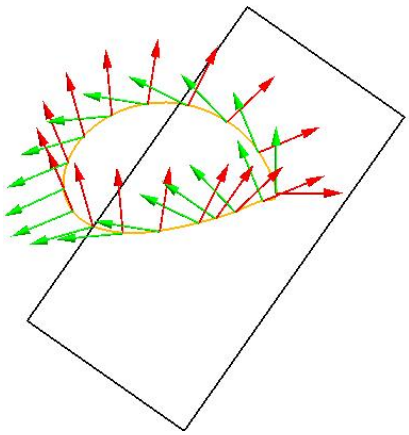


Figure 9.7: The given motion

In this example, we seek to extract four geometric constraints, in the terms of the shape, including circle, circular arc, ellipse and line-segment as introduced in Section 9.5. Especially, the center angle of the circular arc is $2rad$ and the minor semi-axis of the ellipse is of 0.6. For each shape and every FD representation, we render the corresponding error maps. As shown in Figure 9.9 through 10.1, the error map provides a visualization for the quality of the constraint matching indicated by a specific type of FD. For each error map, the error scales are interpolated from minimum through maximum with the color bar as shown in Figure 9.8, where yellow (left) side corresponds to minimum error of that map while the pink (right) side stands for a large shape distinction.

i	X	Y	α
1	3.4788	0.7772	0
2	2.9193	2.0464	0.3902
3	1.9396	2.9964	0.7654
4	0.7202	3.5061	1.1111
5	-0.5380	3.5524	1.4142
6	-1.6592	3.1947	1.6629
7	-2.5144	2.5447	1.8478
8	-3.0228	1.7392	1.9616
9	-3.1458	0.9194	2
10	-2.8788	0.2153	1.9616
11	-2.2482	-0.2711	1.8478
12	-1.3115	-0.4844	1.6629
13	-0.1616	-0.4298	1.4142
14	1.0680	-0.1730	1.1111
15	2.2057	0.1805	0.7654
16	3.0634	0.5225	0.3902

Table 9.1: 16 Control Positions

Figure 9.8: Error reference bar

Obviously, the smaller the error is, the more precisely the geometric constraint matches the trace of that point. Moreover, the error maps intrinsically appear to be the gradient fields of errors, and it is the prominent aspect of the error maps that they also imply the better searching area, if the currently selected region dose not cover any reasonable candidate geometric constraints. For example, if we initially observe only the left-lower area instead of the entire current region, then there is no pleasant point satisfying a circle, see Figure 9.9-(a) to (d), but fortunately, the colorful gradient field clearly advises the search of the right-upper area for a bet-

ter circular point constraint. In addition to error map, signature plot provides an alternative indicator to dynamically observe the matching between the signature of the selected trial trajectory and that of the constraint curve. When two shapes are similar, their signature plots reasonably match.

For the interest of this paper, four types of Fourier descriptors are compared for constraint retrieval. Experiments show all of them lead to consistent results. However, the complex coordinate function and the centroid distance function are more sensitive to the shape difference compared to the curvature function and the cumulative angle. As illustrated by Figure 9.9 through 10.1, the former two FDs always generate error maps of more contrast, while the retrieval results recommended by the later two FDs are quite ambiguous which label a great portion of points with yellow and view their trace as similar shapes.

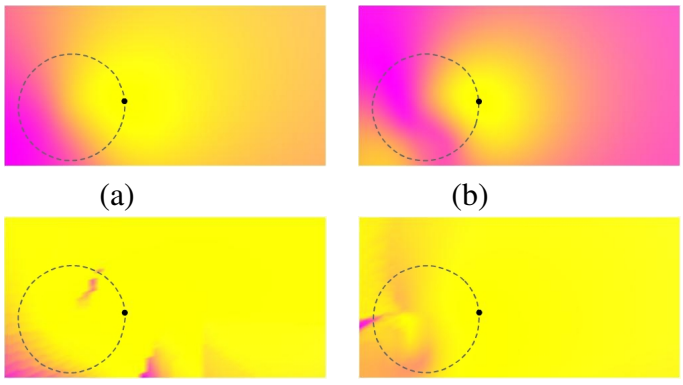


Figure 9.9:
The error map for the point constraint of a circle: Yellow points have smaller error than the pink ones for approximation of a circle. for this map, the trajectory of each point is approximated using the three lowest-frequency FDs.
(a): FD derived based on position function
(b): FD derived based on centroid distance
(c): FD derived based on curvature signature
(d): FD derived based on cumulative angle

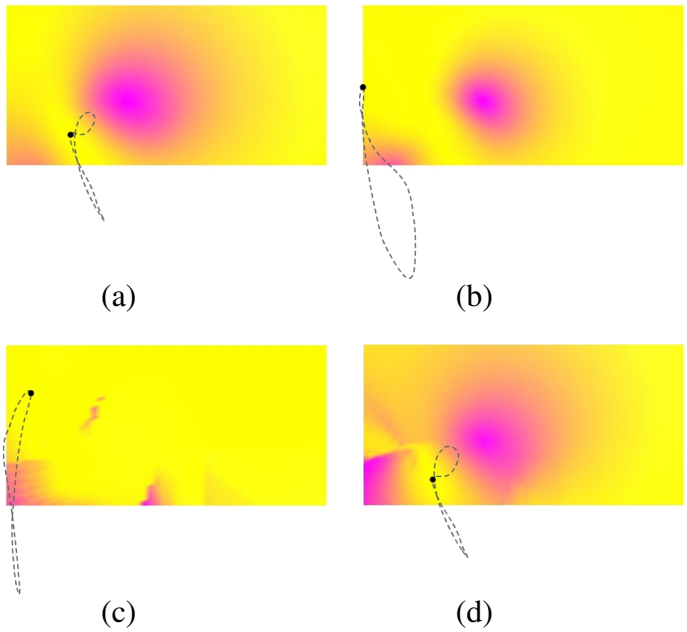


Figure 9.10:
The error map for the point constraint of a circular arc with center angle of $2rad$: Yellow points have smaller error than the pink ones for approximation of an arc. for this map, the trajectory of each point is approximated using the three lowest-frequency FDs.
(a): FD derived based on position function
(b): FD derived based on centroid distance
(c): FD derived based on curvature signature
(d): FD derived based on cumulative angle

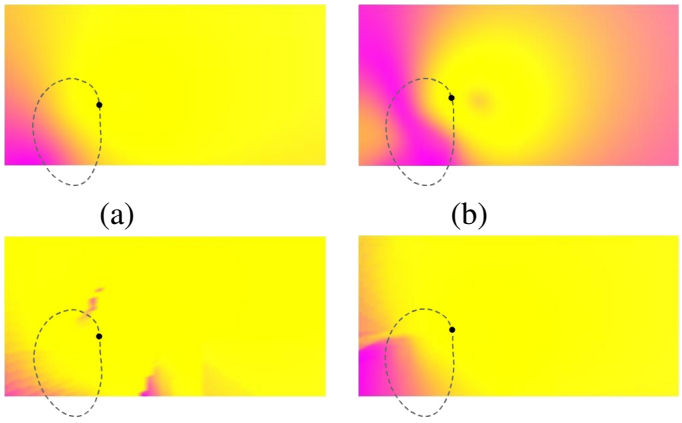


Figure 9.11:
The error map for the point constraint of an ellipse with minor semi-axis of 0.6: Yellow points have smaller error than the pink ones for approximation of an ellipse. for this map, the trajectory of each point is approximated using the three lowest-frequency FDs.
(a): FD derived based on position function
(b): FD derived based on centroid distance
(c): FD derived based on curvature signature
(d): FD derived based on cumulative angle

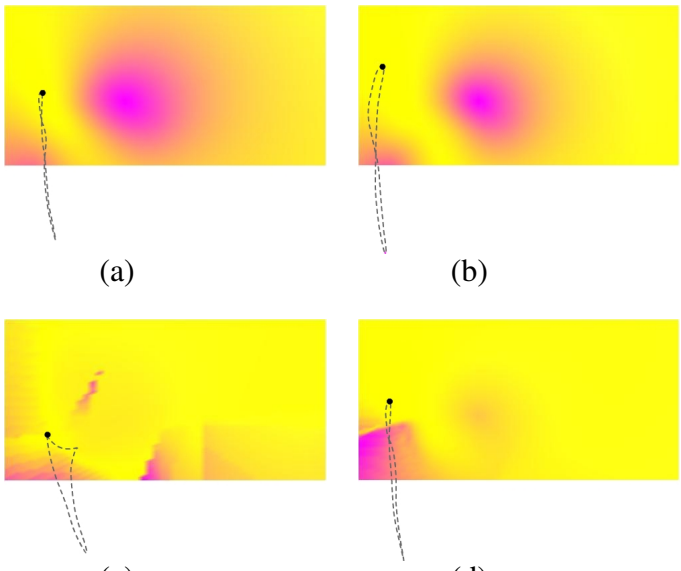


Figure 9.12:
The error map for the point constraint of a line-segment: Yellow points have smaller error than the pink ones for approximation of a line-segment. for this map, the trajectory of each point is approximated using the three lowest-frequency FDs.
(a): FD derived based on position function
(b): FD derived based on centroid distance
(c): FD derived based on curvature signature
(d): FD derived based on cumulative angle

Chapter 10

Kinematic Acquisition of Geometric Constraints for Task Centered Mechanism Design: Line Constraints

Points and lines are the two primitive categories of planar geometry. Both planar point geometry and line geometry share similar properties due to their duality. In chapter 7, we have discussed the kinematic acquisition for point geometric constraints, and this chapter will extend the kinematic acquisition to line geometry by taking advantage of the point-and-line duality.

10.1 Moving Line Trajectory

Similar to the representation of a planar displacement (d_x, d_y, α) of a rigid body by a homogeneous transform of the points thereon as shown in Eq. (7.1), such a displacement can be alternatively reflected by the homogeneous transform of the lines amounted on that rigid body. A line in the moving plane **M** can be represented by $\mathbf{l} = (l_1, l_2, l_3)$, where $(\frac{l_1}{\sqrt{l_1^2+l_2^2}}, \frac{l_2}{\sqrt{l_1^2+l_2^2}})$ is the unit vector perpendicular to the line and pointing from the origin to the line, and $\frac{l_3}{\sqrt{l_1^2+l_2^2}}$ is the perpendicular distance from the origin to the line; it is favorable to represent the normalized line coordinate as $(\cos\beta, \sin\beta, -l)$, where β is the angle between the horizontal

axis and the perpendicular unit vector and l gives the distance. When a specific line $\mathbf{l}$ of $\mathbf{M}$ is required to stay on a line construct $f(\mathbf{L}) = 0$ in $\mathbf{F}$, a line constraint, $f([\bar{H}(d_x, d_y, \alpha)]\mathbf{l}) = 0$ is imposed, where (see [113])

$$\mathbf{L} = [\bar{H}(d_x, d_y, \alpha)]\mathbf{l} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ \bar{d}_x & \bar{d}_y & 1 \end{bmatrix} \mathbf{l} \tag{10.1}$$

with $\bar{d}_x = -d_x \cos\alpha - d_y \sin\alpha, \quad \bar{d}_y = d_x \sin\alpha - d_y \cos\alpha$. For example, a line may be required to pass through a fixed point, to make a certain angle with a given line, or to be tangent to a given curve such as a circle. The homogeneous transform matrix of line, $[\bar{H}(d_x, d_y, \alpha)]$, is inverse-transpose to the homogeneous transform matrix of point, $[H(d_x, d_y, \alpha)]$ in Eq. (7.1):

$$[\bar{H}(d_x, d_y, \alpha)] = [H(d_x, d_y, \alpha)]^{-T} \tag{10.2}$$

Now consider the following linear equation:

$$V_1 L_1 + V_2 L_2 + V_3 L_3 = 0. \tag{10.3}$$

This equation can either interpreted either as constraining a point $\mathbf{V}$ on a line $\mathbf{L}$ or as constraining a line $\mathbf{L}$ such that it passes through a point $\mathbf{V}$. The role of point coordinates $\mathbf{V}$ and line coordinates $\mathbf{L}$ are completely symmetric. This leads to a general principle, known as duality, in projective plane geometry, i.e., geometric transformations that replace points by lines and lines by points while preserving incidence properties among the transformed objects. This means that any theorem about incidences between points and lines in the projective plane may be transformed into another theorem about lines and points, by a substitution of the appropriate words

[80]. Because of this, in this paper, we restrict our study to point-geometric constraints only.

The planar motion of a straight line can be described as a one-parameter family of lines given implicitly by:

$$F(t, (V_1, V_2, V_3)) = V_1 L_1(t) + V_2 L_2(t) + V_3 L_3(t) = 0. \tag{10.4}$$

where $(L_1(t), L_2(t), L_3(t))$ give the homogeneous coordinates of the moving line at time t and (V_1, V_2, V_3) are the points on the line at that moment. The trajectory generated by the line family is a curve, referred as the *envelope* of this line family, that touches every member tangentially and the moving line is so called as the generator of the envelope curve. Hence, a continuous curve can be view as generated by the tangent lines, and vice verse, this continuous curve is the envelope of these tangents.

It is straightforward to recover the motion of the tangent line from a continuous curve in the form of line homogenous coordinate. Say $\mathbf{V}(t)$ is a point on the curve ε at time t and $\mathbf{L}$ is the tangent line at this moment to determine. The tangent line satisfies two conditions: (a) $\mathbf{V}(t)$ is on $\mathbf{L}$ as introduced by Eq. (10.4). (b) in the homogenous coordinates the tangent line, (L_1, L_2) is a vector perpendicular to the tangent vector, $(V_x'(t), V_y'(t))$, of the curve at time t. Hence we have

$$(\frac{V_1(t)}{V_3(t)})' L_1 + (\frac{V_2(t)}{V_3(t)})' L_2 = 0 \tag{10.5}$$

which is equivalent to

$$\frac{dV_1(t)}{dt} L_1 + \frac{dV_2(t)}{dt} L_2 + \frac{dV_3(t)}{dt} L_3 = 0 \tag{10.6}$$

after some algebra and with the consideration of Eq. (10.4). Hereafter, the motion of the line can be explicitly represented as

$$\mathfrak{F} = \{\mathbf{L}(t) | \mathbf{V}(t) \cdot \mathbf{L} = 0 \& \mathbf{V}'(t) \cdot \mathbf{L} = 0\} \tag{10.7}$$

Up till now, we can conclude that the line motion constrained by a continuous curve is unique.

Now we consider the envelope traced by the line motion. In this case, we compute the tangent point $\mathbf{V}$ based on the line motion $\mathbf{L}(t)$. The line envelope can be interpreted as the collection of the intersections of nearby lines $F(t, (V_1, V_2, V_3))$ at all time t. The intersection at time t is the point at which the tangent line contacts the envelope at that moment. This point on the envelope is obtainable by simultaneously solving Eq. (10.4), which indicates $\mathbf{L}(t)$ passes the point $\mathbf{V}$, together with the following constraint

$$\frac{\partial F}{\partial t} = V_1 \frac{dL_1(t)}{dt} + V_2 \frac{dL_2(t)}{dt} + V_3 \frac{dL_3(t)}{dt} = 0 \tag{10.8}$$

To sum up, based on the motion of its straight line, the line trajectory can be uniquely determined as its envelope ε as described below.

$$\varepsilon = \{\mathbf{V}(t) | \mathbf{V} \cdot \mathbf{L}(t) = 0 \& \mathbf{V} \cdot \mathbf{L}'(t) = 0\} \tag{10.9}$$

Readers please refer to [???] for the derivation of Eq. (10.8) as well as the details of the general theory of envelopes from the perspective of differential geometry.

In this chapter, we explain connection between be the line motion and the envelope curve from the viewpoint of duality transformation. Consider a curve $C_{\mathbf{V}}$

traced by point $\mathbf{V}(t)$ and ruled by the associated tangent $\mathbf{L}(t)$. After duality transform, $\mathbf{V}(t)$ becomes a moving line denoted by $\tilde{\mathbf{V}}(t)$, while the $\mathbf{L}(t)$ turns to be a point $\tilde{\mathbf{L}}(t)$ to trace the curve, say $C_{\mathbf{L}}$. $\mathbf{V}(t)$ and $\tilde{\mathbf{V}}(t)$ share the same homogeneous coordinates but are dual geometric elements: one is a line, and the other is a point. The same to $\mathbf{L}(t)$ and $\tilde{\mathbf{L}}(t)$. It can be proved that after duality transformation, the line $\tilde{\mathbf{V}}(t)$ is tangent to curve $C_{\mathbf{L}}$ at point $\tilde{\mathbf{L}}(t)$. Simply speaking, the pair of a tangent line and a tangent point of one curve swaps to the pair of a tangent point and a tangent line of another curve by the duality transformation between point and line. Therefore, the problem to find the envelope $\mathbf{V}(t)$ for the line motion $\mathbf{L}(t)$ can be converted to seek for the tangent line $\tilde{\mathbf{V}}(t)$ of the curve $\tilde{\mathbf{L}}(t)$. The straightforward answer Eq. (10.7) to the later problem explain Eq. (10.9) in a comprehensive way: (a) the tangent line $\tilde{\mathbf{V}}(t)$ passes the point $\tilde{\mathbf{L}}(t)$,

$$\tilde{\mathbf{L}}(t) \cdot \tilde{\mathbf{V}} = 0 \tag{10.10}$$

This gives rise to the first condition in Eq. (10.9). (b) in the homogenous coordinates of the line $\tilde{\mathbf{V}}$, (V_1, V_2) is a vector perpendicular to the tangent vector of at point $\tilde{\mathbf{L}}(t)$, $((L_1/L_3)', (L_2/L_3)')$

$$\tilde{\mathbf{L}}'(t) \cdot \tilde{\mathbf{V}} = 0 \tag{10.11}$$

which is Eq. (10.8) and the section condition of Eq. (10.9). Essentially, the envelope conditions in Eq. (10.9) are the dual form of the tangent motion conditions in Eq. (10.7).

In view of Eq. (10.7) and (10.9), the mutual generation between the line motion and the envelope curve is bijective and unambiguous, in the sense that one line motion results a unique envelope curve and this curve recovers the same lime motion.

The converse statement is also true: a curve can set up a unique line motion and this motion duplicates the same curve as its envelope. Consequentially, the line motion and the envelope curve is equivalent; the representation of a line motion can be replaced by the envelope curve, and vise verse, the continuous curve can be expressed by a line motion.

10.2 Explicit Description of Geometric Constraints

In this paper, we restrict ourselves to line-geometric constraints, i.e., a specific line on the moving body is required to roll on a given curve. In order to focus on trajectory of the line or, in the other words, the shape of the curve, we seek to develop a representation of a line trajectory who generates the curve that is invariant with respect to rigid-body transformation in the plane as well as scaling.

Recall the linear transformation, see Eq.(7.2), in the plane that preserves the shape but not the scale of a curve which sequentially undergoes a rotation for angle ξ, a scaling for factor λ and a translation (c_x, c_y). There exists the associated transform for the line trajectory. Let $\mathbf{g} = (\cos\eta, \sin\eta, g)$ and $\mathbf{G} = (G_1, G_2, G_3)$ denote the line trajectory and its new configuration after an equiform displacement (in Figure 7.2):

$$\mathbf{G} = [\bar{E}]\mathbf{g} = \begin{bmatrix} \cos\xi & -\sin\xi & 0 \\ \sin\xi & \cos\xi & 0 \\ \bar{D}_x & \bar{D}_y & \kappa \end{bmatrix} \mathbf{g} \tag{10.12}$$

where $\bar{D}_x = -c_x\cos\xi - c_y\sin\xi$, $\bar{D}_y = c_x\sin\xi - c_y\cos\xi$ and $\kappa = 1/\lambda$. Thus, an equiform displacement has four degrees of freedom and are defined by the parameters $(\xi, \kappa, \bar{D}_x, \bar{D}_y)$.

The main task in kinematic acquisition for line constraints is to identify a line $\mathbf{l}$ $(\cos\beta, \sin\beta, l)$ in the moving body such that its trajectory $\mathbf{L}$ best approximates one of the constraint curves in a library of "standard constraint line motion". Another factor that affects this constraint matching process is the parameterizations of the constraint curve and the trajectory $\mathbf{L}$. Ideally, an intrinsic parametrization for each of the line can be carried out to remove the influence of parametrization in the shape matching process. For example, One might like to convert the line trajectories to the envelope curves, hence the arc length for each of the envelopes can be carried out to remove the influence of parametrization in the shape matching process, and thereafter, computational comparison of either the envelopes or the envelope arc-length parameterized line trajectory leads a solution to the constraint retrieval. However, in this chapter, we take a computationally less expensive approach of normalizing the parameterizations of these closed line trajectories so that they are in the range of $[0, 1]$. Since the periods of these trajectories are made equal to one, we have

$$\mathbf{g}(t) = \mathbf{g}(t-1), \quad \mathbf{G}(t) = \mathbf{G}(t-1), \quad \mathbf{L}(t) = \mathbf{L}(t-1) \tag{10.13}$$

where $t =\in [0, 1]$. Another factor that affects the constraint matching process is the choice of initial line on each of these trajectories. To address this issue, we provide a shift factor t_Δ so that the initial line on the constraint line trajectory can vary in order to find a better match. Incorporating the shift factor, we can define the normalized parameters, t_G and t_L, for the constraint curve $\mathbf{G}(t_G)$ and a point trajectory of the given motion, $\mathbf{L}(t_L)$ as

$$t_L = t, \qquad t_G = t + t_\Delta \tag{10.14}$$

wherein $t \in [0, 1]$ and the shift parameter t_Δ indicates how much the motion-shape correspondence has to be shifted. As a result, $\mathbf{L}(t_L)$ on $[0, 1]$ match $\mathbf{G}(t_G)$ on $[t_\Delta, 1 + t_\Delta]$.

10.3 Least-Square Approximation for Constraint Identification

This section presents an algorithm for identifying a given constraint from line trajectories of a specified motion task. This work may be considered as an extension of an earlier work on kinematic acquisition of point trajectories introduced in chapter 7.

We assume that the motion task has been given explicitly as an ordered sequence of N discrete positions and a set of N ordered lines belong to a given constraint curve has also been given. Consequently, instead of using continuous parameter t_V and t_G as given by Eq. (10.14), we use indexes i ($i = 1, \cdots, N$) to represent the sequence of lines and positions, where the shift parameter t_Δ becomes an integer n, which means that the starting line on the geometric constraint line set is shifted by n lines. Thus the squared distance between the ith line on the trajectory $\mathbf{L}$ and the $(i + n)$th line on the geometric constraint is given by

$$\delta_{i,n}^2 = (\mathbf{L}_i - \mathbf{G}_{i+n})^T (\mathbf{L}_i - \mathbf{G}_{i+n}) \tag{10.15}$$

where $|\cdot|$ denotes the magnitude of a vector.

In view of Eq. (7.1) and (7.2), we have

$$\mathbf{L}_i = [\bar{H}_i]\mathbf{l}, \quad \mathbf{G}_{i+n} = [\bar{E}]\mathbf{g}_{i+n} \tag{10.16}$$

where $[\bar{H}_i]$ is obtained from the 3×3 matrix in (7.1) by replacing α, d_x, d_y with $\alpha_i, d_{x,i}, d_{y,i}$.

Substituting (10.16) into (10.15), after some algebra, we obtain

$$\delta_{i,n}^2 = (\mathbf{A}_{i,n}^T \mathbf{X} - [d_{x,i}, d_{y,i}][\cos\beta, \sin\beta]^T)^2 + 2 - 2\cos(\beta + \alpha_i - \xi - \eta_{i+n}) \quad (10.17)$$

where

$$\mathbf{A}_{i,n}^T \quad = \quad [\quad \cos\eta_{i+n}, \quad \sin\eta_{i+n}, \quad g_{i+n}, \quad -1 \quad] \quad (10.18)$$

are determined from the specified motion and the given geometric constraint, and

$$\mathbf{X} = [\bar{D}_x, \bar{D}_y, \kappa, l]^T \quad (10.19)$$

are the design variables associated with the equiform displacement and choice of the line $\mathbf{l} = (\cos\beta, \sin\beta, l)$ on the moving body. The sum of squares of the deviations is thus given by (Eq. (7.7)):

$$S = \frac{1}{N}\sum_{i=1}^{N} \delta_{i,n}^2 = \frac{1}{N}\sum_{i=1}^{N} \{(\mathbf{A}_{i,n}^T \mathbf{X} - [d_{x,i}, d_{y,i}][\cos\beta, \sin\beta]^T)^2 + 2 - 2\cos(\beta + \alpha_i - \xi - \eta_{i+n})\}. \quad (10.20)$$

The goal for constraint acquisition is to find a line $\mathbf{l} = (\cos\beta, \cos\beta, l)$ on the moving body such that the trajectory of $\mathbf{l}$ best approximates the given line trajectory constraint after an appropriate equiform displacement, $(\xi, \kappa, \bar{D}_x, \bar{D}_y)$. Totally, we have six variables, $(\beta, l, \xi, \kappa, \bar{D}_x, \bar{D}_y)$, to determine.

The least square solution to (10.20) satisfies the following conditions:

$$\frac{\partial S}{\partial \bar{D}_x} = 0, \quad \frac{\partial S}{\partial \bar{D}_y} = 0, \quad \frac{\partial S}{\partial \kappa} = 0, \quad \frac{\partial S}{\partial l} = 0, \quad \frac{\partial S}{\partial \xi} = 0, \quad \frac{\partial S}{\partial \beta} = 0 \quad (10.21)$$

The first four conditions give

$$\left[\sum_{i=1}^{N}\{\mathbf{A}_{i,n}\mathbf{A}_{i,n}^{T}\}\right]\mathbf{X}=\left[\sum_{i=1}^{N}\{\mathbf{A}_{i,n}[d_{x,i},d_{y,i}]\}\right]\begin{bmatrix}\cos\beta\\ \sin\beta\end{bmatrix}. \tag{10.22}$$

which allows $\mathbf{X}$ to be expressed in terms of α.

The fifth condition leads to

$$\sum_{i=1}^{N}\sin(\beta+\alpha_i-\xi-\eta_{i+n})=0 \tag{10.23}$$

which exclusively contains only β and ξ.

The last condition yields

$$[\cos\beta,\sin\beta](\left[\sum_{i=1}^{N}\{\begin{bmatrix}d_{y,i}\\ -d_{x,i}\end{bmatrix}\mathbf{A}_{i,n}^{T}\}\right]\mathbf{X}-\left[\sum_{i=1}^{N}\{\begin{bmatrix}d_{y,i}\\ -d_{x,i}\end{bmatrix}[d_{x,i},d_{y,i}]\}\right]\begin{bmatrix}\cos\beta\\ \sin\beta\end{bmatrix})$$
$$+\sum_{i=1}^{N}\sin(\beta+\alpha_i-\xi-\eta_{i+n})=0 \tag{10.24}$$

The above equation can be simplified by substituting (10.22) and (10.23) into it. Finally it is reduced to an equation including solely α:

$$[\cos\beta,\sin\beta]M[\cos\beta,\sin\beta]^{T}=0 \tag{10.25}$$

where

$$M=\left[\sum_{i=1}^{N}\{\begin{bmatrix}d_{y,i}\\ -d_{x,i}\end{bmatrix}\mathbf{A}_{i,n}^{T}\}\right]\left[\sum_{i=1}^{N}\{\mathbf{A}_{i,n}\mathbf{A}_{i,n}^{T}\}\right]^{T}\left[\sum_{i=1}^{N}\{\mathbf{A}_{i,n}[d_{x,i},d_{y,i}]\}\right]$$
$$-\left[\sum_{i=1}^{N}\{\begin{bmatrix}d_{y,i}\\ -d_{x,i}\end{bmatrix}[d_{x,i},d_{y,i}]\}\right] \tag{10.26}$$

Therefore, β is obtained by solving Eq (10.25), then ξ and $\mathbf{X}$ become available by solving Eq (10.22) and (10.23), respectively, with the known value of β.

Up to this point, it has been assumed that the shift parameter n is known, and the least squares solution can be obtained by solving a set of equations. However, in general, the desired value of n is unknown, and as aforementioned, the order of the positions sampled from the motion and that of the lines on the constraint trajectory are not automatically and optimally correlated. We now introduce a numerical algorithm for shape matching which combines least squares optimization with a direct search dealing with the shift parameter n, as stated below:

Objective: Given a discrete motion M with N sampled positions and a geometric constraint line trajectory $\mathbf{g}$ of N sampled points, find a point on the moving body $\mathbf{l} = (\cos\beta, \sin\beta)$, such that after the constraint line trajectory $\mathbf{g}$ is transformed into to a new configuration $\mathbf{G}$ via an equiform displacement, $(\xi, \kappa, \bar{D}_x, \bar{D}_y)$, the error between the trajectory $\mathbf{L}$ and the constraint line trajectory $\mathbf{G}$ as defined by Eq. (10.20) is at minimum.

Algorithm

For each $n \in [1, \cdots, N]$ **do**

For each $i \in [1, \cdots, N]$ **do**

- Initialize values of $\mathbf{A}_{i,n}, d_{x,i}, d_{y,i}$;
- Substitute $\mathbf{A}_{i,n}, d_{x,i}, d_{y,i}$ into Eq. (10.22), (10.23) and (10.25);

End for

- Solve Eq. (10.25) for β;
- Substitute β into Eq. (10.23) and solve for ξ;
- Substitute β and ξ into Eq. (10.22) and solve for $\mathbf{X}$;

If the total error is currently at minimum **do**

- Update final solution to $\{\beta, l, \xi, \kappa, \bar{D}_x, \bar{D}_y, n\}$;

End if

End for

10.4 Tangent Lines to Conics

The intersecting curves of a circular cone with a plane are referred as *conics*. On cartesian plane, a conic curve can be implicitly described by the quadric algebraic equation:

$$aV_x^2 + bV_xV_y + cV_y^2 + dV_x + eV_x + f = 0 \tag{10.27}$$

Bashelor provided a comprehensive enumeration of the conics by using the *moduli space*[??]. Basically conics can be classified into two groups: nondegenerate and degenerate conics. As the polynomial defining the conic is irreducible, the conic is said to be *nondegenerate*. Such conics include circles, ellipses, hyperbolas, and parabolas. If the polynomial defining the conic factors into a product of linear polynomials, then the conic is just the union of two lines and said to be *degenerate*.

The conic curve can also expressed in homogenous matrix form

$$\mathbf{V}^T\mathbf{C}\mathbf{V} = 0 \text{ with } \mathbf{C} = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix} \tag{10.28}$$

where **C** is called *the matrix of the conic section*. The matrix **C** is nonsingular for the nondegenerate conics. A conic section is homogenously described by the six coefficients (a, b, c, d, e, f), or in the other words, $(\alpha a, \alpha b, \alpha c, \alpha d, \alpha e, \alpha f)$ depict the same conic when $\alpha \neq 0$. Thus, **C** is also homogeneous, since $\alpha\mathbf{C}$ represents the same conic when $\alpha \neq 0$.

The tangent line to the conic at point **V** can be explicitly represented by

$$\mathbf{L} = \mathbf{C}\mathbf{V} \tag{10.29}$$

It is easy to verify that the above line coordinates satisfy the conditions Eq (10.7) for such a line motion that is tangentially constrained by the conic curve. The line $\mathbf{L}$ tangent to the conic $\mathbf{C}$ necessarily satisfies the following condition:

$$\mathbf{L}^T\mathbf{C}^*\mathbf{L} = 0 \tag{10.30}$$

For nondegenerate conics, $\mathbf{C}^* = \mathbf{C}^{-1}$. $\mathbf{C}^*$ is symmetric, and the conic described by $\mathbf{C}^*$ is said to be the dual conic of $\mathbf{C}$. Apparently, Eq (10.30) indicates the tangent line $\mathbf{L}$ is on the dual conic $\mathbf{C}^*$. The tangent line conic is dual to the envelop conic, and conversely the conic is dual to the conic of its tangent generator.

10.5 Implicit Description and Retrieval of Geometric Constraints

Alternatively by taking advantage of duality between conics and tangent conics, the representation of the line motion constrained by conics and invariant to rigid-body transformation in the plane as well as scaling can also be expressed in implicit form. If the given curve is a conic $\mathbf{C}$ and the equiform transform is $E(\xi, \lambda, c_x, c_y)$. After the equaifrom transform, the conic becomes

$$\mathbf{C}' = E^{-T}\mathbf{C}E^{-1} \tag{10.31}$$

and the dual conic for the tangent lines is

$$\mathbf{C}'^* = E\mathbf{C}^*E^T \tag{10.32}$$

All the tangent lines of new configuration satisfy

$$\mathbf{L}^T\mathbf{C}'^*\mathbf{L} = 0 \tag{10.33}$$

In kinematic acquisition to identify a line $\mathbf{l}(\cos\beta, \sin\beta, l)$ on the moving body whose trajectory $\mathbf{L}$ best approximate the line trajectory constrained by $\mathbf{C}$, it is essential to find the $\mathbf{l}$ and E that equalize Eq. (10.33). Because the line constraint $\mathbf{g}$ and its configuration $\mathbf{G}$ after the equiform transform are implicit represented by $\mathbf{C}^*$ and $\mathbf{C}'^*$ respectively, neither the parametrization nor the initial line is not involved in the matching process.

When the motion task is given explicitly as an ordered sequence of N discrete positions, let the N lines be index by i $(i = 1, 2, \cdots, N)$. The structural error for the implicit conic constraint is given by

$$S = \sum_{i=1}^{N} (\mathbf{L}_i^T \mathbf{C}'^* \mathbf{L}_i)^2 \tag{10.34}$$

Substitute $\mathbf{L}_i = [\bar{H}_i]\mathbf{l}$ and Eq. (10.32) into the above equation, and we have

$$S = \sum_{i=1}^{N} (\mathbf{l}^T [\bar{H}_i]^T [E\mathbf{C}^* E^T][\bar{H}_i]\mathbf{l})^2 \tag{10.35}$$

To identify the line constraint for the given motion to to find $\beta, l, \xi, \lambda, c_x, c_y$ to minimize the structural error. However, the structural error function is a quadric polynomial of $\cos\beta, \sin\beta, l, \cos\xi, \sin\xi, c_x, c_y$ and λ. To handle this optimization, we can resort to the search on $\beta, l, \xi, \lambda, c_x, c_y$. Since this search is conducted in six-dimensional space, a more intelligent search method such as the simulated annealing search is recommended.

10.6 From Geometric Constraints to Mechanisms

To design a mechanical device to generate the motion featured by line geometric constraints, we follow the same principle that applied to point geometric constrained motion to realize the geometric constraints using kinematic chains.

Figure?? enumerates some 1-dof mechanisms and classify them based on the geometric constraints they provide. A comprehensive treatise on the generation of the tangents to planar algebraic curves using planar mechanisms can be found in [117].

In what follows we present an example to illustrate the effectiveness of the constraint recognition scheme. We then show that the result leads naturally to task-centered design of mechanisms for the specified task. In this example, the given motion task is artificially designed such that the moving frame carries out two lines: one is tangent to an ellipse and the other is tangent to a circle, see Figure ??. The ellipse has a major axis of length 5 and a minor axis of length 3, and is centralized at point $(-1.5, 1.0)$. The angle between the major axis of the ellipse and the horizontal axis of the reference frame is 30°. The center of the circle is $(2, -3)$ and its radium is 1.5. The rigid body motion is artificially designed such that the x-axis (or line $L_e(0, 1, 0)$) on the moving frame is required to tangentially touch the ellipse at point $P_1^e, P_2^e, \cdots P_n^e$ and meanwhile, another line $L_c(-0.8090, 0.5878, 0)$ on the moving frame is tangent to the circle at points $P_1^c, P_2^c, \cdots P_n^c$. In our example we sampled $n = 63$ tangent points and lines. Since L_e and L_c have to satisfy rigid body condition, once the two constraint curves and P_i^e are determined, and then the P_i^c is accordingly known.

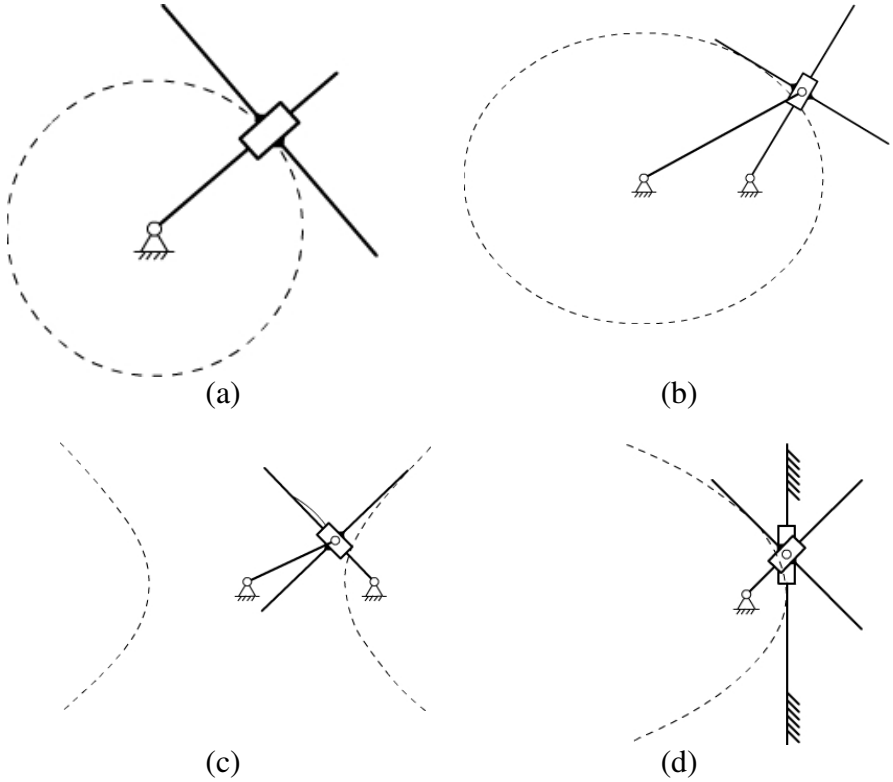


Figure 10.1:
(a): 1-dof mechanism that generates tangent lines of a circle
(b): 1-dof mechanism that generates tangent lines of an ellipse
(c): 1-dof mechanism that generates tangent lines of a hyperbola
(d): 1-dof mechanism that generates tangent lines of a parabola

In the first part of this example, we test our identification algorithm for the explicitly represented constraints. We use the green ellipse in Figure 10.2 as our desired shape. For explicit constraint, the ellipse are given as a sequence of sample points. For the purpose to test the algorithm, sample points on the constraint are identical to those on the ellipse used to construct the example motion. Hence, we expect the line $(0, 1, 0)$ to be identified as tangent moving line to the ellipse. By

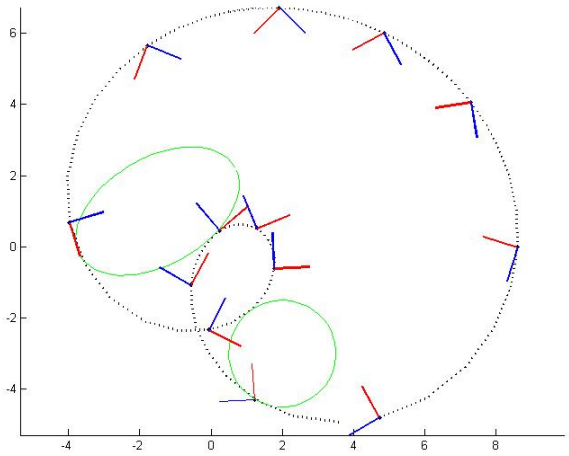


Figure 10.2: The rigid body motion with two lines tangent to one ellipse and one circle, respectively. The red-blue frames are some sample positions of the motion. The origin of the moving frame traces the dash line in black.

run simulated annealing algorithm for 20 times. The we retrieve more than one constraint of the same shape yet of different sizes and locations. Figure 10.3 depicts exactly the same constraint and Figure 10.4 shows another identification result which also preserve the elliptical shape quite well.

In the second part of the example, we examine our identification algorithm for the implicitly represented constraint. Also the ellipse are used for test. Herein, the constraint is given by the matrix of the conic section,

$$\mathbf{C} = \begin{bmatrix} \frac{1}{2.5^2} & 0 & 0 \\ 0 & \frac{1}{1.5^2} & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{10.36}$$

Pleasingly, not only the original constraint is identified, but also by running simulated annealing algorithm for multiple times, we obtain multiple solutions of good quality, see Figure.

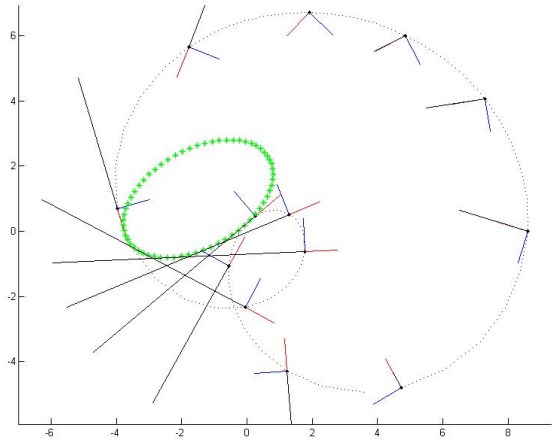


Figure 10.3: The ellipse in green is the constraint identified from the motion, such that it is tangentially touched by the moving line $(0, 1, 0)$. The solid lines in black are some samples of the line motion that is tangent to the identified ellipse.

10.7 Conclusions

In this chapter, we have successfully extend the methods to extract point geometric constraints from a specified motion task to line geometric constraint by resorting to the duality of point and lines. The resulting constraints are then used to identify and synthesize simple mechanisms for constraint generation. This task-centered and constraint based paradigm to mechanism design greatly reduces the complexity of the type synthesis problem.

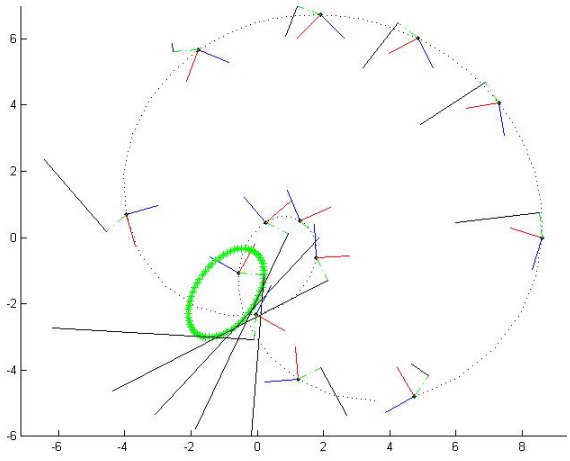


Figure 10.4: The ellipse in green is the constraint identified from the motion, such that it is tangentially touched by the moving line $(-0.4160, 0.9094, 0.7641)$. The solid lines in black are some samples of the line motion that is tangent to the identified ellipse.

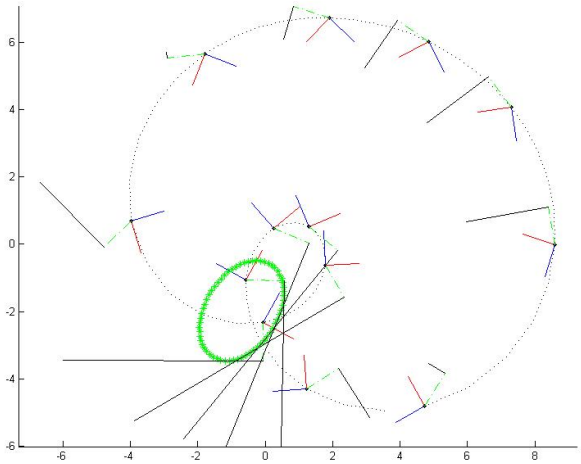


Figure 10.5: The ellipse in green is the constraint identified from the motion, such that it is tangentially touched by the moving line $(-0.4688, 0.8833, 1.1235)$. The solid lines in black are some samples of the line motion that is tangent to the identified ellipse.

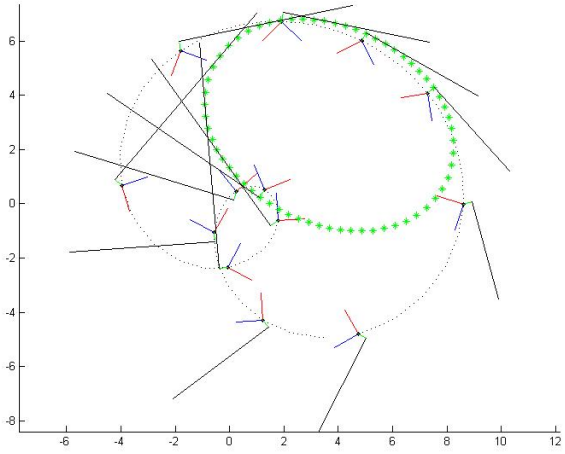


Figure 10.6: The ellipse in green is the constraint identified from the motion, such that it is tangentially touched by the moving line $(0.8385, 0.5449, 0.3279)$. The solid lines in black are some samples of the line motion that is tangent to the identified ellipse.

Chapter 11

Conclusion

This dissertation focuses on the task-centric design problem, and advocates a new geometric constraint based design paradigm, namely "variational kinematic geometry". For the purpose to identify the merits of geometric constraint approach, we conduct a comparative research on both the classical mechanical-centric (see chapter 4 to 6) and new geometric-centric (see chapter 7 to 10) methods. For the mechanical-centric method, we study the synthesis methods for planar 6R closed chain and four-bar mechanism, both of which follow the same design pattern that the type of the mechanism is required to know first and the optimization is used to revise the dimension. In the other words, there is a clear boundary between type synthesis and dimensional synthesis if mechanical-centric approach is applied for mechanism design. We propose the variational kinematic geometry method to deal with the task-centric design from the perspective of geometry. Our method is able to simultaneously synthesize a mechanism both topologically and dimensionally. As shown in chapter 7 to 10), our design approach emphasizes on the geometric feature of the task motion and follows such a procedure that geometric constraints are firstly identified by using shape analysis method, and then multiple design solutions are yielded by choosing diverse feasible mechanisms to generate the geometric con-

straints individually. Our design approach has similarity to ATLAS method in view of both of them are using the geometric information of the task. ATLAS method works in this way: the curves generated by different mechanisms are preprocessed and stored in a library, and to design a mechanism is to look up a curve to match the desired curve and choose the according mechanism. Our method is different from ATLAS method from three ways. Firstly, up to our literature review, existing ATLAS method works for the synthesis of gear 2R chain, gear 5R closed chain, cam and four-bar linkage. Obviously, ATLAS method requires to type of the mechanism to be known. However, our method has no boundary between type and dimension synthesis, and we can with more types than existing ATLAS method can do. Secondly, ATLAS method can be only applied to trajectory tasks, but the geometric method works for motion generation, and we use ATLAS as part of our geometric constraint library to build more complex mechanisms that are not available from ATLAS methods. Thirdly, we use the FD based shape comparison algorithms to retrieve the geometric features. This kind of constraint identification methods are compatible with human vision according to the study by cognitive scientists[107].

Bibliography

[1] Reuleaux, F., 1875. *Theoretische Kinematic*. Fridrich Vieweg, Braunschweig, Germany, (English translation by A.B.W. Kennedy, The kinematics of machinery, 1876, Reprinted Dover, 1963).

[2] Artobolevsky, I. I., 1975. *Mechanisms in Modern Engineering Design, in 2 Volumes (2288 mechanisms)*. Mir Publishers, Moscow.

[3] Hunt, K., 1978. *Kinematic Geometry of Mechanisms*. Oxford University Press.

[4] Phillips, J., 1984. *Freedom in Machinery*, Vol. 1. Cambridge University Press, Cambridge.

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

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

[7] Sandor, G. N., and Erdman, A. G., 1997. *Advanced Mechanism Design: Analysis and Synthesis Vol. 2*. Prentice-Hall, Englewood Cliffs, NJ.

[8] McCarthy, J. M., 1990. *Introduction to Theoretical Kinematics*. MIT.

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

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

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

[12] 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.

[13] Su, H.-J., McCarthy, 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.

[14] Su, H.-J., 2004. "Computer-aided constrained robot design using mechanism synthesis theory". Ph.d. dissertation, University of California, Irvine.

[15] 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.

[16] Wampler, C. W., Morgan, 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.

[17] Krovi, V., Ananthasuresh, 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,* ***123***(3), pp. 359–366.

[18] Su, H.-J., Wampler, C. W., and McCarthy, J. M., 2004. "Geometric design of cylindric prs serial chains". *Journal of Mechanical Design,* ***126***(2), pp. 269–277.

[19] 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.

[20] Lee, E., and Mavroidis, C., 2002. "Solving the geometric design problem of spatial 3r robot manipulators using polynomial homotopy continuation". *Journal of Mechanical Design,* ***124***(4), pp. 652–661.

[21] Mruthyunjaya, T. S., 2003. "Kinematic structure of mechanisms revisited". *Mechanism and Machine Theory,* ***38***(4), pp. 279 – 320.

[22] RUBEL, A. J., and KAUFMAN, R. E., 1977. "Kinsyn iii: A new human-engineered system for interactive computer-aided design of planar linkages". *ASME Journal of Engineering for Industry,* ***99***, p. 440¨C448.

[23] Erdman, A., and Gustafson, J., 1977. Lincages: Linkage interactive computer analysis and graphically enhanced synthesis packages,. Tech. rep.

[24] Erdman, A. G., and Riley, D., 1981. "Computer-aided linkage design using the lincages package". In ASME Design Engineering Technical Conferences, ASME Press. Paper Number 81-DET-121.

[25] Kihonge, J., Vance, J., and Larochelle, P., 2001. "Spatial mechanism design in virtual reality with networking". In ASME 2001 Design Engineering Technical Conferences.

[26] Larochelle, P. M., 1998. "Spades: Software for synthesizing spatial 4c linkages". In ASME Design Engineering Technical Conferences (DETC).

[27] Larochelle, P. M., Dooley, A. P., Murray, A. P., and McCarthy, J. M., 1993. "Sphinx: Software for synthesizing spherical 4r mechanisms". In NSF Design and Manufacturing Systems Conference, p. 607¨C611.

[28] Ruth, D., and McCarthy, J. M., 1997. "Sphinxpc: An implementation of four position synthesis for planar and spherical 4r linkages". In ASME Design Engineering Technical Conferences (DETC).

[29] Tse, D., and Larochelle, P. M., 1999. "Osiris: a new generation spherical and spatial mechanism cad program". In 1999 Florida Conference on Recent Advancements in Robotics.

[30] Perez, A., and McCarthy, J., 2000. "Dimensional synthesis of spatial rr robots". In Advances in Robot Kinematics.

[31] Su, H., and McCarthy, J. M., 2001. "Classification of designs for rrss linkages". In ASME Design Engineering Technical Conferences.

[32] Su, H., Collins, C., and McCarthy, J. M., 2002. "An extensible java applet for spatial linkage synthesis". In 2002 ASME Design Engineering Technical Conferences, ASME.

[33] SyMech. http://www.symech.com/.

[34] WATT. http://www.heron-technologies.com.

[35] Freudenstein, F., M. E., 1979. "The creation of mechanisms according to kinematic structure and functions". *Environ. Plann,* ***6***, pp. 375–391.

[36] Chiou, S. J., and Kota, S., 1999. "Automated conceptual design of mechanisms". *Mechanisms and Machine Theory,* ***34***(3), pp. 467–495.

[37] Yan, H., 1998. *Creative Design of Mechanical Devices*. Springer, Berlin.

[38] Hoeltzel, D., and Chieng, W., 1990. "Knowledge-based approaches for the creative synthesis of mechanisms". *CAD,* ***22***, pp. 57–67.

[39] Yang, B., P. D. U. D. J. K., 1990. "An integrated system for design of mechanisms by an expert system-domes: Theory". *ASME J. Mech. Des.,* ***112***, pp. 488–493.

[40] Vucina, D., and Freudenstein, F., 1991. "An application of graph theory and nonlinear programming to the kinematic synthesis of mechanisms". *Mechanism and Machine Theory,* ***26***(6), pp. 553–563.

[41] Sedlaczek K., Gaugele, T., and Eberhard, P., 2005. "Topology optimized synthesis of planar kinematic rigid body mechanisms". In Multibody Dynamics 2005, ECCOMAS Thematic Conference, J. C. J.M. Goicolea and J. G. Orden, eds.

[42] Frecker, M. I., Ananthasuresh, G. K., Nishiwaki, S., Kikuchi, N., and Kota, S., 1997. "Topological synthesis of compliant mechanisms using multi-criteria optimization". *Journal of Mechanical Design,* ***119***(2), pp. 238–245.

[43] Hayes, M. J. D., and Zsombor-Murrary, P. J., 2004. "Towards integrated type and dimensional synthesis of mechanisms for rigid body guidance". *Proceedings of the CSME Forum 2004*, pp. 53–61.

[44] Faltings, B., 1990. "Qualitative kinematics in mechanisms". *Artificial Intelligence,* ***44***(1-2), pp. 89–119.

[45] Suh, C. H., and Radcliffe, C. W., 1978. *Kinematics and Mechanism Design*. Wiley, New York.

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

[47] Jin, Z., and Ge, Q. J., 2007. "Computer aided synthesis of piecewise rational motion for planar 2r and 3r robot arms". *ASME Journal of Mechanical Design,* ***129***(10), pp. 1031–1036.

[48] Jin, Z., and Ge, Q. J., 2007. "Rational motion interpolation under kinematic constraints of planar 6r closed chain". In ASME 2007 International Design

Engineering Technical Conferences & Computers and Information in Engineering Conference, ASME Press. Paper No. DETC2006-99650.

[49] Purwar, A., Zhe, J., and Ge, Q. J., 2008. "Computer aided synthesis of piecewise rational motions for spherical 2r and 3r robot arms". *ASME Journal of Mechanical Design,* ***130***(11).

[50] Purwar, A., Jin, Z., and Ge, Q. J., 2008. "Rational motion interpolation under kinematic constraints of spherical 6r closed chains". *ASME Journal of Mechanical Design,* ***130***(6), pp. 062301–1 – 062301–9.

[51] Purwar, A., Jin, Z., and Ge, Q. J., 2008. "Computer aided synthesis of rational motion under kinematic constraints of spatial ss open chains". In ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference.

[52] Funtoura Costa, L., and Cesar, R. M., 2001. *Shape Analysis and Classification: Theory and Practice*. CRC Press, Boca Raton, FL.

[53] Ravani, B., and Roth, B., 1984. "Mappings of spatial kinematics". *Journal of Mechanisms Transmissions and Automation in Design-Transactions of the ASME,* ***106***(3), pp. 341–347.

[54] Larochelle, P. M., Murray, A. P., and Angeles, J., 2007. "A distance metric for finite sets of rigid-body displacements via the polar decomposition". *Journal of Mechanical Design,* ***129***(8), pp. 883–886.

[55] Larochelle, P., and McCarthy, J. M., 1995. "Planar motion synthesis using an approximate bi-invariant metric". *Journal of Mechanical Design,* ***117***(4), pp. 646–651.

[56] Tse, D. M., and Larochelle, P. M., 2000. "Approximating spatial locations with spherical orientations for spherical mechanism design". *Journal of Mechanical Design,* ***122***(4), pp. 457–463.

[57] Etzel, K., and McCarthy, J. M., 1996. "Spatial motion interpolation in an image space of so(4)". In Proc. 1996 ASME Mechanisms Conference.

[58] Coxeter, H., 1965. *Non-Euclidean Geometry*. University of Toronto Press, Toronto.

[59] Ge, Q. J., 1998. "On the matrix realization of the theory of biquaternions". *ASME Journal of Mechanical Design,* ***120***(3), pp. 404–407.

[60] Purwar, A., and Ge, Q. J., 2009. "Reconciling distance metric methods for rigid body displacements". In Proceedings of the 2009 ASME IDETC Conferences, ASME, pp. DETC2009–87718, Submitted.

[61] Angeles, J., 2006. "Is there a characteristic length of a rigid-body displacement?". *Mechanism and Machine Theory,* ***41***(8), pp. 884 – 896.

[62] LAROCHELLE, P. M., MURRAY, A. P., and ANGELES, J., 2007. "A distance metric for finite sets of rigid-body displacements via the polar decomposition". *Journal of Mechanical Design,* ***129***(8), pp. 883–886.

[63] Farin, G., 1996. *Curves And Surfaces for Computer-Aided Geometric Design: A Practical Guide*, 4th ed. Academic Press, New York.

[64] Hoschek, J., and Lasser, D., 1993. *Fundamentals of Computer Aided Geometric Design*. A K Peters.

[65] Piegl, L., and Tiller, W., 1995. *The NURBS Book*. Springer, Berlin.

[66] Boggess, A., and Narcowich, F. J., 2001. *A First Course in Wavelets with Fourier Analysis*. Prentice Hall.

[67] Ge, Q. J., Srinivasan, L., and Rastegar, J., 1997. "Low-harmonic rational Bézier curves for trajectory generation of high-speed machinery". *Computer Aided Geometric Design,* ***14***(3), pp. 251–271.

[68] Sánchez-Reyes, J., 1998. "Harmonic rational Bézier curves, p-Bézier curves and trigonometric polynomials". *Computer Aided Geometric Design,* ***15***, pp. 909–923.

[69] Ge, Q. J., and Ravani, B., 1994. "Computer-aided geometric design of motion interpolants". *ASME Journal of Mechanical Design,* ***116***(3), pp. 756–762.

[70] Ge, Q. J., and Ravani, B., 1994. "Geometric construction of bezier motions". *ASME Journal of Mechanical Design,* ***116***(3), pp. 749–755.

[71] 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.

[72] Park, F. C., and Ravani, B., 1995. "Bezier curves on riemannian-manifolds and lie-groups with kinematics applications". *ASME Journal of Mechanical Design,* ***117***(1), pp. 36–40.

[73] Kim, M. S., and Nam, K. W., 1995. "Interpolating solid orientations with circular blending quaternion curves". *Computer-Aided Design,* ***27***(5), pp. 385–398.

[74] Juttler, B., and Wagner, M. G., 1996. "Computer-aided design with spatial rational b-spline motions". *ASME Journal of Mechanical Design,* ***118***(2), pp. 193–201.

[75] Zefran, M., and Kumar, V., 1998. "Interpolation schemes for rigid body motions". *Computer-Aided Design,* ***30***(3), pp. 179–189.

[76] Srinivasan, L. N., and Ge, Q. J., 1998. "Fine tuning of rational b-spline motions". *ASME Journal of Mechanical Design,* ***120***(1), pp. 46–51.

[77] Zhe, J., and Ge, Q. J., 2007. "Computer aided synthesis of piecewise rational motions for planar 2r and 3r robot arms". *ASME J. of Mechanical Design,* ***129***(10).

[78] Ge, Q. J., Purwar, A., and Wu, J., 2008. "Kinematic convexity of spherical displacements and its application to collision prediction". In SPM '08:

Proceedings of the 2008 ACM symposium on Solid and physical modeling, ACM, pp. 375–380.

[79] PARK, F., 1995. "Distance metrics on the rigid-body motions with applications to mechanism design". *JOURNAL OF MECHANICAL DESIGN,* ***117***(1), MAR, pp. 48–54.

[80] Stolfi, J., 1991. *Oriented Projective Geometry*. Academic Press Professional, Inc.

[81] O'Rourke, J., 2001. *Computational Geometry in C*. Cambridge Univ. Press.

[82] Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. T., 1996. "The quickhull algorithm for convex hulls". *ACM Transactions on Mathematical Software,* ***22***(4), pp. 469–483.

[83] Ravani, B., and Roth, B., 1983. "Motion synthesis using kinematic mappings". *Journal of Mechanisms Transmissions and Automation in Design-Transactions of the Asme,* ***105***(3), pp. 460–467.

[84] Wagner, M. G., 1994. "A geometric approach to motion design". Ph.d. dissertation.

[85] Röschel, O., 1998. "Rational motion design - a survey". *Computer-Aided Design,* ***30***(3), pp. 169–178.

[86] Purwar, A., Chi, X., and Ge, Q. J., 2008. "Automatic fairing of two-parameter rational b-spline motion". *ASME Journal of Mechanical Design,* ***130***(1), p. 011003.

[87] Bodduluri, R. M. C., and McCarthy, J. M., 1992. "Finite position synthesis using image curve of a spherical four-bar motion". *ASME J. of Mechanical Design,* ***114***(1).

[88] Bodduluri, R., 1990. "Design and planned movement of multi-degree of freedom spatial mechanisms". Ph.d. dissertation.

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

[90] Burmester, L. E. H., 1888. *Lehrbuch der kinematik*. A. Felix, Leipzig,.

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

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

[93] Ge, Q. J., 1990. "Kinematics constraints as algebraic manifolds in the clifford algebra of projective three space". Ph.d. dissertation, University of California, Irvine.

[94] 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. 504–510.

[95] Hoeltzel, D. A., and Chieng, W.-H., 1990. "Pattern matching synthesis as an automated approach to mechanism design". *Journal of Mechanical Design,* ***112***(2), pp. 190–199.

[96] Freudenstein, F., 1959. "Harmonic analysis of crank-and-rocker mechanisms with application". *J. Appl. Mech.,* ***26***, pp. 673–675.

[97] Funabashi, H., and Freudenstein, F., 1979. "Performance criteria for high-speed crank-and-rocker linkages .1. plane crank-and-rocker linkages". *Journal of Mechanical Design - Transaction of The ASME,* ***101***(1), pp. 20–25.

[98] Fanhang, K., Midha, A., and Bajaj, A., 1988. "Synthesis of harmonic motion generating linkages .1. function generation". *Journal of Mechanisms Transmissions and Automation in Design - Transactions of The ASME,* ***110***(1), MAR, pp. 16–21.

[99] Fanhang, K., Midha, A., and Bajaj, A., 1988. "Synthesis of harmonic motion generating linkages .2. path and motion generation". *Journal of Mechanisms Transmissions and Automation in Design - Transactions of The ASME,* ***110***(1), MAR, pp. 22–27.

[100] 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.

[101] Chu, J., and Cao., W., 1996. "Optimum synthesis of linkage mechanism for path generation using harmonic characteristic parameters". *Journal of Mechanical Transmission (in Chinese),* ***2***.

[102] Wu, X., C. J. W. C., and Cao, W., 1998. "A new method of dimensional synthesis for planar 4-bar path generation with prescribed timing". In Proceedings of the 25th Bienical Mechanisms Conference, ASME, pp. DETC1998–5932.

[103] Chu, J., Wang, L., and Wu, C., 2004. "Relationship between properties of coupler curve and link's dimensions in 4-bar mechanisms". *Science In China Ser. E Engineering and Materials Sciences (in Chinese),* ***34***(7), pp. 753–762.

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

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

[106] Nie, X., and Krovi., V., 2005. "Fourier methods for kinematic synthesis of coupled serial chain mechanismss". *ASME J. of Mechanical Design,* ***127***(2), pp. 232–241.

[107] Cortese, J. M., and Dyre., B. P., 1996. "Perceptual similarity of shapes generated from fourier descriptors". *Journal of Experimental Psychology: Human Perception and Performance,* ***22***(1), pp. 133–143.

[108] Wu, J., Ge, Q. J., and Gao, F., 2009. "An efficient method for synthesizing crank-rocker mechanisms for generating low harmonic curves". In Proceedings of the 2009 ASME Mechanisms and Robotics Conference, ASME, pp. DETC2009–87140.

[109] 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.

[110] Kirkpatrick, S., Gelatt, C. D., Jr., and Vecchi, M. P., 1983. "Optimization by simulated annealing". *Science,* ***220***, pp. 671–680.

[111] Corana, A., Marchesi, M., Martini, C., and Ridella, S., 1987. "Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm". *ACM Trans. Math. Softw.,* ***13***(3), pp. 262–280.

[112] Hunt, K., 1978. *Kinematic Geometry of Mechanisms*. Clarendon Press, Oxford.

[113] Bottema, O., and Roth, B., 1979. *Theoretical Kinematics*. Dover Publication Inc., New York.

[114] Erdman, A. G., and Sandor, G. N., 1997. *Mechanism Design: Analysis and Synthesis*, 3 ed. Prentice Hall, Upper Saddle River, NJ.

[115] Jüttler, B., and Wagner, M., 1996. "Computer aided design with spatial rational b-spline motions". *ASME Journal of Mechanical Design,* ***119***(2), pp. 193–201.

[116] Sarkisyan, Y. L., G. K. C., and Roth, B., 1973. "Kinematic geometry associated with the least-square approximation of a given motion". *ASME J. Engineering for Industry,* ***95***(2), pp. 503–510.

[117] Artobolevsky, I. I., 1964. *Mechanisms for the Generation of Plane Curves*. Macmillan Co., New York.

[118] 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 IDETC Conferences, ASME, pp. DETC2010/MECH–28287.

[119] Zahn, C. T., and Roskies, R. Z., 1967. "Fourier descriptors for plane closed curves". *IEEE Transactions on Pattern Analysis and Machine Intelligence,* ***C-21***(3), pp. 269–281.

[120] Granlund, G., 1972. "Fourier preprocessing for hand print character recognition". *IEEE Trans. Computers,* ***21***, pp. 195–201.

[121] Wu, J., Ge, Q. J., Gao, F., and Guo, W. Z., 2010. "On the extension of a fourier descriptor based method for four-bar linkage synthesis for generation

of open and closed paths". In Proceedings of the 2010 ASME Mechanisms and Robotics Conference, ASME, pp. DETC2010–29028.

[122] Alter, I., and Schwartz, E. L., 1988. "Psychophysical studies of shape with fourier descriptor stimuli". *Perception,* ***17***, pp. 191–202.