Applying Numerical Algebraic Geometry to Kinematics

Charles W. Wampler and Andrew J. Sommese

Abstract Many problems from kinematics are questions about mappings between algebraic spaces. This chapter presents a mathematical framework for such problems and discusses how numerical algebraic geometry, a computational approach based mainly on polynomial continuation, can be applied to solving them. Publicly available software for numerical algebraic geometry, such as the Bertini package, facilitates the solution of such problems, allowing kinematicians to solve with ease problems that were previously considered extremely difficult or intractable.

1 Introduction

Advances in algorithms and computer speed have brought about a new paradigm in kinematics. The proportion of effort a kinematician must exert in heavy manipulations of algebraic expressions is greatly diminished through the use of computer algorithms, and instead the kinematician may concentrate on formulating the problem and interpreting the answer so as to analyze a geometrically constrained motion or to design a device to produce desired motions. This way of working requires, however, an understanding of what kind of results the computer algorithms are capable of providing and how to use them effectively. This article condenses material from [50] providing the schema of a *mechanism space* that encapsulates most of the sorts of questions arising in kinematics and then summarizes how numerical algebraic geometry, a computational approach based on polynomial continuation, can be applied to solving such problems. In addition to [50], the application of polynomial

Charles W. Wampler
Manufacturing Systems Laboratory, General Motors Global R&D Center, Warren, Michigan, USA
e-mail: charles.w.wampler@gm.com

Andrew J. Sommese
Applied and Computational Mathematics and Statistics, University of Notre Dame, Indiana, USA
e-mail: sommese@nd.edu

continuation to kinematics has been addressed specifically in the tutorials [47, 38] and in substantial portions of the monograph [40].

Some notable milestones in the application of polynomial continuation to kinematics include early demonstrations that the inverse kinematics problem of general six-revolute serial-chain manipulators has 16 solutions [43], that the forward kinematics problem of general Stewart-Gough platforms has 40 solutions [33], and that the nine-point path synthesis problem for four-bar linkages has 1442 triples of Roberts-cognate solutions [48, 49]. Some indication of the difficulty of these problems is that the one with the lowest root count, the "6R problem," was once declared, in the equivalent form of the 7R spatial loop, the "Mount Everest of kinematics" [13]. While the nine-point path synthesis problem was not fully solved until nearly 70 years after its first statement in 1923 [1], the eventual cracking of the problem was presaged by partial solutions generated in 1963, in the early days of applying computers to kinematics, with a heuristic version of continuation [35]. With subsequent improvements in the technique of continuation and increases in computer speed, even the nine-point problem is now a routine calculation used as a test case for software packages in numerical continuation. (At present, it is still beyond the range of symbolic methods in computer algebra.)

These early successes all share the property that the questions to be answered have a finite number of solutions. In other words, the solution sets to be computed are *zero-dimensional*. But it is common in kinematics to explore problems having higher-dimensional solution sets. Examples include the motion curve traced out by a 1-degree-of-freedom (1DOF) mechanism, the boundary of the reachable workspace of a robot, or sets of design alternatives that satisfy an under-specified precision-point mechanism synthesis problem. (An example of the latter case is the classical center-point/circle-point Burmester curves for four given locations of a body in the plane [11].) Algebraic curves, surfaces, and beyond are all of interest in the field.

An approach for consistent treatment of higher-dimensional cases was begun in [39], where the term "numerical algebraic geometry" was coined. The essential construct in the approach is a *witness set*, in which general linear equations are appended to slice out a finite number of representative points, called *witness points*, on a higher-dimensional algebraic set. Many properties of the set can be gleaned from its witness set, and the set can be explored by tracking the witness points as the linear slicing space is moved continuously. The solution set of a system of polynomial equations can be factored into its irreducible components, each represented by a witness set, and given witness sets for two or more irreducible components, algorithms exist for finding their intersection.

The main techniques of numerical algebraic geometry are freely available in the software package Bertini [5, 6]. PHCpack [44] also implements some algorithms of the field, while Hom4PS2 [17], POLSYS_PLP [52], and POLSYS_GLP [42] offer only algorithms for computing isolated solutions.

This chapter first describes a way of viewing problems from kinematics in terms of mappings between algebraic spaces. Mechanisms, including robots, that consist of rigid links connected by the most common kinds of joints have kinematic relations that are naturally polynomial. After establishing this basic framework, atten-

tion shifts to the algorithms of numerical algebraic geometry and how to use them. Beginning with the following section, the remainder of this chapter is excerpted from [50], with permission, edited for continuity.

2 Notation

This chapter uses the following notations.

- $\mathrm{i} = \sqrt{-1}$, the imaginary unit.
- For polynomial system $F = \{f_1, \ldots, f_n\}$, $F : \mathbb{C}^N \to \mathbb{C}^n$, and $y \in \mathbb{C}^n$,

$$\mathscr{V}(F) = \mathscr{V}(f_1, \ldots, f_n) := \{x \in \mathbb{C}^N \mid F(x) = 0\}.$$

- "DOF" means "degree(s)-of-freedom."
- $\mathbb{T}^k$ is the k-dimensional torus, the cross product of k circles.
- $\mathbb{C}^*$, pronounced "Cee-star," is $\mathbb{C} \setminus 0$, the complex plane omitting the origin.
- $\mathbb{P}^n$ is n-dimensional projective space, the set of lines through the origin of $\mathbb{C}^{n+1}$.
- Points in $\mathbb{P}^n$ may be written as homogeneous coordinates $[x_0, \ldots, x_n]$, not all zero. The coordinates are interpreted as ratios, so $[x_0, \ldots, x_n] = [\lambda x_0, \ldots, \lambda x_n]$ for any $\lambda \in \mathbb{C}^*$.
- A quaternion u is written in terms of the elements $\mathbf{1}, \mathbf{i}, \mathbf{j}, \mathbf{k}$ as

$$u = u_0\mathbf{1} + u_1\mathbf{i} + u_2\mathbf{j} + u_3\mathbf{k}, \quad u_0, u_1, u_2, u_3 \in \mathbb{C}.$$

  A quaternion u with $u_0 = 0$ can be interpreted as an ordinary spatial vector. We use $u * v$ to denote the quaternion product and u' to denote quaternion conjugation.

3 Algebraic Kinematics

Kinematicians are quite accustomed to writing problems as systems of polynomial equations. For example, we often begin with a standard Denavit-Hartenberg formulation in which for each rotational joint angle θ_i, the trigonometric functions $\sin\theta_i$ and $\cos\theta_i$ appear. But by the simple maneuver of defining variables

$$c_i = \cos\theta_i, \quad s_i = \sin\theta_i,$$

and appending the trigonometric identity

$$c_i^2 + s_i^2 = 1,$$

such expressions become polynomial. In this section, we step back a moment to see why so many problems in kinematics are algebraic at their core.

3.1 Rigid-Body Motion Spaces

Consider first that the six-dimensional set of rigid-body transformations in three-space, $SE(3)$. The defining properties of $SE(3)$ are that each transformation must:

- preserve distances between points (the body is rigid), and
- preserve handedness (the body does not transmute into its mirror image).

The notation $SE(3)$ stands for 'special Euclidean transforms on 3-space'.

At root, rigid-body transforms are algebraic because squared distances are algebraic. Translation by a vector $\mathbf{p} \in \mathbb{R}^3$ and orthogonal transformation by a matrix $C \in O(3) = \{C \in \mathbb{R}^{3\times 3} \mid C^T C = I\}$ preserve distance, but to preserve handedness, C must be restricted to $SO(3)$ by requiring $\det C = 1$.

The most useful representations of $SE(3)$ are as follows.

- $(\mathbf{p}, C) \in \mathbb{R}^3 \times SO(3)$, where $SO(3) = \{C \in \mathbb{R}^{3\times 3} \mid C^T C = I, \det C = 1\}$. This acts on a vector $\mathbf{v} \in \mathbb{R}^3$ to transform it to $\mathbf{u} = C\mathbf{v} + \mathbf{p}$.
- The 4×4 homogeneous transform version of this, where $(\mathbf{p}, C)$ are placed in a matrix so that the transform operation becomes

$$\begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix} = \begin{bmatrix} C & \mathbf{p} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix}.$$

- Study coordinates $[e, g] \in S_6^2 \subset \mathbb{P}^7$, where $e = (e_0, e_1, e_2, e_3)$, $g = (g_0, g_1, g_2, g_3)$ and S_6^2 is the six-dimensional hypersurface given by the equation

$$e_0 g_0 + e_1 g_1 + e_2 g_2 + e_3 g_3 = 0, \tag{1}$$

  known as the Study quadric. Interpreting e and g as quaternions, the transform operation is

$$\mathbf{u} = (e * \mathbf{v} * e' + g * e')/(e * e').$$

In all three cases, the representations live on an algebraic set. The transform operation is also algebraic in the first two cases and equations involving the Study transform operation become algebraic after clearing $e * e'$ from denominators.

There are several subgroups of $SE(3)$ that are of interest. Most prominent is $SE(2)$, the set of planar rigid-body transformations, with representations as follows.

- $(\mathbf{p}, C) \in \mathbb{R}^2 \times SO(2)$, where $SO(2) = \{C \in \mathbb{R}^{2\times 2} \mid C^T C = I, \det C = 1\}$. The transform rule looks identical to the spatial case: $\mathbf{u} = C\mathbf{v} + \mathbf{p}$.
- The unit-circle form $\{(x, y, s, c) \in \mathbb{R}^4 \mid c^2 + s^2 = 1\}$. This is the same as the former with $\mathbf{p} = x\mathbf{i} + y\mathbf{j}$ and

$$C = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}.$$

- The tangent half-angle form $(x, y, t) \in \mathbb{R}^3$, in which rotations become

$$C = \frac{1}{1+t^2} \begin{bmatrix} 1-t^2 & -2t \\ 2t & 1-t^2 \end{bmatrix}.$$

- Isotropic coordinates $\{(p,\overline{p},\theta,\overline{\theta}) \in \mathbb{C}^4 \mid \theta\overline{\theta} = 1\}$. Real transforms must satisfy $p^* = \overline{p}$ and $\theta^* = \overline{\theta}$. The action of transform $(p,\overline{p},\theta,\overline{\theta})$ on a vector given by isotropic coordinates $(v,\overline{v})$ is the vector $(u,\overline{u})$ given by

$$(u,\overline{u}) = (p+\theta v, \overline{p}+\overline{\theta}\overline{v}).$$

Again, each of these representations lives on an algebraic set and has an algebraic transform operation (after clearing denominators in the tangent half-angle form). Clearly, $SE(2)$ is a three-dimensional space.

Another subspace of interest is the set of *spherical transforms*, that is, just $SO(3)$, another three-dimensional space. This is $SE(3)$ with the translational portion set identically to zero. The terminology "spherical" derives from the fact that this is the set of motions allowed by a spherical ball set in a spherical socket of the same diameter.

It is useful to note that since points transform algebraically, so do lines and planes, as these may be formed as linear combinations of two or three points, respectively. Labeling the points that define them, the order of the points defines an orientation for the lines and planes. Unbound unit vectors also transform algebraically, that is, by ignoring translation and applying only rotation.

3.2 Algebraic Joints

A mechanism is a collection of rigid bodies connected by joints. Without the joints, each body could move with six degrees of freedom anywhere in $SE(3)$. Typically, we declare one body to be "ground" and measure the locations of all the other bodies relative to it, so a collection of n bodies lives in $SE(3)^{n-1}$. Joints are surfaces of contact between bodies that constrain the motion of the mechanism to a subset of $SE(3)^{n-1}$. *Algebraic joints* are those which constrain a mechanism to *algebraic subsets* of $SE(3)^{n-1}$.

The most important joints for building mechanisms are the *lower-order pairs*. These are pairs of identical surfaces that can stay in full contact while still allowing relative motion. In other words, they are formed by a surface that is invariant under certain continuous sets of displacements. The lower-order pairs form six possible joint types, having the following standard symbols: R, revolute; P, prismatic; H, helical (screw); C, cylindrical; E, plane; and S, Spherical. The importance of the lower-order pairs derives from the fact that surface-to-surface contact spreads forces of contact over a larger area, reducing stresses that might wear out the machinery.

Fortunately – from the viewpoint of an algebraic geometer – five of these six joint types are algebraic. The exception is the H joint, which produces a translation proportional to rotation angle θ along with a rotation that depends on $\cos\theta$ and $\sin\theta$. The mixture of θ with $\cos\theta$ and $\sin\theta$ makes the motion non-algebraic. An alternative line of reasoning is to observe that a helix and a plane containing its symmetry axis intersect in an infinite number of isolated points. Any algebraic curve in

$\mathbb{R}^3$ intersects a plane in at most a finite number of isolated points. However, helical joints are rarely used as a direct motion constraint in a manner that impacts kinematic analysis. Instead, screws are usually used to transmit power along a prismatic joint. Consequently, the geometric motion of a great many mechanisms is governed by algebraic joints.

To demonstrate that a joint type is algebraic, one may write down the constraint conditions it imposes between the transforms for the two bodies in contact, say A and B. The algebraic lower-order pairs can be reduced to equating some combination of points, lines, or planes. As these are all transformed algebraically, equating them gives an algebraic constraint. In brief, one may confirm algebraicity for each joint by noting the following equivalences:

R: equate a point and an oriented line through it in A to similar in B,
P: equate a line and an oriented plane containing it in A to similar in B,
C: equate a line of A to one of B,
E: equate a plane of A to one of B, and
S: equate a point of A to one of B.

This suffices to show that the lower-order pairs R, P, C, E, and S are all algebraic.

Each joint can be described either *extrinsically* in terms of the constraint it imposes, as above, or *intrinsically* in terms of the freedom it allows between the transform for, say, body B relative to body A. Using 4×4 transform notation, and letting ${}^iT^j$ be the transform for frame j relative to frame i, one has

$$ {}^0T^B = {}^0T^A {}^AT^B \tag{2}$$

Suppose that the joint between A and B is the kth joint of a mechanism, an algebraic lower-order pair. Then, the relative transform ${}^AT^B$ can be written as

$$ {}^AT^B = A_k X B_k, \tag{3}$$

where A_k and B_k are constant transforms describing the location of the joint in bodies A and B, resp., and X is variable of the form

$$ X = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & a \\ \sin\theta & \cos\theta & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

For each joint type, the contents of X vary as follows.

R: Use (4) with $a = b = c = 0$, leaving θ as the joint variable.
P: Use (4) with $\theta = a = b = 0$, leaving c as the joint variable.
C: Use (4) with $a = b = 0$, leaving θ and c both as joint variables.
E: Use (4) with $c = 0$, leaving θ, a, and b all as joint variables.
S: Use

$$ X = \begin{bmatrix} C & 0 \\ 0 & 1 \end{bmatrix}$$

with $C \in SO(3)$ as the joint freedom.

The foregoing descriptions are meant mainly to show that the kinematic relations of rigid-body mechanisms involving only algebraic joints are all algebraic. Other formulations can sometimes be more succinct or convenient. When modeling the joints with a low number of freedoms (R, P, C) it is usually more convenient to use an intrinsic formulation, while S joints are usually best modeled extrinsically. In some cases, one may avoid introducing transforms for some of the links altogether. For example, see §3.5.5 for a formulation of the kinematics of the Stewart-Gough platform that avoids introducing any transforms for the links that compose the legs.

3.3 Mechanism Types, Families, and Spaces

Having shown that mechanisms built with rigid links and algebraic joints have kinematic relations that are polynomial, we move on to show that a wide variety of kinematics problems can be placed into a common format involving mappings between algebraic sets. To do so, we need the definitions of a mechanism type and a mechanism family.

Definition 1. A *mechanism type* is defined by the number of links, n_L, and a symmetric $n_L \times n_L$ adjacency matrix T whose (i, j)th element denotes the type of joint between links i and j, one of R, P, H, C, E, S, or $\emptyset$, where $\emptyset$ indicates no connection. By convention, all diagonal elements are $\emptyset$.

(Each joint appears twice in the matrix: $T_{i,j} = T_{j,i}$ are the same joint.) We assume here that the joints are limited to the lower-order pairs, but the list of possibilities could be extended. The enumeration of all possible mechanism types for each value of n_L without double-counting mechanisms that are isomorphic under renumbering of the links is a problem in discrete mathematics. Choosing a prospective mechanism type is the first step in a mechanism design effort, and methods for guiding the enumeration of promising alternatives fall into the category of *type synthesis*. In this paper, we assume that this crucial step is already done so that we begin with a mechanism type.

Each mechanism type has an associated parameter space. We have seen in § 3.2 one way to model each of the algebraic lower-order pairs, R, P, C, E, and S, extrinsically in terms of feature points, oriented lines, and oriented planes. Alternatively, in the intrinsic formulation of (3), the transforms A_k and B_k parameterize the joints. The cross-product space of all these geometric features forms a universal parameter space for the mechanism type. One may choose to model the joints in a more parsimonious way, but we assume that in the alternative model there still exists a parameterization for each joint and an associated parameter space for all the joints taken together. For example, for a succession of R and P joints, the Denavit-Hartenberg (D-H) formalism gives a minimal parameterization. (See [23] or any modern kinematics textbook for a definition.) The D-H parameters are link lengths, link offsets,

and link twist angles. Treating the twists as unit circles, the parameter space becomes algebraic.

Definition 2. A *universal mechanism family* (T,Q) is a mechanism type T with an associated parameter space Q describing the geometry of the joints. We assume that Q is irreducible.

If one has a parameter space Q that is not irreducible, each irreducible component should be considered to define a separate universal mechanism family.

Definition 3. A *mechanism family* (T,Q') is a subset of a universal mechanism family (T,Q) restricted to an irreducible algebraic subset $Q' \subset Q$.

Examples of the common sorts of algebraic restrictions that define a mechanism family include the condition that the axes of two R joints in a certain link must be parallel, perpendicular, or intersecting, etc. As a particular example, consider that the universal family of spatial 3R serial-link chains includes the family of 3R planar robots, wherein the R joints are all parallel. One should appreciate that there can be subfamilies within families, and so on.

For certain mechanisms, all points of the links move in parallel planes, hence the links move in $SE(2)$ and the mechanism is said to be *planar*. In particular, a mechanism family wherein all joints are either rotational R with axis parallel to the world **z**-direction or prismatic P with axis perpendicular to the world **z**-direction is planar.

Definition 4. The *link space* Z for an n link mechanism is $SE(3)^{n-1}$, where one of the links is designated as ground $(\mathbf{p},C) = (0,I)$. Any of the isomorphic representations of $SE(3)$ from § 3.1 can be used as models of $SE(3)$. If the mechanism family is planar, then $Z = SE(2)^{n-1}$ in any of its isomorphic representations from § 3.1.

Definition 5. The *mechanism space* M of a mechanism family (T,Q) is the subset of $Z \times Q$ that satisfies the joint constraints.

Proposition 1. *If a mechanism family is built with only the algebraic joints R, P, C, E, and S, then its mechanism space is algebraic.*

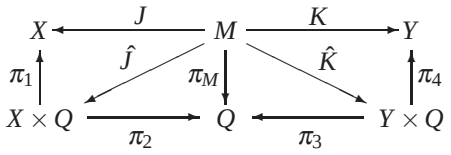
Proof. Section 3.1 shows that Z is algebraic and Q is algebraic by assumption. That is, Z and Q are sets defined by algebraic equations. Section 3.2 shows that the algebraic joints impose algebraic constraints on the coordinates of Z and Q, and hence all the defining equations for M are algebraic. □

Definition 6. A *mechanism* is a member of a mechanism family (T,Q) given by a set of parameters $q \in Q$.

3.4 Kinematic Problems in a Nutshell

In this section, we present an abstract formulation that summarizes all the main types of geometric problems that arise in kinematics. In the next section, we will discuss more concretely how to map a mechanism into this formulation.

The key to our formulation is the following diagram:



(5)

The main elements of the diagram are four sets X, M, Y, Q and three maps J, K, π_M. The four sets are as follows.

- X is the *input space* of the mechanism. In robotics, it is usually called the "joint space." Its coordinates are typically quantities that we command by controlling motors or other actuators.
- Y is the *output space*, often called the "operational space" in robotics. Its coordinates are the final output(s) we wish to obtain from the mechanism, such as the location of a robot's hand.
- Q is the *parameter space* of a family of mechanisms. It is the set of parameters necessary to describe the geometry of the joints in each link. Each point in Q is therefore a specific mechanism with designated link lengths, etc. The whole set Q constitutes a family of mechanisms, such as the set of all 6R robot arms, with the coordinates of Q representing all possible link lengths, etc. We assume that Q is an irreducible algebraic subset of some $\mathbb{C}^m$, that is, it is an irreducible component of $\mathscr{V}(G)$ for some system of algebraic functions $G : \mathbb{C}^m \to \mathbb{C}^{m'}$. If $\mathscr{V}(G)$ has more than one irreducible component, then each such component is considered a different family of mechanisms.
- M is the *mechanism space*, which describes all possible configurations of the mechanism for all possible parameters. Let Z be the space of all possible locations of the links when they are disconnected. That is, for an N-link spatial mechanism with one link designated as ground, $Z = SE(3)^{N-1}$. Then, M is the subset of $Z \times Q$ where the link locations satisfy the constraints imposed by the joints between them. Let $F : Z \times Q \to \mathbb{C}^c$ be a set of polynomials defining the joint constraints. Then, $M = \mathscr{V}(F) \cap \mathscr{V}(G)$ is an extrinsic representation of M. Each point $(z,q) \in M$ is a specific mechanism $q \in Q$ in one of its assembly configurations $z \in Z$. In some cases, it is more natural to describe M intrinsically via an irreducible set, say Θ, that parameterizes the freedoms of the joints of the mechanism, so that Z becomes $\Theta \times SE(3)^{N-1}$. We will use this, for example, to describe M for 6R serial-link robots. In such a representation, F includes the equations that define Θ along with the equations relating link poses to joint freedoms and equations for the constraints imposed by closing kinematic loops. Formulating such equations is part of the art of kinematics, and we will not delve into it in this paper beyond what is necessary to present specific examples.

After choosing a representation for $SE(3)$, and if present, for the joint freedom space Θ, the space Z is a subspace of some Euclidean space, $Z \subset \mathbb{C}^v$, and $z \in Z$ has coordinates $z = (z_1, \ldots, z_v)$.

Three maps are defined on M, as follows.

- $J : M \to X$ is the *input map*, which extracts from M the input values. The symbol J acknowledges that the inputs are usually a set of joint displacements.
- $K : M \to Y$ is the *output map*, which extracts from M the output values.
- $\pi_M : M \to Q$ is a projection that extracts the parameters from M. It is the natural projection operator on $Z \times Q$ restricted to M given by $\pi_M : (z,q) \mapsto q$.

If the maps F,G,J,K are merely analytic (instead of algebraic) and the spaces X,M,Y,Q are analytic, the above framework still applies, and we may pose questions in the analytic setting. In particular, H joints are analytic but not algebraic. The advantage of restricting to the algebraic setting is the existence of a much more powerful algebraic theory that enables the use of algebraic techniques, including numerical algebraic geometry, to be employed in answering the questions.

The commutative diagram is completed by defining $\hat{J} := (J, \pi_M)$ and $\hat{K} := (K, \pi_M)$ and the associated natural projections $\pi_1, \pi_2, \pi_3, \pi_4$.

It should be understood that M characterizes a family of mechanisms, such as the family of spatial 6R serial-link robots, the family of planar four-bar linkages, or the family of Stewart-Gough platforms. Maps J and K are tailored to an application of the mechanism. For a four-bar function generator, J gives the input angle and K gives the output angle, while for a four-bar path generator, K gives instead the position of the coupler point.

Using the diagram of (5) succinctly summarizes the algebraic setting of almost all kinematic problems. The problems can be broadly classified into three types of problems:

- Analysis (mobility analysis, forward and inverse kinematics, workspace analysis),
- Synthesis (precision point problems), and
- Exceptional mechanisms.

We describe each of these in more detail next.

3.4.1 Analysis

In analysis problems, one has a specific mechanism, say $q^* \in Q$, and one wishes to analyze some aspect of its motion.

Definition 7. The *motion of a mechanism* given by parameters $q^* \in Q$ in a family with mechanism space M is $\pi_M^{-1}(q^*) = M \cap \mathscr{V}(q - q^*) \subset Z \times Q$. This can also be called the *motion fiber over* q^*.

In the following, it is also convenient to define the inverses of J and K:

$$J^{-1}(x) = \{(z,q) \in M \mid J(z,q) = x\}, \qquad K^{-1}(y) = \{(z,q) \in M \mid K(z,q) = y\}.$$

These are defined for $x \in X$ and $y \in Y$, respectively. In the set $J^{-1}(x)$ for a particular $x \in X$, q is not fixed, so this inverse applies across a whole mechanism family.

When we wish to address just one particular mechanism, q^*, we want to consider the inverse of $\hat{J}$ instead:

$$\hat{J}^{-1}(x,q^*) = \{(z,q) \in M \mid \hat{J}(z,q) = (x,q^*)\}.$$

Similarly, we have:

$$\hat{K}^{-1}(y,q^*) = \{(z,q) \in M \mid \hat{K}(z,q) = (y,q^*)\}.$$

The basic problems in analysis are as follows.

- **Motion decomposition** of a *mechanism* breaks $\pi_M^{-1}(q^*)$ into its irreducible components, often called assembly modes by kinematicians. (See § 7 for a description of irreducible components.) The numerical irreducible decomposition of $\pi_M^{-1}(q^*)$ finds the dimension and degree of each assembly mode and provides a set of witness points on each.
- **Motion decomposition** of a *mechanism family* breaks M into its irreducible components. If $A \subset M$ is one of these components, then $\pi_M(A) \subset Q$ is the subfamily of mechanisms that can be assembled in that mode, $\dim \pi_M(A)$ is the dimension of the subfamily, and $\dim A - \dim \pi_M(A)$ is the mobility of that mode.
- **Mobility analysis** seeks to find the degrees of freedom (DOFs) of the mechanism, that is, mobility is $\dim \pi_M^{-1}(q^*)$. As the dimension of an algebraic set is always taken to be the largest dimension of any of its components, this definition of mobility picks out the assembly mode (or modes) having the largest number of DOFs. There are simple formulas, known as the Gruebler-Kutzbach formulas, that correctly estimate the mobility for a wide range of mechanisms, and even more mechanisms submit to refined analysis based on displacement group theory, but there exist so-called "paradoxical" mechanisms that have higher mobility than these methods predict. To handle all cases, one needs to analyze the equations defining M in more detail taking into account that q^* may be on a subset of Q having exceptional mobility.
- **Local mobility analysis** finds the mobility of a mechanism in a given assembly configuration. That is, given $(z^*,q^*) \in Z \times Q$, one wishes to find

$$\text{Local mobility} := \dim_{(z^*,q^*)} \pi_M^{-1}(q^*). \tag{6}$$

A mechanism can have more than one assembly mode, corresponding to the irreducible components of $\pi_M^{-1}(q^*)$. The local mobility is the dimension of the assembly mode that contains the given configuration, z^*, or if there is more than one such mode, the largest dimension among these.
- **Forward kinematics** seeks to find the output that corresponds to a given input x^* for a mechanism q^*. That is, for $x^* \in X$ and $q^* \in Q$, one wishes to find

$$FK(x^*,q^*) := K(\hat{J}^{-1}(x^*,q^*)). \tag{7}$$

Example: given the joint angles of a particular 6R serial-link robot, find its hand pose.

- **Inverse kinematics** is similar to forward kinematics but goes from output to input. For $y^* \in Y$ and $q^* \in Q$ find

$$IK(y^*, q^*) := J(\hat{K}^{-1}(y^*, q^*)). \tag{8}$$

  Example: given the hand pose of a particular 6R serial-link robot, find all sets of joint angles that reach that pose.
- **Singularity analysis** finds configurations where the maps lose rank. If we have found a motion decomposition of the mechanism, then for each assembly mode $A \subset \pi_M^{-1}(q^*)$ there is an associated input space $J(A)$ and an output space $K(A)$. The input and output maps have Jacobian matrices $\partial J/\partial z$ and $\partial K/\partial z$. Assume for the moment that A is a reduced algebraic set. (For example, $\mathscr{V}(x-y)$ is a reduced line in the (x,y)-plane, while the double line $\mathscr{V}((x-y)^2)$ is non-reduced.) For generic points $(z,q^*) \in A$, the Jacobian matrices have a constant rank, say $\operatorname{rank}[\partial J/\partial z(z,q^*)] = r_J$ and $\operatorname{rank}[\partial K/\partial z(z,q^*)] = r_K$. Then, there may be input and output singularities, as follows.

  Input Singularities: $\{(z,q^*) \in A \mid \operatorname{rank}\frac{\partial J}{\partial z}(z,q^*) < r_J\}$. In the common case that $\partial J/\partial z$ is square and generically full rank, these are the special configurations where, to first order, the mechanism can move without any change in its input.

  Output Singularities: $\{(z,q^*) \in A \mid \operatorname{rank}\frac{\partial K}{\partial z}(z,q^*) < r_K\}$. In the common case that $\partial K/\partial z$ is square and generically full rank, these are the special configurations where, to first order, the mechanism can move without any change in its output.

  If A is a non-reduced assembly mode, one might wish to consider the input and output singularities of the reduction of A, which can be analyzed via a deflation of A. (See § 7.3.
- **Workspace analysis** seeks to find all possible outputs of a robot or mechanism. Ignoring limits on the inputs, this is just the set $K(\pi_M^{-1}(q^*))$. The main concern in practice is the set of outputs for real assembly configurations, so letting $A_{\mathbb{R}}$ denote the real points in an assembly mode A, the corresponding workspace is $K(A_{\mathbb{R}})$. Output singularities and joint limits induce boundaries in the workspace.
  Example 1: for a 6R serial-link robot, find all possible poses that the hand can reach.
  Example 2: for a given four-bar linkage with output defined as the position of its coupler point, find the coupler curve.
  Example 3: for a 6-SPS (Stewart-Gough) platform robot with limits on the leg lengths, find all possible poses of the moving platform.

The motion of a mechanism over the complexes contains its real motion, but the extraction of the real motion from the complex one can be difficult, all the more so as the dimensionality of the motion grows. See § 7.5 for a discussion.

The problems presented above mainly concern the geometry of a mechanism's motion, where principally angles, positions, and poses enter the picture. As indicated by the questions of singularity analysis, one may also be concerned with differential relations between these, so that joint rates, linear velocity, and angular ve-

locity may become objects of study. Since these are all related to the mechanism space M through its derivatives, these too fit into the algebraic setting, as do static forces and torques, through the principles of virtual work.

3.4.2 Synthesis

While analysis determines how a mechanism moves, synthesis finds mechanisms that move in a specified way. Synthesis problems begin with a set of desired outputs or a set of input/output pairs and seek to find the mechanisms that will produce these. Synthesis tends to be harder than analysis because one must consider the ability of the mechanism to reach each desired state. In essence, we must consider multiple copies of M simultaneously. The relevant construction in algebraic geometry is called the *fiber product*. Instead of studying M, one works with $M \times_Q M = \mathscr{V}(F(z_1;q),F(z_2;q))$, which comprises two copies of the motion for the same mechanism. Clearly, the fiber product operation can be extended to triple fiber products and higher. Forming the k-fold fiber product

$$M_Q^k := \underbrace{M \times_Q \cdots \times_Q M}_{k \text{ times}},$$

if $\pi_{M,k}$ is the projection from M_Q^k that picks out its parameters, q, then $\pi_{M,k}^{-1}(q^*)$ for $q^* \in Q$ gives back k copies of the motion fiber over q^*. We may also define a map K_k acting on M_Q^k to produce k outputs and a map JK_k acting on M_Q^k to produce k input/output pairs.

With these maps, we may define several kinds of synthesis problems. The following problems are known as *precision point* problems, since there is a set of specified points which the mechanism must interpolate exactly.

- **Output synthesis** seeks mechanisms that can reach a set of specified outputs. For $(y_1,\ldots,y_k) \in Y^k$, we wish to find the set

$$\{q \in Q \mid K_k(\pi_{M,k}^{-1}(q)) = (y_1,\ldots,y_k)\}.$$

  Kinematicians distinguish between different types of output synthesis.

  Path synthesis finds mechanisms where the path of a point of the mechanism interpolates a set of given points. In this case, K is defined on M such that $Y \subset \mathbb{C}^3$.

  Body guidance In this case, the output is the pose of one body of the mechanism, that is, $Y \subset SE(3)$. The purpose of the mechanism is to guide that body through a set of specified poses.

- **Input/output synthesis** seeks mechanisms that produce a coordinated input/output relationship specified by a set of input/output pairs. For $((x_1,y_1),\ldots,(x_k,y_k)) \in (X \times Y)^k$, we wish to find the set

$$\{q \in Q \mid JK_k(\pi_{M,k}^{-1}(q)) = ((x_1,y_1),\ldots,(x_k,y_k))\}.$$

A common case is a 1DOF mechanism, such as a four-bar, with the input being the angle of one link with respect to ground. Then, with K defined as in a path synthesis problem, the input/output problem becomes *path synthesis with timing*. Similarly, one can have *body guidance with timing*. (The nomenclature derives from an assumption that the input moves at a constant rate.) If the input and output are both angles of the mechanism, then input/output synthesis becomes *function generation*, as the precision points approximate some desired functional relationship between input and output.

What makes these problems difficult is that the whole system of equations defining M is repeated k times, increasing the total degree of the system exponentially in k.

For any precision point problem, there is a maximum number of precision points that can be specified. Roughly speaking, this is the total number of independent parameters in the mechanism family under consideration divided by the number of constraints placed on the parameters by each precision point. If more than the maximum number of precision points is specified, then there will in general be no mechanism that interpolates them exactly. One may then reformulate the problem by defining an error metric and seek mechanisms whose motion best fits the specified approximation points. This is analogous to finding a best-fit line that approximates three or more points.

We should note that all these synthesis problems have been formulated only at the geometric level. It is also possible to specify motions at the level of velocity or acceleration or to mix specifications at several levels. For a 1DOF motion, differential relations can be approximated by limits as precision points approach each other. For this reason, classical synthesis theory sometimes distinguishes between *finitely-separated precision points* and *infinitesimally-separated precision points*. We will not discuss synthesis problems involving differential relations further here.

3.4.3 Exceptional Mechanisms

While M describes the motion of an entire family of mechanisms, $\pi_M^{-1}(q^*)$ is the motion of a particular mechanism in the family. For any generic $q \in Q$, attributes such as the mobility of the mechanism or the local mobilities of its assembly modes all stay constant. However, there may be algebraic subsets of Q where mobilities increase. These exceptional mechanisms are often called "overconstrained mechanisms," as a slight perturbation of the parameters off of the exceptional set into a generic position suddenly brings in extra constraints that reduce mobility. One may define subsets of M where the local mobility is constant, that is,

$$\mathscr{D}_k^* = \{(z,q) \in M \mid \dim_{(z,q)} \pi_M^{-1}(q) = k\}. \tag{9}$$

The closures of these, $\mathscr{D}_k = \overline{\mathscr{D}_k^*}$, are algebraic sets. When $\mathscr{D}_j \subset \mathscr{D}_k$, $j > k$, we say that $\mathscr{D}_j$ is an exceptional set of mechanisms, a family of overconstrained mechanisms.

The discovery of exceptional mechanisms is perhaps the most difficult kind of kinematics problem. One may think of these as a kind of synthesis problem where the only thing that is specified about the motion is its mobility. As in the precision-point synthesis problems, it turns out that fiber products play a central role. We leave further discussion of this to § 7.6.

3.5 Fitting into the Nutshell

Sections 3.1 and 3.2 show that any mechanism composed of n rigid links connected by any combination of R, P, C, E, or S joints leads to a set of constraint equations that is algebraic in the link locations $(\mathbf{p}_j, C_j) \in SE(3)$, $j = 1, \ldots, n$ and is also algebraic in the parameters defining the joints. In § 3.3,3.4, we put forward a schema that formulates a wide variety of kinematics problems in terms of spaces X, M, Y, Q and maps J, K, π between them. In this section, we will detail how some example mechanism types fit into this schema.

3.5.1 Planar 3R Robots

Consider first the universal family of 3R planar serial-link robots. These have $n_L = 4$ links, one of which is ground. The adjacency matrix has R in each element of the super- and sub-diagonals and $\emptyset$ everywhere else. Since the mechanism is planar, the link space is $Z = SE(2)^3$. Using the reference frames as indicated in Figure 3.5.1, we have coordinates for Z as

$$z_1 = (P_x, P_y, x_1, y_1), \quad z_2 = (Q_x, Q_y, x_2, y_2), \quad z_3 = (R_x, R_y, x_3, y_3), \tag{10}$$

where (P_x, P_y) are the coordinates of point P, etc., and $x_j = \cos\phi_j$, $y_j = \sin\phi_j$, $j = 1, 2, 3$. Here, ϕ_1, ϕ_2, ϕ_3 are the absolute rotation angles of the links. Accordingly, the algebraic equations defining the link space Z are

$$x_j^2 + y_j^2 - 1 = 0, \qquad j = 1, 2, 3. \tag{11}$$

The parameters of the mechanism are just the link lengths (a, b, c), so the parameter space Q is $\mathbb{C}^3$. In the plane, the constraint imposed on two links by a rotational joint is the coincidence of the point of connection. Point $O = (0, 0)$ in the ground must coincide with point $(-a, 0)$ in the reference frame of link 1:

$$(0, 0) = (P_x - ax_1, P_y - ay_1). \tag{12}$$

Similarly, the other two joints impose the constraints

$$(P_x, P_y) = (Q_x - bx_2, Q_y - by_2) \text{ and } (Q_x, Q_y) = (R_x - cx_3, R_y - cy_3) \tag{13}$$

Accordingly, Eqs. 11–13 define the mechanism space M.

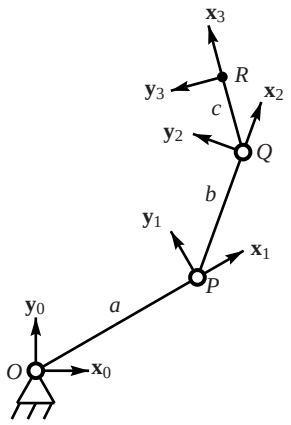


Fig. 1 Planar 3R robot with reference frames.

To complete the picture, we need the maps π_M, J, K. The projection $\pi_M : M \to Q$ simply picks out the parameters:

$$\pi : (z_1, z_2, z_3, a, b, c) \mapsto (a, b, c). \tag{14}$$

Assuming the input space $X = \mathbb{T}^3$ is the relative rotation angles $(\theta_1, \theta_2, \theta_3)$ represented by cosine/sine pairs, the difference formulas for cosine and sine give

$$J : (z_1, z_2, z_3, a, b, c) \mapsto ((x_1, y_1), (x_2x_1 + y_2y_1, y_2x_1 - x_2y_1), (x_3x_2 + y_3y_2, y_3x_2 - x_3y_2)). \tag{15}$$

Finally, assuming the output space $Y = SE(2)$ is the location of reference frame 3, the output map is

$$K : (z_1, z_2, z_3, a, b, c) \mapsto (z_3). \tag{16}$$

If instead the robot is applied to just positioning point R in the plane, we have $Y = \mathbb{C}^2$ with the output map

$$K' : (z_1, z_2, z_3, a, b, c) \mapsto (R_x, R_y). \tag{17}$$

With these definitions, the problems of forward kinematics, inverse kinematics, reachable workspace, and exceptional sets all fit neatly into the nutshell schema.

3.5.2 Spatial 6R robots

The case of spatial 6R robots is quite similar to the 3R planar case, but we shall choose to handle the joint constraints by introducing variables implicitly modeling the freedom of the joints rather than explicitly writing constraint equations. A 6R serial-link chain has $n_L = 7$ links, one of which is ground. The adjacency matrix

has entries of R on the super- and sub-diagonals and $\emptyset$ elsewhere. Let the link space be $Z = \mathbb{T}^6 \times SE(3)^6$, with unit circle representations of the joint angles and 4×4 homogeneous transforms for the link locations, so that Z is represented as $z = \{(c_j, s_j, {}^0T^j),\ j = 1,\ldots,6\}$ with

$$c_j^2 + s_j^2 - 1 = 0, \qquad j = 1,\ldots,6. \tag{18}$$

The first factor of Z is precisely the joint space $X = \mathbb{T}^6$ and the output is the location of the "hand," the last frame in the chain, ${}^0T^6$. The link parameters are 4×4 transforms $A_j \in SE(3)$, $j = 0,1,\ldots,6$. One can use general transforms, but the Denavit-Hartenberg formalism shows that by choosing reference directions aligned with joint axes and their common normals, it suffices to parameterize the A_j as

$$A_j = \begin{bmatrix} 1 & 0 & 0 & a_j \\ 0 & \alpha_j & -\beta_j & -\beta_j d_j \\ 0 & \beta_j & \alpha_j & \alpha_j d_j \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad j = 0,\ldots,6. \tag{19}$$

In this expression, (α_j, β_j) are a cosine/sine pair for the twist of link j, a_j is the length of the link (along its **x**-direction), and d_j is the link offset distance (along its **z**-direction). To keep A_j in $SE(3)$, these must satisfy

$$\alpha_j^2 + \beta_j^2 - 1 = 0, \qquad j = 0,\ldots,6. \tag{20}$$

With this parameterization, the parameter space is $Q = \mathbb{T}^7 \times \mathbb{C}^{14}$ with coordinates $q = \{(\alpha_j, \beta_j, a_j, d_j),\ j = 0,\ldots,7\}$. Joint rotations $R_z(c_j, s_j)$ of the form

$$R_z(c,s) = \begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{21}$$

alternate with relative link displacements A_j to give the transforms of the link locations as

$${}^0T^1 = A_0 R_z(c_1, s_1) A_1, \qquad {}^0T^j = {}^0T^{j-1} R(c_j, s_j) A_j, \qquad j = 2,\ldots,6. \tag{22}$$

Combining these gives

$${}^0T^6 = K_{6R}(z,q) := A_0 \prod_{j=1}^{6} R_z(c_j, s_j) A_j \tag{23}$$

Equations 18–21 define the mechanism space M in terms of coordinates (z,q). The associated maps from M to Q, $X = \mathbb{T}^6$, and $Y = SE(3)$ are

$$\pi_M : (z,q) \mapsto q, \qquad J : (z,q) \mapsto ((c_j, s_j), j = 1,\ldots,6), \qquad K = K_{6R} : (z,q) \mapsto {}^0T^6. \tag{24}$$

3.5.3 Four-Bar Linkages

The four-bar has four links with four R joints. If we call the ground link 0, the two links connected to ground as links 1 and 2, and the coupler as link 3, then the adjacency matrix has entries $T_{1,3} = T_{2,3} = T_{0,1} = T_{0,2} = \mathrm{R}$. Using isotropic coordinates (see §3.1), let $(\phi, \bar{\phi})$ represent the orientation of the coupler link and let $(\theta_1, \bar{\theta}_1)$ and $(\theta_2, \bar{\theta}_2)$ be the rotations of links 1 and 2, and let $(p, \bar{p})$ be the coupler point position. [Recall that in isotropic coordinates, we represent a vector, say $\mathbf{a} = \alpha\mathbf{i} + \beta\mathbf{j}$, by a complex number $a = \alpha + \beta\mathrm{i}$ and its conjugate $\bar{a} = \alpha - \beta\mathrm{i}$.] Hence, the link space Z is given by coordinates $z = (p, \bar{p}, \phi, \bar{\phi}, \theta_1, \bar{\theta}_1, \theta_2, \bar{\theta}_2)$ subject to the unit length conditions of

$$\phi\bar{\phi} = \theta_1\bar{\theta}_1 = \theta_2\bar{\theta}_2 = 1 \tag{25}$$

Referring to Figure 2, a four-bar can be parameterized by $Q = \mathbb{C}^{10}$ with coordinates $q = (a_1, \bar{a}_1, a_2, \bar{a}_2, b_1, \bar{b}_1, b_2, \bar{b}_2, \ell_1, \ell_2)$. With these notations, the mechanism space for four-bar linkages is the solution set of the equations

$$\begin{aligned} \ell_1\theta_1 &= p + \phi b_1 - a_1, \qquad & \ell_1\bar{\theta}_1 &= \bar{p} + \bar{\phi}\bar{b}_1 - \bar{a}_1, \\ \ell_2\theta_2 &= p + \phi b_2 - a_2, \qquad & \ell_2\bar{\theta}_2 &= \bar{p} + \bar{\phi}\bar{b}_2 - \bar{a}_2, \\ \theta_1\bar{\theta}_1 &= 1, \quad \theta_2\bar{\theta}_2 = 1, \quad \phi\bar{\phi} = 1. \end{aligned} \tag{26}$$

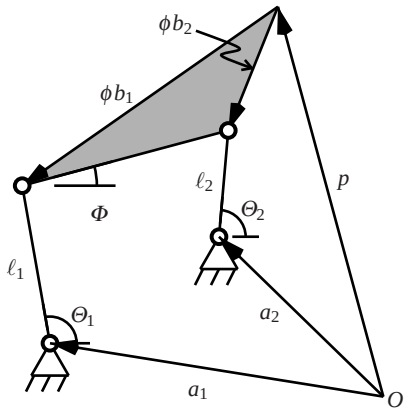


Fig. 2 Vector diagram of a four-bar linkage.

The input and output spaces and their associated maps depend on the application of the mechanism. For path generation, we have output $Y = \mathbb{C}^2$ with map $K = K_{\mathrm{path}} : (z,q) \mapsto (p, \bar{p})$. For body guidance, $Y = SE(2)$ with $K = K_{\mathrm{guide}} : (z,q) \mapsto (p, \bar{p}, \phi, \bar{\phi})$. If timing along the coupler curve or timing of the body motion are of concern, we

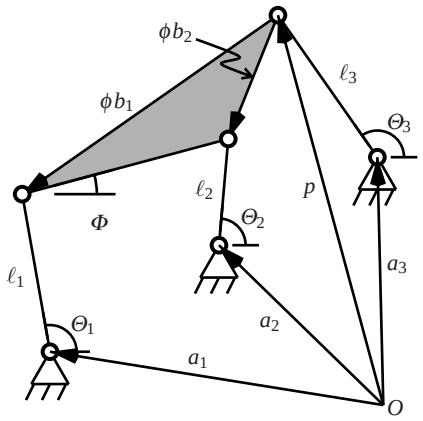


Fig. 3 3-RPR planar platform robot.

may name the angle of one of the links connected to ground as input, say $X = \mathbb{T}^1$ given by $J : (z,q) \mapsto (\theta_1, \bar{\theta}_1)$. For function generation, the input is the angle of link 1 and the output is the angle of link 2, so $J : (z,q) \mapsto (\theta_1, \bar{\theta}_1)$ and $K = K_{\text{fcn}} : (z,q) \mapsto (\theta_2, \bar{\theta}_2)$.

The raw equation sets that come out of the above formulation often benefit from further algebraic manipulation before submitting the system to a numerical solution procedure. In particular, for the path synthesis problem, it can be beneficial to eliminate some variables. In the nine-point path synthesis problem—in general, nine is the maximal number of precision points that can be exactly interpolated—we wish to find parameters q such that the coupler curve $K_{\text{path}}(\pi_M^{-1}(q)) \subset \mathbb{C}^2$ passes through points $(p_i, \bar{p}_i)$, $i = 0, \ldots, 8$. With a little algebra, this can be reduced to solving a system of eight polynomials

$$f_{cc}(p_j, \bar{p}_j; q) = 0, \qquad j = 1, \ldots, 8, \tag{27}$$

each of degree seven. See [48, 50] for derivations, based on a similar formulation from Roth and Freudenstein [35].

3.5.4 Planar 3-RPR Platforms

A 3-RPR planar platform robot has a moving triangle supported from a stationary triangle by three RPR legs, as in Figure 3. Coordinates for the mechanism space M of the 3-RPR planar platform are an extension of those for the four-bar with $a_3, \bar{a}_3, \theta_3, \bar{\theta}_3, \ell_3$ appended with the additional equations:

$$\ell_3 \theta_3 = p - a_3, \quad \ell_3 \bar{\theta}_3 = \bar{p} - \bar{a}_3, \quad \text{and} \quad \theta_3 \bar{\theta}_3 = 1. \tag{28}$$

However, there is a shuffle in which coordinates are parameters and which are variables of the motion. The new maps are

$$\begin{array}{rl} \pi & : (z,q) \mapsto (a_1, \bar{a}_1, a_2, \bar{a}_2, a_3, \bar{a}_3, b_1, \bar{b}_1, b_2, \bar{b}_2) \\ J & : (z,q) \mapsto (\ell_1, \ell_2, \ell_3) \\ K & : (z,q) \mapsto (p, \bar{p}, \phi, \bar{\phi}) \end{array} \tag{29}$$

3.5.5 Stewart-Gough Platforms

For the forward and inverse kinematics problems of the 6-SPS platform, we do not need to explicitly represent transforms for the upper and lower leg segments. It is enough to use the leg lengths and the transform for the moving platform. Hence, the link space is $Z = \mathbb{C}^6 \times SE(3)$, and if we use Study coordinates for $SE(3)$, the space is $Z = \mathbb{C}^6 \times S_6^2$, where $S_6^2 \subset \mathbb{P}^7$ is the Study quadric given by (1). With leg lengths $L_1, \ldots, L_6$, the coordinates of Z are $(L_1, \ldots, L_6), [e,g]$. The parameter space Q consists of the vectors $\mathbf{a}_j, \mathbf{b}_j \in \mathbb{C}^3$, $j = 1, \ldots, 6$, that specify the centers of the S joints in the base and moving platforms. The mechanism space M is given by the Study quadric along with the leg-length equations

$$L_i^2 = ||(g * e' + e * \mathbf{b}_j * e')/(e * e') - \mathbf{a}_j||_2^2, \qquad j = 1, \ldots, 6, \tag{30}$$

which after expanding and clearing denominators becomes for $j = 1, \ldots, 6$

$$\begin{aligned} 0 = {} & g * g' + (\mathbf{b}_j * \mathbf{b}_j' + \mathbf{a}_j * \mathbf{a}_j' - L_j^2) e * e' + (g * \mathbf{b}_j' * e' + e * \mathbf{b}_j * g') \\ & - (g * e' * \mathbf{a}_j' + \mathbf{a}_j * e * g') - (e * \mathbf{b}_j * e' * \mathbf{a}_j' + \mathbf{a}_j * e * \mathbf{b}_j' * e') \end{aligned} \tag{31}$$

The input space $X = \mathbb{C}^6$ is the set of leg lengths $L_1, \ldots, L_6$, and the output space is $Y = S_6^2$ is the Study quadric for the transform of the moving platform. The maps J, K are the associated natural projections.

4 Overview: Numerical algebraic geometry

The fundamental problem in numerical algebraic geometry is to numerically compute and manipulate the solution set of a system of polynomials

$$f(x) := [f_1(x), \ldots, f_n(x)], \tag{32}$$

where $x = (x_1, \ldots, x_N) \in \mathbb{C}^N$. As we have seen in the preceding sections, problems in kinematics often concern parameterized systems, that is, polynomial systems of the form

$$f(x,q) := [f_1(x,q), \ldots, f_n(x,q)] \tag{33}$$

with $x \in \mathbb{C}^N$ and $q \in Q$, where Q is an irreducible algebraic set. We may have simply $Q = \mathbb{C}^M$, a Euclidean space of M independent parameters, but we may also have a Q formed with elements from $SE(3)$ or the unit circle. It is important to note that in practice, an engineer might not know the exact twist angle of a link, but when it

comes to solving the associated polynomial system, it is known that the sine/cosine pair of the angle must lie on the unit circle.

Historically, within numerical algebraic geometry, the problem of finding isolated solutions for *square systems*, i.e., systems such as Eq. 32 in the case $n = N$, came first. The most basic tool is homotopy continuation (or continuation for short), which consists of studying a square system of polynomials $f(x)$ by starting with a simpler system $g(x)$ that we know how to solve and deforming $g(x)$ and the solutions of $g(x) = 0$ to $f(x)$ and the solutions of $f(x) = 0$. A good source for explication of this basic approach is the book [25], with more modern treatments given in [20, 40].

The solution of non-square systems ($n \neq N$ in Eq. 32) came considerably later than methods for square systems. The techniques we employ always reformulate such problems to reduce them once again to finding isolated solutions.

At first sight, the case of $n > N$ (more equations than unknowns) would seem to be numerically unstable, as a small perturbation of the system can obliterate a solution. The is true for any solution sets of dimension greater than $N - n$. The fact that saves the day is that the existence of such solutions depends on the parameters lying exactly on a parameter space, and we assume we know the exact equations that define that space. Consider a pair of parameters that should lie on a unit circle. When we work numerically, the pair will rarely lie exactly on the circle, but the exact unit circle condition is known, and by using extra digits, the pair of parameters can be placed as close as needed to the true parameter space. Thus, to make the computation of sets of dimension greater than $N - n$ robust, the Bertini software package [5] implements *adaptive multiprecision arithmetic* (see §6.3) that adjusts the number of digits as needed. This same technology stabilizes the numerics of isolated singular and near-singular roots of square systems.

To provide a complete ability to solve systems of polynomials, one must be able to deal not just with isolated roots but also with higher-dimensional solution sets (curves, surfaces, etc.). Obviously, this arises when $n < N$, as there are not enough equations to determine an isolated root.[1] The approach of dealing with this in numerical algebraic geometry is a data structure called a *witness set* in which extra linear equations are introduced in order to cut out general isolated points on the higher dimensional sets. These points can then be computed using the techniques for finding isolated solution points. To cut out isolated points on a set of dimension m, one must augment the original system with $N - m$ general linear equations. Thus, if the solution set is has dimension $m > N - n$, the augmented system has $n + m > N$ equations, which has the potential for being numerically unstable. The procedure for stabilizing this situation depends on *randomization*, as described in §6.9.

[1] At least not in complex space. In real space, singular isolated roots are possible with $n < N$. See §7.5.

5 Finding Isolated Roots

In this section we briefly discuss the continuation method of finding isolated solutions of a polynomial system (32). Various aspects of how this process is made robust and efficient are highlighted. From the viewpoint of a user of software packages for continuation, a few things, particularly the division of variables into groups for multihomogenization, requires some level of expertise from the user, but most of these measures can be automated without user input. The motivation for a user to understand the basic solution processes is for making an informed choices of which software to use, as discussed further in §8, and which algorithms to use within a chosen software package. Although it will not be addressed here, advanced users may also change configuration settings that can affect the speed and robustness to adapt performance to the specific needs of their applications.

5.1 Homotopy

For the square case of N polynomials in N unknowns, one of the most classical homotopies is the total-degree homotopy that uses the "γ trick" of [27]:

$$H(x,t) := (1-t)\begin{bmatrix} f_1(x) \\ \vdots \\ f_N(x) \end{bmatrix} + \gamma t \begin{bmatrix} g_1(x) \\ \vdots \\ g_N(x) \end{bmatrix}, \tag{34}$$

where each polynomial g_j has degree the same as f_j and the solution set of the system $(g_1(x),\ldots,g_N(x)) = 0$ consists of $d_1 \cdots d_N$ nonsingular isolated solutions. When γ is chosen as a random complex number, then with probability one, the homotopy satisfies the properties:

1. $\big\{(x,t)\big|t \in (0,1]; x \in \mathbb{C}^N; H(x,t) = 0\big\}$ is a union of $d_1 \ldots d_N$ full-rank paths, say $x_1(t),\ldots,x_{d_1\cdots d_N}(t)$, starting at the solutions of $H(x,1) = 0$; and
2. the set of limits $\lim_{t\to 0} x_j(t)$ that are finite include all the isolated solutions of $H(x,0) = 0$.

This theory justifies the use of the very simple start system defined by $g_j(x) = x_j^{d_j} - 1$, $j = 1,\ldots,N$.

6 Multihomogeneous homotopies

Constructing good homotopies with the number of paths not too different from the number of isolated solutions of f was an important research topic at the end of the 20th century. There is detailed discussion of this topic in [40, Chap.8̃]. For systems

that are not too large, which includes many mechanism systems, multihomogeneous can be quite useful. Multihomogeneous homotopies were first proposed in [26] and discussed in [40, § 8.4].

For a hint of what difference the selection of homotopy can make, consider the 3-RPR forward kinematics problem of §sec:3RPR, which is to solve the 10 equations (26,28) for the 10 unknowns $(\theta_1, \theta_2, \theta_2, \phi, p, \bar{\theta}_1, \bar{\theta}_2, \bar{\theta}_2, \bar{\phi}, \bar{p})$. Six of the equations are linear but the four unit-length equations are degree 2, for a total degree of 16. Yet, as a 2-homogeneous system, with variables divided into two groups as $(\theta_1, \theta_2, \theta_2, \phi, p), (\bar{\theta}_1, \bar{\theta}_2, \bar{\theta}_2, \bar{\phi}, \bar{p})$, the system has at most $\binom{4}{2} = 6$ isolated roots, which is in fact the exact root count for general cases. In the Bertini software package, the move from a total degree homotopy to a 2-homogeneous one is done by simply using two separate `variable_group` statements to declare the variables.

A more impressive example is the nine-point path synthesis problem mentioned in §3.5.3. In the Roth-Freudenstein formulation of (27), the total degree of the system is $7^8 = 5,764,801$. An alternative in [48] uses a 2-homogeneous formulation of the problem that has a root count of just 286,720. A special homotopy that takes advantage of a 2-way symmetry reduces the number of paths to 143,360, which is about one-fortieth (1/40) that of the total degree homotopy. This formulation lead to the first complete solution of this classical problem in kinematics.

6.1 Sparse homotopies

A polynomial of degree d in N variables can have $\binom{N+d}{N}$ different monomials but problems arising in applications typically have many fewer than this. Multihomgeneous homotopies take advantage of sparseness associated to limited mixing of products between variables within groups. This is a common occurrence in kinematics, but it does not capture all the kinds of sparseness that may arise. In particular, in a multihomogeneous formulation, variable groupings stay fixed across the whole system of polynomials.

Linear product homotopies capture sparseness at a finer scale, including, for example, groupings of variables that change from one polynomial to the next. The main alternatives are laid out in [40, chap. 8], based on theory developed in [45] and generalized in [30]. Versions of this are available in POLSYS_PLP [52] and POLSYS_GLP [42].

Polyhedral homotopies take full advantage of any sparse structure in a polynomial system. While the multihomogeneous and linear product homotopies require the user to identify good groupings of the variables—no efficient method is known for finding the best groupings—the polyhedral method completely automates the creation of a homotopy with the minimal number of paths for systems with the sparse structure of the target system. The leading approach for polyhedral homotopies is described in [21] and is implemented in HOM4PS2 [17].

Unfortunately, the formation of a polyhedral homotopy depends on an intricate combinatorial calculation, called the mixed volume, whose complexity grows

rapidly with the number of variables and which is not easily parallelizable. So while it is an excellent approach for small to medium size problems, it becomes untenable for large ones.

6.2 Regeneration homotopy

For larger systems, the regeneration approach discussed scales up more readily than the polyhedral approach [15, 14]. Moreover, it can take advantage of structure in the system beyond just sparsity. In particular, the coefficients that appear in a polynomial system may satisfy interrelationships that reduce the root count. In the kinematics context, this might arise as parameters that satisfy a unit circle condition or that lie in $SE(3)$. Regeneration does not build this structure into the homotopy from the beginning, as is done in the sparse homotopies, but rather it discovers structure by treating the system equation-by-equation. Regeneration methods are available in the Bertini software package.

6.3 Adaptive multiprecision

Higher-precision arithmetic (i.e., greater than double precision) makes the basic process of path tracking more bulletproof, while adaptive multiprecision, in which precision is adjusted up or down as needed, accomplishes this goal with greater efficiency. Consider the nine-point path-synthesis problem for four-bars just mentioned above. Of the 143,360 paths in the homotopy used in [48], all but 4326 end on various degenerate sets. The 4326 roots of interest appear in a three-way symmetry, as expected from the classical result known as Roberts cognates [11, 34]. The original computations in 1992 on this problem were conducted in double precision followed by a check for any points missing from the expected symmetry groups. Reruns in extended precision cleaned up any paths having questionable numerical stability, filling in the missing points and thereby establishing with high confidence that the solution list was complete. More recent experiments with a path-tracker having adaptive multiprecision found that in order to be tracked accurately, 0.83% of the paths required precision higher than double precision somewhere in the middle of the paths before returning to double precision (see [8, §5.3]). This approach consistently finds the entire solution set without requiring any reruns or other corrective actions. Although in the early 1990s, this was an extraordinarily difficult computation, we now use this problem as a moderately difficult test system.

6.4 Parallelism

One of the highly advantageous features of polynomial continuation is that all the paths of a homotopy can be tracked independently. For many algorithms in the field, this makes the bulk of the computation "embarrassingly parallel." The Bertini software package offers a parallel version for the Linux operating system [5]. Unfortunately, it is much harder to efficiently parallelize the steps that set up a polyhedral homotopy, so that, as of this writing, the leading polyhedral package, Hom4PS2 [17], is only available for single-processor systems.

6.5 Solutions at infinity

One difficulty in path tracking is paths that go to infinity as $t \to 0$. Tracking such a path may be computationally expensive as it is infinitely long and numerical conditioning may be poor as the magnitudes of the solution variables grow. Morgan's projective transformation trick [24] is to work on a random coordinate patch in the projective space containing $\mathbb{C}^N$. This maneuver keeps the magnitude of the variables and the path lengths finite. It is common for polynomial continuation packages, such as Bertini, to perform homogenization automatically.

6.6 Multiplicities and Deflation

It is widely appreciated that for a polynomial in one variable, the multiplicity of a solution is governed by the number of derivatives that vanish there. In several variables, multiplicity still makes sense, but directional derivatives and algebraic relations between them come into play, so the situation is more complicated.

For the moment, consider only isolated solutions of a polynomial system. (We take up multiplicity as it applies to higher-dimensional sets in §7.3.) When the multiplicity μ of solution z^* is greater than one, z^* is said to be a singular solution. Such solutions are difficult to work with numerically. A primary problem is that the vanishing derivatives ruin the convergence properties of Newton's method near the singular point. For this reason, tracking paths to z^* from a good homotopy for z^* is computationally expensive and often impossible in double precision. To deal with these points effectively, we use endgames (see §6.7) and adaptive precision (see §6.3).

Deflation is another approach for dealing with singular points [32, 31, 18, 12, 19, 14]. Since singularities are caused by the vanishing of derivatives, deflation is a process for re-establishing regularity by including equations that are only satisfied by solutions with derivatives that vanish to the correct order at the singular point. The main difficulty with this procedure lies in determining the rank of certain matrices formed from derivatives of the equations. This leads to a vicious circle, since

computing the singular solution accurately is the initial objective, and one needs an accurate value for the solution to determine the ranks. The upshot is that for isolated solution points the cost of computing a deflation system often dwarfs the cost of computing the point accurately using the endgame methods in the next subsection. Yet, deflation can be of great service when working with higher dimensional solution sets of multiplicity greater than one. (See §7.3.)

6.7 Endgames

Let $H(x,t) = 0$ be a homotopy, and let $z(t)$ with $t \in (0,t]$ be one of its solution paths. Endgames refer to the process of computing $x^* := \lim_{t\to 0} z(t)$. We may assume by using Morgan's projective transformation trick, §6.5, that x^* is finite, i.e., $x^* \in \mathbb{C}^N$. However, when x^* is singular, more than one path may be converging to the same spot, and all those paths become more and more difficult to track as t approaches zero.

There are several ways to circumvent this problem, but all depend on the fact that instead of just tracking t along the real line, we can consider what happens as t moves into the complex plane near the origin. One of the most effective ways of computing x^* is to track t in a small circle around the origin and to compute a Cauchy integral [29], parallelized in [3].

6.8 Parameter homotopy

The schema for kinematics problems in §3.4 shows that they naturally arise as systems of parameterized polynomials. This fact can be used to reduce the computational cost of solving more than one problem from the same parameterized family. The power of this concept when applied to finding isolated roots derives from the fact that once one has solved a single general example from a parameterized family, one has a bound on the number of isolated roots of any other member of the family. Moreover, one can find all isolated roots of any subsequent examples in a parameter homotopy that tracks solutions from the first example as the parameters are moved along a general, continuous, path in parameter space, starting at the parameters of the first example and ending at those of the new target system.

As a first example, consider the forward kinematics problem for general Stewart-Gough (6-SPS) platforms, given by (1,31). These are 7 equations in $[e,g] \in \mathbb{P}^7$, all quadratic. One can solve a general member of this family using a total-degree homotopy having $2^7 = 128$ paths and find the problem has just 40 solutions. One can solve any other example in the family with a coefficient-parameter homotopy that has just 40 paths. Moreover, there are several different subfamilies of interest wherein some of the S joints coincide. One of these is the octahedral family where the base and moving links are both triangles, with two legs terminating at each vertex. For this

family, the problem has only 16 roots, appearing in a two-way symmetry. (Reflection of the mechanism through the plane of the base does not alter the geometry.) Since a coefficient-parameter homotopy respects this symmetry, only eight paths need to be tracked. As discussed in [40, § 7.7], after solving one general member of any Stewart-Gough subfamily, the remaining ones can be solved with an optimal number of paths by coefficient-parameter homotopy. Although these problems are all simple enough that a elimination approach can be devised—and this has been done for most cases—each special case requires a new derivation. In contrast, homotopy methods cover all the cases seamlessly.

A more extreme illustration of the power of the coefficient-parameter homotopy technique is provided by the nine-point path-synthesis problem for four-bars. As we mentioned earlier, the best multihomogeneous formulation found for the problem has 143,360 paths of which only 4326 have finite endpoints. So after a one-time execution of that homotopy for a general example, all subsequent examples can be solved with a coefficient-parameter homotopy having only 4326 paths. But the story gets even better, because the 4326 solutions appear in a three-way symmetry called Robert's cognates [34]. The coefficient-parameter homotopy respects this symmetry, so only one path in each symmetry group needs to be tracked, resulting in a homotopy with only 1442 paths. This is nearly a 100-fold decrease in the number of paths compared to the original multihomogeneous homotopy (which was already a 40-fold decrease from the total degree homotopy).

6.9 Randomization

Situations may arise where the number of equations, n, is greater than the number of variables, N. A case in point is the 6R inverse kinematics problem, which is to (23) for (c_i, s_i), $i = 1, \ldots, 6$ subject to the unit circle conditions, $c_i^2 + s_i^2 = 1$, $i = 1, \ldots, 6$. Since the transform equation (23) is equivalent to 12 polynomials (the bottom row of the 4×4 matrices is trivial), we have altogether 18 equations in only 12 variables. Even so, we expect solutions, because each transform lives on $SE(3)$, so the equations are compatible. We have seen that polynomial continuation is capable of finding all isolated solutions in the square case ($n = N$), but $n > N$ requires extra measures.

Sometimes, one can pick out a subset of equations and be sure of getting all isolated roots, the remaining equations being redundant. But this takes extra knowledge about the structure of the system, because there exist systems where solving a subset does not work. An example is the system

$$xy = 0, \quad x(x-y-1) = 0, \quad y(x-y-1) = 0, \tag{35}$$

which has 3 isolated roots, (0,0), (1,0), and (0,1), whereas each subsystem formed by any two of the three equations has only one isolated root.

A method which does work in general is to take N random linear combinations of the equations. With probability one, this preserves all isolated roots of the original system, although it may introduce additional extraneous roots. In the case of (35), the randomized system has four isolated roots: the original three and an extraneous one that depends on the random coefficients chosen in forming the linear combinations. We call this process *randomization*.

An important property of randomization is that it numerically stabilizes solutions. Numerical evaluation of a polynomial inevitable introduces small perturbations: the evaluation is not exact. In a strict sense, the numerically evaluated system, consisting of $n > N$ randomly perturbed equations, will have no solutions. But a perturbed randomized system, being square, still has exact solutions, and these will be close to the solutions to the original exact system.

A similar trick works for higher dimensional sets. The case of isolated solutions for $n > N$ is the $m = 0$ special case of a solution set of dimension m with $m > N - n$. With probability one, a system of $N - m$ random linear combinations of the original n equations preserves all solution sets of dimension m, but may introduce extraneous sets at that dimension as well. This leads us into the next topic: positive-dimensional solution sets.

7 Computing Positive-Dimensional Sets

We already mentioned in §4 that in numerical algebraic geometry, positive-dimensional sets (curves, surfaces, etc.) are represented by witness sets. An irreducible algebraic set is an algebraic set that cannot be expressed as a union of a finite number of proper algebraic subsets. One of the main goals in numerical algebraic geometry is to compute, for a given polynomial system F, all the irreducible components of $\mathscr{V}(F)$. This is called the *numerical irreducible decomposition*, and it consists of one witness set for each irreducible component of $\mathscr{V}(F)$. In kinematics, when $F(z,q) = 0$ is the polynomial system for a mechanism family, as in the schema presented in §3.4, then for a particular mechanism, say $q^* \in Q$, the irreducible components of $\mathscr{V}(F(z,q^*))$ are the *assembly modes* of the mechanism. Although it may seem nonintuitive, the assembly modes might not all have the same dimension: the same mechanism can sometimes have a different number of DOFs depending on which mode it is assembled in. It is even possible for such assembly modes to meet, meaning that a mechanism could change its number of DOFs at certain special configurations. Such mechanisms have been called "kinematotropic" mechanisms [53].

A witness set, $\mathscr{A}$, for an m-dimensional irreducible algebraic set $A \subset \mathbb{C}^N$ is a data structure having three members:

- a polynomial system F such that A is an irreducible component of $\mathscr{V}(F)$,
- a generic linear space $L \subset \mathbb{C}^N$ of dimension $N - m$ (equivalently, m random linear equations), and
- the set of isolated points $W = L \cap A$.

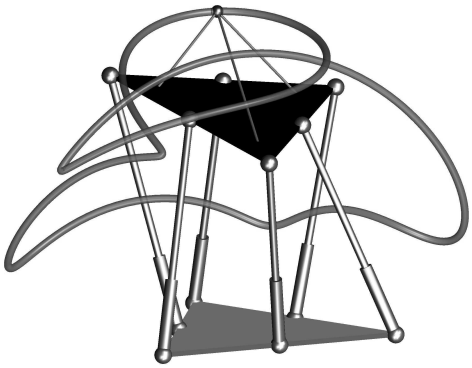


Fig. 4 Griffis-Duffy platform of Type I.

We usually write this as the triple $\mathscr{A} = \{F, L, W\}$, and by context, L may mean either the set of linear equations or the linear space they define. In the numerical irreducible decomposition of $\mathscr{V}(F)$, F itself plays the role of the first member, the random linear equations are constructed by use of a random number generator, and the witness points W are found by polynomial continuation. When $m = N - n$, the system formed by appending m linear equations to the original n equations of F produces a square system, so W can be found using a conventional homotopy. When $m > N - n$, randomization is used, as described in §6.9, to produce a system of only $N - m$ equations so that once again we obtain a square system when the linear equations are appended.

A complete description of the procedures for computing a numerical irreducible decomposition are beyond the scope of this chapter. In short, one proceeds by testing every possible dimension m and factoring the witness points at each dimension according to the irreducible components. In numerical work, the use of intersections with a linear spaces to find higher dimensional sets was first proposed in[39], where the term *numerical algebraic geometry* was coined. See [40] for a full exposition or [50] for a briefer summary of the various techniques used to make computation of the numerical irreducible decomposition practical. The current preferred approach for descending through the dimensions is the regenerative cascade [15], which is the default method in the Bertini software package.

One interesting example of the application of the numerical irreducible decomposition is a special case of the Stewart-Gough platform called the Griffis-Duffy Type I architecturally-singular platform. These have base and moving platforms that are equilateral triangles, with legs connecting vertices of the base to midpoints of the moving platform and vice versa in a cyclic pattern [16, 38]. No matter what the leg lengths are, a general case of this type of platform has a motion curve in Study coordinates of degree 28. This is illustrated in Figure 4, where the path of a small sphere attached to the moving plate is shown. This path has degree 40 in $\mathbb{R}^3$. For a special case of this in which the two triangles are congruent and the leg lengths are

equal, this curve factors into five pieces: four sextics and one quartic. The numerical irreducible decomposition is able to find all of these [38].

7.1 Membership Tests

If one has a witness set $\mathscr{A} = \{F, L, W\}$, as above, for m-dimensional irreducible component $A \subset \mathscr{V}(F)$ and a point $z^* \in \mathscr{V}(F)$, it can sometimes be of interest to know if z^* is in A. The main membership tests used in practice are variants of the monodromy membership test (see [40, Chap. 15.4] for details). In short, we form a linear system $L'(z) = B(z - z^*) = 0$ (so z^* is a solution) with B a random matrix in $\mathbb{C}^{m \times N}$ and set up a homotopy

$$H(z,t) = \{F(z), tL(z) + (1-t)L'(z)\} = 0 \tag{36}$$

where L is the system of linear equations from $\mathscr{A}$. Then, $z^* \in A$ if and only if one of the homotopy paths starting from the points in W for $t = 1$ lands on z^* as $t \to 0$. The Bertini package provides this test.

7.2 Component sampling

It can also be of interest to generate additional points on an irreducible component, that is, to sample the component. To do so randomly, one merely follows the same homotopy as in (36), except that L' is chosen completely at random. This functionality is also provided by Bertini.

7.3 Deflation Revisited

Just as isolated solution points may appear with multiplicity greater than one (double points, triple points, etc.), positive-dimensional irreducible solution components also may appear with higher multiplicity. Such solution components are said to be *nonreduced*. The generic multiplicity of an irreducible component is the same as the multiplicity of the witness points W considered as isolated solutions of the augmented system $\{F, L\} = 0$. The methods for deflating isolated solutions mentioned in §6.6 can be applied to the augmented system at the witness points, and the conditions placed on derivatives can be carried forward as L is deformed. Deflation is necessary to efficiently carry out monodromy membership or component sampling on a nonreduced irreducible component, as otherwise the homotopy paths of (36) would be singular at every step along t.

7.4 Local Dimension

Given a solution x^* of a polynomial system $f(x) = 0$, it can be of interest to determine the local dimension at x^* of the solution set $\mathscr{V}(f)$. This means the dimension of the irreducible component of $\mathscr{V}(f)$ that contains x^*, or if there is more than one such component, the largest dimension among them. This has a technical use in computing the numerical irreducible decomposition, where points must be sorted by dimension. The method in [4] handles this task.

The determination of local dimension also has direct applicability to kinematics in finding the local mobility of a mechanism in some given pose, as defined in (6). The rank of the Jacobian matrix tells a kinematician how many infinitesimal DOFs exist, but does not indicate how many of these extend to finite motion DOFs. To settle the issue often requires the computation of higher order derivatives. In general, without computing an irreducible decomposition for the whole solution set, one does not have enough information to limit the number of derivatives that must be checked, but pre-specifying the order of the derivatives keeps computation finite and yields the depth-bounded local dimension [46]. For large enough depth, this is the correct local dimension, but even if one stops short of the theoretically sufficient depth, one may obtain a practically sufficient result, as the difference between an infinitesimal DOF associated to a very high multiplicity and a true finite DOF can become academic. A high multiplicity isolated root in the rigid-body model may in fact exhibit substantial motion when small elastic deformations of the links, which are always present in a physical device, enter the picture. See [46] for the method and some kinematic examples.

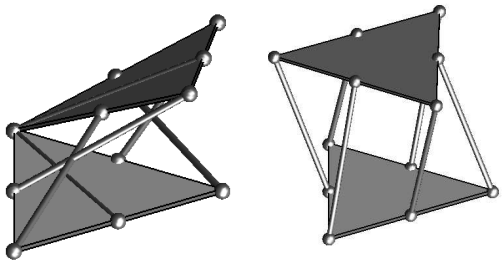
7.5 Real Sets

Throughout this article we have dealt almost exclusively with *complex* solutions, even though isolated and positive dimensional *real* solutions are the main interest for most applications.

For isolated solutions, we may simply pick the isolated real solutions out of the isolated complex solutions. For positive dimensional sets, extra work is required, and the complexity of extracting a complete description of the real set, every start and stop and every self-crossing, can be high. One illustration of the difficulty is that in singular situations, the real and complex dimensions can differ. For example, consider the equation $x^2 + y^2 = 0$, which when considered in complex space defines a pair of lines (namely, $x = \pm iy$), but which only has an isolated real solution at the origin, where the lines cross in a singularity.

One-dimensional sets (curves) are relatively straightforward [22]. The key notion is to consider a family of parallel hyperplanes sweeping across $\mathbb{R}^N$. Each hyperplane that kisses the curve in a tangency marks a turning point in the real curve, and in between, the planes cut the curve transversely. Thus, every arc of the real curve is found by first solving for the tangency condition and then slicing the curve in

between the turning points. Isolated real points, if they exist, are found as real tangencies that have no incoming real arcs. Recently, an algorithm has been developed along similar lines for solving the more difficult problem of finding the real points in a complex surface [10].



On a quadric On the quartic

Fig. 5 Selected poses of the foldable Stewart-Gough platform. From [22] by permission, ©SIAM.

In § 7, we illustrated the Griffis-Duffy Type I platform robot, a special case of the Stewart-Gough (6-SPS) platform, and mentioned that the motion for the Griffis-Duffy Type II subcase factors into five pieces: four sextics and one quartic. In [22], an even more special example of a Griffis-Duffy Type II robot is considered, one whose leg lengths are all equal to the altitude of the base and moving triangles (which are congruent equilateral triangles). This robot is unusual in that it can fold up into the plane with both triangles coinciding. Its motion is a curve that factors even more finely than general Type II cases into three double lines, three quadrics, and four quartics. (The sum of the degrees $3\cdot 2+3\cdot 2+4\cdot 4=28$ is the same as the degree of the irreducible curve in the Type I case.) Numerical irreducible decomposition finds this factorization, and the technique sketched above extracts the real curves inside these complex factors. Poses on two of these motion curves are shown in Figure 5.

7.6 Exceptional Sets

Many problems may be rephrased as a problem of finding the set of parameters where some exceptional behavior occurs. An interesting case in kinematics is finding overconstrained mechanisms, i.e., mechanisms of a given family that have more degrees of freedom than most of the other mechanisms in the family.

A smattering of planar and spatial overconstrained mechanisms are known, including, for example, the Bennett [9] spatial four-bar and the Griffis-Duffy Type

I and II 6-SPS mechanisms mentioned above. However, to date, investigations of overconstrained mechanisms have employed specialized arguments for the specific mechanism family under consideration. The fiber product approach to finding exceptional sets described in [41] has the potential to provide a general approach applicable to many mechanism families. Implementation of that approach in a form that can handle mechanisms of an interesting level of complexity is still a research topic.

8 Software

There are several software packages that compute isolated solutions of polynomial systems, Bertini [5], HOM4PS-2.0 [17], Hompack90 [51] and its extensions [52, 42], and PHCpack [44]. Hompack90 has general parallel tracking facilities. HOM4PS-2.0 has the best implementation of polyhedral methods, but is not a parallel code. Only Bertini and PHCpack implement algorithms of numerical algebraic geometry for positive-dimensional solution sets.

PHCpack and Bertini both allow the user to define their own homotopy and prescribed start points, but HOM4PS-2.0 currently does not.

HOM4PS-2.0 uses only double precision arithmetic to perform computations. To varying degrees, both PHCpack and Bertini have the capability of using higher precision arithmetic. PHCpack does not currently have the capability of adapting the precision based on the local conditioning of the homotopy path. This means that more human interaction is needed to verify that the precision is set appropriately to accurately and reliably perform the requested computations.

The more advanced algorithms of numerical algebraic geometry (including the powerful equation-by-equation methods for finding isolated solutions) place strong requirements on the underlying numerical software [6]. For example, without secure path-tracking and adaptive precision, computing the numerical irreducible decomposition for systems that involve more than a few variables is not possible.

Only Bertini gives the numerical irreducible decomposition directly. Exceptional features of Bertini include:

- secure path-tracking;
- adaptive multiprecision [7, 8];
- utilities for working with polynomial systems given as straight-line programs;
- the numerical irreducible decomposition [37, 40];
- equation-by-equation methods such as regeneration [14, 15];
- local dimension testing [4] (see §7.4); and
- various endgames (see §6.7) including the Cauchy endgame [29], and a parallel endgame based on it [3].

9 Conclusions

This chapter shows how problems in kinematics can be formulated as algebraic systems, thereby introducing the concept of a mechanism space and its associated input and output maps. This provides a framework for understanding the definitions of a variety of kinematics problems, including analysis problems, such as the forward and inverse kinematics problems for robots, and synthesis problems that seek to design mechanisms that produce a desired motion.

Since algebraic kinematics is a subset of algebraic geometry, the computational tools for systems of polynomials can be applied. In particular, numerical algebraic geometry, based on polynomial continuation, has matured into a set of tools for finding and manipulating solution sets of any dimension. Since the bulk of computation is spent tracking a large number of independent homotopy paths, the methods naturally scale to large parallel computing environments, well suited for the needs of 21st-century kinematicians.

References

1. Alt H (1923) Über die erzeugung gegebener ebener kurven mit hilfe des gelenkvierecks. ZAMM, 3(1):13–19
2. Basu S, Pollack R, Roy M-F (2006). Algorithms in real algebraic geometry, Springer-Verlag, Berlin, 2nd edition
3. Bates DJ, Hauenstein JD, Sommese AJ (2011) A parallel endgame. Contemporary Math. 556:25–35
4. Bates DJ, Hauenstein JD, Peterson P, Sommese AJ (2009) A numerical local dimension test for points on the solution set of a system of polynomial equations. SIAM J. Num. Anal. 47:3608–3623
5. Bates DJ, Hauenstein JD, Sommese AJ, Wampler CW (2012) Bertini: Software for numerical algebraic geometry.
   www.nd.edu/∼sommese/bertini. Cited 26 June 2012
6. Bates DJ, Hauenstein JD, Sommese AJ, Wampler CW (2008) Software for numerical algebraic geometry: a paradigm and progress towards its implementation. In: Stillman ME, Takayama N, Verschelde J (eds) Software for Algebraic Geometry, IMA Volumes in Mathematics and Its Applications 148:1–14. Springer, New York
7. Bates DJ, Hauenstein JD, Sommese AJ, Wampler CW (2008) Adaptive multiprecision path tracking. SIAM J. Numer. Anal. 46:722–746
8. Bates DJ, Hauenstein JD, Sommese AJ, Wampler CW (2009) Stepsize control for adaptive multiprecision path tracking. Contemporary Math. 496:21–31
9. Bennett GT (1903) A new mechanism. Engineering 76:777–778
10. Besana GM, Di Rocco S, Hauenstein JD, Sommese AJ, Wampler CW (2012) Cell decomposition of almost smooth real algebraic surfaces (preprint)
11. Bottema O, Roth B (1979) Theoretical kinematics, North-Holland Publishing Co., Amsterdam. Reprinted: Dover Publications, New York, 1990.
12. Dayton B, Zeng Z (2005) Computing the multiplicity structure in solving polynomial systems. In: Proc. of ISSAC 2005, 166–123, ACM, New York
13. Freudenstein F (1973) Kinematics: Past, present and future. Mech. Mach. Theory 8:151-160
14. Hauenstein JD, Sommese AJ, Wampler CW (2011) Regeneration homotopies for solving systems of polynomials. Math. Comp. 80:345–377

15. Hauenstein JD, Sommese AJ, Wampler CW (2011) Regenerative cascade homotopies for solving polynomial systems. Appl. Math. Comp. 218(4):1240–1246
16. Husty ML, Karger A (2000) Self-motions of Griffis-Duffy type parallel manipulators. In: Proc. 2000 IEEE ICRA (CDROM) San Francisco, CA, April 24-28
17. Lee T-L, Li T-Y, Tsai CH (2008) HOM4PS-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method. Computing 83:109–133
18. Leykin A, Verschelde J, Zhao A (2006) Newton's method with deflation for isolated singularities of polynomial systems. Theor. Comp. Sci, 359:111–122
19. Leykin A, Verschelde J, Zhao A (2008) Higher-order deflation for polynomial systems with isolated singular solutions. In: Dickenstein A, Schreyer F-O, Sommese AJ (eds) Algorithms in Algebraic Geometry, 79–97. Springer, New York
20. Li T-Y (1997) Numerical solution of multivariate polynomial systems by homotopy continuation methods. In: Acta numerica 6:399–436 Cambridge Univ. Press, Cambridge
21. Li T-Y (2003) Numerical solution of polynomial systems by homotopy continuation methods. In: Handbook of numerical analysis XI:209–304. North-Holland, Amsterdam
22. Lu Y, Bates D, Sommese AJ, Wampler CW (2007) Finding all real points of a complex curve. In: Proc. Midwest Algebra, Geometry and Its Interactions Conference, Contemporary Math. 448:183–205 AMS
23. McCarthy JM (2000) Geometric design of linkages. Springer, New York
24. Morgan AP (1986) A transformation to avoid solutions at infinity for polynomial systems. Appl. Math. Comput. 18(1):77–86
25. Morgan AP (1987) Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems. Prentice Hall Inc., Englewood Cliffs, NJ
26. Morgan AP, Sommese AJ (1987) A homotopy for solving general polynomial systems that respects *m*-homogeneous structures. Appl. Math. Comput. 24:101–113
27. Morgan AP, Sommese AJ (1987) Computing all solutions to polynomial systems using homotopy continuation. Appl. Math. Comput. 24:115–138. Errata (1992) Appl. Math. Comput. 51:209
28. Morgan AP, Sommese AJ (1989) Coefficient-parameter polynomial continuation. Appl. Math. Comput. 29(2):123–160. Errata (1992) Appl. Math. Comput. 51:207
29. Morgan AP, Sommese AJ, Wampler CW (1991) Computing singular solutions to nonlinear analytic systems. Numer. Math. 58(7):669–684
30. Morgan AP, Sommese AJ, Wampler CW (1995) A product-decomposition bound for Bezout numbers. SIAM J. Numer. Anal. 32(4):1308–1325
31. Ojika T (1987) Modified deflation algorithm for the solution of singular problems. I. A system of nonlinear algebraic equations. J. Math. Anal. Appl. 123:199–221
32. Ojika T, Watanabe S, Mitsui T (1983) Deflation algorithm for the multiple roots of a system of nonlinear equations. J. Math. Anal. Appl. 96:463–479
33. Raghavan M (1993) The Stewart platform of general geometry has 40 configurations. ASME J. Mech. Des. 115:277–282
34. Roberts S (1875) On three-bar motion in plane space. Proc. London Math. Soc. VII:14–23
35. Roth B, Freudenstein F (1963) Synthesis of path-generating mechanisms by numerical means. J. Eng. Industry, Trans. ASME 85(Series B):298–306
36. Sommese AJ, Verschelde J (2000) Numerical homotopies to compute generic points on positive dimensional algebraic sets. J. Complexity, 16(3):572–602
37. Sommese AJ, Verschelde J, Wampler CW (2001) Numerical decomposition of the solution sets of polynomial systems into irreducible components. SIAM J. Numer. Anal. 38(6):2022–2046
38. Sommese AJ, Verschelde J, Wampler CW (2004) Advances in polynomial continuation for solving problems in kinematics. ASME J. Mech. Des. 126:2:262-268
39. Sommese AJ, Wampler CW (1996) Numerical algebraic geometry. In: Renegar J, Shub M, Smale S (eds) The mathematics of numerical analysis (Park City, UT, 1995), Lectures in Appl. Math. 32:749–763. AMS, Providence, RI
40. Sommese AJ, Wampler CW (2005) The numerical solution of systems of polynomials arising in engineering and science. World Scientific, Singapore

41. Sommese AJ, Wampler CW (2008) Exceptional sets and fiber products. Found. Comp. Math. 28:171–196
42. Su H-J, McCarthy JM, Sosonkina M, Watson LT (2006) Algorithm 857: POLSYS_GLP—a parallel general linear product homotopy code for solving polynomial systems of equations. ACM TOMS 32(4):561–579
43. Tsai L-W, Morgan AP (1985) Solving the kinematics of the most general six- and five-degree-of-freedom manipulators by continuation methods. ASME J. Mech., Trans., Auto. Des. 107:48–57
44. Verschelde J (1999) Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. ACM TOMS 25(2):251–276
45. Verschelde J, Cools R (1993) Symbolic homotopy construction. Appl. Algebra Engrg. Comm. Comput. 4(3):169–183
46. Wampler CW, Hauenstein JD, Sommese AJ (2011) Mechanism mobility and a local dimension test. Mech. Mach. Theory 46(9):1193–1206
47. Wampler CW, Morgan AP, Sommese AJ (1990) Numerical continuation methods for solving polynomial systems arising in kinematics. 112:1:59–68
48. Wampler CW, Morgan AP, Sommese AJ (1992) Complete solution of the nine-point path synthesis problem for four-bar linkages. ASME J. Mech. Des. 114:153–159
49. Wampler CW, Morgan AP, Sommese AJ (1997) Complete solution of the nine-point path synthesis problem for four-bar linkages – Closure. ASME J. Mech. Des. 119:150–152
50. Wampler CW, Sommese AJ (2011) Numerical algebraic geometry and algebraic kinematics. Acta Numerica 20:469–567. doi: 10.1017/s0962492911000067
51. Watson LT, Sosonkina M, Melville RC, Morgan AP, Walker HF (1997) Algorithm 777: HOMPACK90: a suite of Fortran 90 codes for globally convergent homotopy algorithms. ACM TOMS 23(4):514–549
52. Wise SM, Sommese AJ, Watson LT (2000) Algorithm 801: POLSYS_PLP: A partitioned linear product homotopy code for solving polynomial systems of equations, ACM TOMS, 26(1):176–200
53. Wohlhart K (1996) Kinematotropic linkages. In: Lenarcic J, Parenti- Castelli V (eds) Recent Advances in Robot Kinematics 359368, Kluwer, Dordrecht