Computing Saddle Graphs via Homotopy Continuation for the Approximate Synthesis of Mechanisms

Aravind Baskar, Mark Plecnik, and Jonathan D. Hauenstein

University of Notre Dame, Notre Dame, IN 46556, USA

Abstract

An approach for approximate kinematic synthesis of mechanisms is proposed in this paper which computes a graph that identifies minima of an objective function as vertices and connections between them as edges. Such a graph is interactively presented to a designer, whereby edges are continuously traversed to navigate families of design candidates in between minima. Candidates are evaluated continuously according to auxiliary considerations for the exploration of design trade-offs. Relevant design specifications tend to be particular per application and are either unclear as how to incorporate into an objective, or clear but with great consequence to the complexity of function evaluation. Computing the aforementioned graphs begins with finding all minima and saddles of an objective function through polynomial homotopy continuation. Connections between minima that minimize their maximum objective value must pass through a saddle to do so. Therefore, after gathering saddles, each is perturbed both ways in its least eigendirection to seed gradient descent paths which connect two minima when pieced together. Discovered connections between minima are organized into a graph, where edges correspond to gradient descent paths.

Keywords: Mechanisms, Optimization, Homotopy continuation, Saddle graph

1. Introduction

Kinematic synthesis aims to find the dimensions of a mechanism after desired constraints have been posed on its motion. For exact synthesis, the number of constraints and dimensional design variables are equal. For approximate synthesis, the former exceeds the latter. Our approach to approximate synthesis begins by constructing an objective function from motion specifications, of which minima and saddles can be computed via homotopy continuation. Minima are subsequently connected by computing gradient descent paths emanating from saddles that are pieced together and organized into a graph which we call a saddle graph. The resulting paths in design space connect two minima while minimizing the maximum objective value along the way (which is the saddle it passes through). A saddle graph organizes stationary points and their connections by representing saddles and minima as vertices and gradient descent paths as edges. After computing and constructing a saddle graph, it may be displayed to a designer for perusal in an interactive and continuous fashion.

The utility of such interactive exploration is augmented by evaluating all design candidates that comprise a saddle graph according to auxiliary performance metrics. To navigate trade-offs, this information is relayed to the designer during interaction with the saddle graph. The value of a saddle graph hinges on the supposition that a designer would recognize greater utility along one of its edges rather than on a minimum vertex. Such cases arise because applications often drive a diverse set of design specifications which together do not lend to a neat objective function. Some specifications have no clear formalization, some do but greatly increase the complexity of function evaluations, and some are omitted due to low priority. Blending such specifications into an application-specific objective leads to intricate optimization that hinders generalizability and is thus not pursued here. Instead, we form the objective from motion specifications, which take a polynomial form, and evaluate auxiliary metrics on an ad hoc basis after saddle graphs have been formed, aiding in generalizability.

1Corresponding author: Aravind Baskar abaskar@nd.edu
1.1. Background

Early works that applied homotopy continuation for the exact synthesis of mechanisms include [1, 2, 3]. This is achieved by finding the roots of a system of polynomial equations that arise naturally from the mathematical model. This body of work has evolved with the development of newer and better techniques to address the problem of finding these roots. As homotopy continuation advanced, it enabled solutions to more complicated problems of exact kinematic synthesis [4, 5, 6, 7]. On a different front, the methods of approximate kinematic synthesis have focused on optimization [8, 9], Fourier descriptors [10], evolutionary algorithms [11, 12], and machine learning [13]. Approximate synthesis allows for the inclusion of a much greater number of motion specifications, and with numerical optimization techniques, inequality constraints can be handled as well. Both search-based and gradient-based algorithms are popular in these works. A less-explored pathway to design mechanisms is to frame the optimization objective and to find all the critical points of the same. In this hybrid framework, the design problem poses an objective and the tools of numerical continuation are used to compute minima and saddles. The general characteristics of optimization problems that can be solved using homotopy continuation are as follows:

1. The objective function must have a finite number of stationary points
2. It is preferable to construct an objective function whose monomial structure is invariant with respect to the number of design specifications, enabling a unified formulation.

Liu and Yang [14], in one of the first works of this kind, solved a class of problems in the design of four-bar mechanisms and reported a root-count of 33 for these optimization problems. More recently, more complicated problems, including the design of systems with two degrees of freedom, can be solved using more advanced numerical continuation techniques [15]. In these works, emphasis was on finding all of the isolated critical points. The current work aims to construct a network of critical points by establishing connections between them leading to a thorough exploration of the design space. Since the construction of these networks are enabled from the computation of saddle points, we refer to these networks as saddle graphs. The construction involves the tracking of gradient-descent paths starting from the saddle points leading into the adjoining minima. This leg of the work has precedence in literature, such as Morse and Morse-Smale complexes [16, 17, 18]. One-dimensional slow invariant manifolds for dynamical systems arising in [19] were used to compute a saddle network in studying the equilibria and slow dynamics in the composition space of reactive systems. In a more recent work [20], a similar network is used to study an energy landscape related to the snappability of pin-jointed bar frameworks. In both these applications, directed graphs connecting saddle points to the adjoining minima are computed similar to our notion of saddle graphs. The novelty of this work is to leverage the construction of saddle graphs to find parametric families of solutions that aid in the development of a design tool in the optimal synthesis of mechanisms.

1.2. Motivation

The goal is to develop a tool that offers continuous families of design candidates to a user. This is important since, from a designer’s standpoint, there are several auxiliary design considerations that are difficult to account for in an objective function such as branching or circuit defects. For a four-bar linkage and other mechanisms derived from four-bar loops, inequality constraints can be derived to detect branching and then implemented with a constrained optimizer [21, 22]. However, for more complex mechanisms, such inequalities are unavailable. Branching defects can be detected through a discretized solution of the direct kinematics which is easy and flexible via post-processing analysis but difficult to force-fit into an objective function. Moreover, the topic of branch and circuit defects has more gray area than is usually considered by past literature. For example, [23, 24] found utility in “defective” mechanisms which produced most but not all precision points. Therefore, the designer’s and application’s tolerance for kinematic inaccuracy could be additionally included into an objective, neither of which information tends to be readily available. Such a tolerance can be modeled by adding weighting factors to the objective, which often leads to hand-tuning until a desired minimum is found. Such hand-tuning activities cast into doubt the correlation between an objective value and true design utility. In [25], this uncertainty is described as “fuzziness” in the design criteria and constraints.

Dimensional sensitivity provides a second example point. Generically, such a metric measures how changes in dimension affect an output motion. In practice, what entails changes in dimension and output motion is application-specific. Sometimes components of the design space gradient with respect to an objective
sufficiently describe sensitivity, sometimes eigenvalues of the design space Hessian are additionally required, and for specific output motions, a full Monte Carlo simulation may be needed. Performing these analyses individually is more or less tractable, but incorporating such activities into an objective is at best cumbersome and at worst intractable.

Another metric that is straightforward to evaluate in post-process but cumbersome to incorporate into an objective is its spatial envelope, which is frequently computed as a maximum length or area over all relevant configurations of a mechanism. Inequality constraints, commonly imposed on link lengths, are naturally handled with interior point methods [26]. In our methodology, we do not handle inequality constraints up front, but instead assess link packaging as auxiliary considerations in post process. Finally, we argue that some design considerations are never formalized and remain latent to the designer’s mind. For example, the designer might expect a certain aesthetic. Although never formalized, these considerations additionally skew the designer’s judgment of utility away from the minima of the objective function. This all motivates the need for delivering families of design candidates that the designer can choose from.

![Process flow diagram of homotopy-based optimization.](image)

**Figure 1:** Process flow diagram of homotopy-based optimization.

### 1.3 Outline

A flow diagram is presented in Fig. 1 which summarizes our saddle graph approach. The rest of the paper is organized as follows. Section 2 provides details regarding homotopy-based optimization and saddle graphs.
Section 3 summarizes four-bar function generation which is applied in Section 4 to a humanoid finger and a flapping motion for a hummingbird. A discussion regarding the results is provided in Section 5 followed by a short conclusion in Section 6.

2. Homotopy-based optimization

The process flow for the mathematical modeling of homotopy-based optimization can be summarized in the following five step procedure:

1. For any given problem specification such as the design of a function generator, the vector loop equations are derived. All the passive configuration variables are eliminated to arrive at a scalar equation referred to as the residue condition hereon. This elimination step is crucial to render the optimization problem invariant to the number of design specifications. Otherwise, the number of variables would increase with the number of specifications. The residue condition is a function of, say, $n$ design variables and the design specifications such as the angular displacements given by the function to be generated.

2. For $N > n$ generic design specifications, it is impossible to meet all conditions exactly with zero residual error. Hence, an objective function is proposed as a sum of squares of the residue condition for the $N$ design specifications. The residue condition and hence objective function are polynomial.

3. The first-order necessary conditions of optimality are derived by setting the partial derivatives of the function with respect to the design vector of dimension $n$ to zero. This always leads into a square system with the same number of equations and variables. The monomial structure of this system of polynomials is one measure of complexity of the underlying system and can be used to dictate the choice of the numerical technique used to find the roots. Note that the monomial structure is invariant with respect to the number of design specifications for a given model.

4. Several standard methods of numerical continuation techniques exist for solving polynomial systems such as a multi-homogeneous homotopy [27], polyhedral homotopy [28], regeneration [29], and monodromy-based techniques [3, 5, 30, 31]. For small systems with, say, expected root counts in the hundreds, a straight-forward multi-homogeneous homotopy is often preferred based on its simplicity. For more complicated systems with a comparatively higher root count, monodromy-based techniques can be quite efficient [32]. A generic design specification is randomly chosen and one of the available techniques is used to compute the critical points. This is called the ab initio step and the solution set found serves as the start points for the subsequent step, namely, parameter homotopy.

5. Once the ab initio step is completed, the polynomial system is considered solved. Any different design specification set for this model can be solved via a parameter homotopy from the start system found in the previous step to the target of interest. By successfully completing this step, all possible critical points of the objective function are known leading to full exploration of the design space. This is one of the main advantages of formulating a polynomial objective function and utilizing numerical continuation techniques over using traditional optimization toolboxes that only find one or a few minima.

2.1. Critical points

Critical points of an optimization problem are generally classified as either a minimum, maximum, or a saddle point based on the definiteness of Hessian matrix of the objective function. If all the eigenvalues of the Hessian matrix evaluated at a critical point are positive, then the critical point is a minimum. On the other hand, if all of the eigenvalues are negative, then the critical point is a maximum. A unified classification scheme is adopted in this work to classify a critical point as a saddle of index $k$ where $k$ is the number of eigenvalues of the Hessian matrix evaluated at the critical point which are negative. By this scheme, for an $n$-dimensional problem, an index 0 saddle is a minimum and an index $n$ saddle is a maximum. This classification scheme simplifies the discussion on saddle graphs.

2.2. Saddle graphs

Suppose that $f(d)$ is the polynomial objective function of the optimization problem where $d$ is the set of $n$ design variables. Let $d^*$ denote a saddle point of index $k$ satisfying the first-order necessary condition for optimality, namely, $f_d(d^*) = 0$ where $f_d$ is the gradient vector of $f$ with respect to $d$. Let $f_{dd}$ be the Hessian matrix of $f$ with respect to $d$. Then, the eigenvalues of $f_{dd}(d^*)$ are referred to as principal curvatures.
and the corresponding eigenvectors are the principal directions of curvature. For \( j = 1, \ldots, k \), let \( \mathbf{e}_j \) be the principal directions corresponding to the negative principal curvatures \( \lambda_j < 0 \) of \( f_{\mathbf{d}}(\mathbf{d}^*) \). Therefore, in a small neighborhood of \( \mathbf{d}^* \), these principal directions with negative principal curvature represent a basis for the subspace of directions for which the objective \( f \) instantaneously decreases.

Consider the following:

\[
\frac{d\mathbf{d}}{dt} = -f_{\mathbf{d}} - \mathbf{d}^*, \quad \frac{d\mathbf{d}}{dt}(0) = s \cdot \mathbf{e}_j,
\]

where \( s \in \{-1, +1\} \). The solution \( \mathbf{d}_s(t) \) to (1) represents a solution to the gradient descent optimization emanating from a saddle point along a negative principal direction. For the problems under consideration, we know \( f \) is bounded below (in fact, nonnegative since a sum of squares) and coercive, i.e., \( \lim_{\mathbf{d} \to \mathbf{d}^*} f(\mathbf{d}) = \infty \), which implies that \( \lim_{t \to \infty} \mathbf{d}_s(t) \) is a saddle point of index at most \( k - 1 \). In particular, all solutions to (1) have bounded length.

Numerically, one typically combines the initial value conditions in (1) into a single condition:

\[
\mathbf{d}(0) = \mathbf{d}^* + \delta \cdot s \cdot \mathbf{e}_j
\]

for some \( 0 < \delta \ll 1 \). Due to numerical error when computing the saddle point \( \mathbf{d}^* \) and corresponding principal direction \( \mathbf{e}_j \), implementing the initial condition using a perturbation, and employing a numerical scheme such as Runge-Kutta-Fehlberg [33, pp. 539-549] for approximating \( \mathbf{d}_s(t) \), the corresponding numerical trajectory will, almost surely, lead to a local minima since local minima are stable solutions of gradient descent. This is not a concern as the goal is to compute paths connecting minima together.

For an index 1 saddle point \( \mathbf{d}^* \), the two gradient descent solutions, denoted \( \mathbf{d}_+(t) \) and \( \mathbf{d}_-(t) \), must lead to two minima (index 0 saddles). In some cases, they both could lead to the same minimum albeit via distinct paths. If the two minima are distinct, the mountain pass theorem [34, p.114] provides that the corresponding path between the two minima and \( \mathbf{d}^* \) is a minimizer of the maximum value of the objective function over the set of smooth paths connecting the two minima. These paths are referred as separatrix lines of valley type in [35]. Note that such a path can easily be constructed when the saddle points are known while constructing such a path is an extremely challenging problem when only the minima are known.

For saddles of index at least 2, there are multiple negative principal directions. In the following analysis, we only consider one principal direction which has the most negative principal curvature. One reason for this consideration is that this direction is dominant in that other orthogonal directions where the principal curvature is larger, i.e., either positive or less negative, is attenuated fast in comparison. Further, due to numerical considerations described above, numerical tracking of gradient descent paths starting from high index saddles almost surely converge to minima. In particular, saddle points of positive index are unstable equilibria for gradient descent and there are practical challenges in numerically approximating such unstable paths.

This process allows for all of the saddle points of index \( \geq 1 \) to be connected with the set of local minima. We represent this connectivity in the form of a graph, which we call the saddle graph. Each edge of this graph represents a continuous parametric family of mechanisms. In particular, this graph provides greater insights by representing the design space extensively as opposed to just examining the minima. Auxiliary design considerations can be used to sift through this graph of solutions to identify candidate designs.

### 2.2.1. Example of a saddle graph

To illustrate the construction of saddle graphs, consider Himmelblau’s function

\[
f(l_1, l_2) = (l_1^2 + l_2 - 11)^2 + (l_1 + l_2^2 - 7)^2
\]

which is bounded below and coercive as shown in Fig. 2a. There are nine critical points: four minima (index 0), four index 1 saddle points, and a local maximum (index 2). The index 1 saddles connect the minima via gradient descent paths as a consequence of the mountain pass theorem forming a roughly quadrilateral shape as shown in Fig. 2b.

For the index 2 saddle (local maximum), Fig. 2b plots the trajectory arising from the principal direction with the most negative principal curvature. We demonstrate why we made this selection by considering various trajectories resulting from small perturbations along any linear combination of the two principal descent directions. Each plot in Fig. 3 shows trajectories with various sizes of perturbations \( \delta \) emanating...
from the same combinations of negative principal directions. Thus, when $\delta$ is sufficiently small, all the descent paths converge with the one along the principal direction with the most negative principal curvature thereby justifying the restriction of saddle graph analysis to the principal direction with the most negative principal curvature.

One observes that the saddle graph captures all the main characteristics of Himmelblau’s function near each minima. Translating to mechanisms, this means that the saddle graph captures the main characteristics of the design space near each minima and potentially opens more choices for the designer. The following demonstrate the utility of saddle graphs in the context of a four-bar function generation.

3. Function generation of a four-bar mechanism

The mathematical model of the optimization follows [35]. Here, we expand on it at greater detail for completeness. Consider the four-bar mechanism shown in Fig. 4. For function generation, the objective is to coordinate the angles $\mu$ and $\psi$ swept by the proximal links in a predefined manner from a home configuration yet to be determined. Since such a function generated by the mechanism does not change when the mechanism is rotated, stretched, or translated, this allows us to fix the two ground pivots at $(0,0)$ and $(1,0)$, thereby removing four free choices from the model. The two proximal links are defined at a home or reference configuration using the variables $(u,v)$ and $(s,t)$, respectively, as shown. The floating coupler link is of length $r$. We consider a rotated configuration of the mechanism where the angular displacement of the two proximal links are $\mu_j$ and $\psi_j$, respectively, which are the design specifications.
Figure 4: Schematic of the function generation problem.

Let us first write a vector loop equation running from the ground-pivot at origin to the other pivot at (1, 0) between the home configuration and the displaced configuration. Considering the rigidity of the coupler link of length \( r \), the following equation can be written:

\[
r^2 - \| R[\mu_j] \cdot (u, v)^\top - (1, 0) - R[\psi_j] \cdot (s, t)^\top \|^2 = 0,
\]

where \( R[\alpha] \) represents 2D rotation matrix of angle \( \alpha \) (say). Note that, this step already removes the passive angle \( \phi_j \) from the model. Upon expansion, a modifier variable \( r_m = \frac{1}{2} \left( -1 + r^2 - s^2 - t^2 - u^2 - v^2 \right) \) is introduced to simplify the equation and reduce the degree of above equation in the variable \( r \). The simplified residue condition is:

\[
\eta_j := r_m + (su + tv) \cos(\mu_j - \psi_j) + (tu - sv) \sin(\mu_j - \psi_j) - s \cos \psi_j + t \sin \psi_j + u \cos \mu_j - v \sin \mu_j = 0.
\]

The above condition is associated with the design position \( j \). The number of variables in the design set is 5, namely, \( d = \{ r_m, s, t, u, v \} \). Hence, for a generic design specification, up to a maximum of 5 design positions can be solved with zero residue. This is the classical exact synthesis problem that admits a maximum of three feasible solutions [36, p. 210]. On the other hand, this work deals with an optimization problem where the number of design positions can be arbitrarily large. Since most practical problems require more than just five positions, it is useful to solve this problem in an optimal fashion.

3.1. Optimization model

The objective function must be one that reduces the residue across all design positions via a sum of squares of the residuals, namely:

\[
f = \frac{1}{2} \sum_{j=1}^{N} \eta_j^2.
\]

A point to note is that the approach does not directly minimize the error in the function generated, instead it relies on the residue condition to indirectly achieve the same. The issue with a direct minimization approach is that \( \mu_j \) and \( \psi_j \) would be variables that proliferate with the number of design positions in such a model with (2) being the constraint as opposed to being the objective. Hence, the proposed objective function is deemed appropriate.

This is an unconstrained optimization problem in the five design variables, \( d = \{ r_m, s, t, u, v \} \). The critical points of the objective function are the points where the gradient of the objective function is zero, which is the necessary condition for optimality, namely:

\[
f_d = \sum_{j=1}^{N} \eta_j \frac{\partial \eta_j}{\partial d} = 0.
\]

This results in a system of five polynomial equations in five variables. Irrespective of the number of design positions, \( N \), the structure of the above set of polynomial equations remain the same, in the sense of the distinct monomials present in them. This means the complexity of the problem is invariant to the
number of design positions, which provides an important advantage. A way of quantifying the complexity of the problem is by finding an upper bound to the maximum number of roots this polynomial system admits. The total degree of the system is $2 \cdot 3^4 = 162$ since the degrees of the five polynomials to solve are $2, 3, 3, 3, 3$, respectively. Hence, the number of critical points to this optimization problem is bounded by 162. Tighter upper bounds exist for these polynomial systems based in algebraic geometry. For example, a multi-homogeneous Bézout number [37] of 53 can be found using the grouping $\{(r_n, s, t), \{u, v\}\}$. There also exists a sharper Bernstein-Kushnirenko-Khovanskii (BKK) bound [37] of 33 (including the trivial solution at the origin) for this system. Note that these numbers are only upper bounds for the maximum number of roots and the actual number called the root count may be lesser than the smallest of these bounds. We refer to Appendix A for more details on the nature of these equations. In the following, a numerical continuation technique is used to estimate the root count for the system of equations formulated in this section via direct computation.

3.2. Ab initio solve

Any general parameterized square system of polynomials admits a finite number of isolated zeroes, called roots. According to the theory of numerical continuation, once the roots of a numerically general [38] version of a target polynomial system have been completely found ab initio, those roots may serve as start points for computationally efficient parameter homotopies to target systems with engineering relevance. This is a consequence of the technique called parameter homotopy [39] in which such a subsequent target system can be solved via a continuous deformation process from the solved system to the unsolved one. Theoretically, with probability one, there exists a one-to-one map between the roots of these two systems.

For the ab initio solve, as described in Item 4 in Section 2, the choice of the technique is usually incumbent upon the scale and complexity of the problem. For relatively simple problems such as the function generation one being discussed here, a multi-homogeneous homotopy is adequate. A generic parameter set $(\mu_j, \psi_j)$ for $j = 1, 2, ..., N$ of random complex numbers is chosen to define the ab initio system to be solved. Using the partition $\{(r_n, s, t), \{u, v\}\}$ of two homogenized groups, Bertini [38, 40] is used to track 53 paths yielding the 25 roots. All the computations of this work are carried out using a Intel® Core™ 2.80 GHz system using a single core. It is of note that one of the 25 solutions is the trivial solution where all five variables take the value zero. Unexpectedly, this seemingly useless trivial solution provides strong utility for the designer via a saddle graph which will be expanded upon in the numerical case studies.

Remark. It is worth pointing out that some critical points of an objective function lie at infinity. For example, $f(x) = x^2$ has a minimum at $x = 0$ and a maximum as $x$ tends to infinity. It is possible to represent critical points at infinity by working with projective coordinates as demonstrated in [19]. However, for the numerical examples considered in this work, the critical points at infinity were degenerate and did not lead to results with meaningful engineering impact. Therefore, further studies on the matter were not pursued.

4. Numerical case studies

In this paper, we consider two numerical case studies. For each, we report on the computation of a parameter homotopy to find all saddles, the computation of eigenvalues of the Hessian matrix to classify each saddle, the computation of gradient descent paths from saddles of index $\geq 1$ perturbed in negative principal directions, the organization of saddles into a graph according to descent path connections yielding a saddle graph, and the evaluation of such graphs according to auxiliary design considerations.

4.1. Humanoid finger

The first case study is that of the design of a humanoid finger, adopted from [35]. This example deals with the design of a constrained 3R humanoid finger with two degrees of freedom. The human finger consists of three phalanges: proximal, middle, and distal phalanx. For most people, the motion between middles and distal phalanxes is coordinated. In building a humanoid finger, their motions can be coupled using a four-bar mechanism by solving a function generation problem. Since the range of motion is continuous, it is desirable to have as many discrete design positions as possible. Thus, this problem falls within the framework of the homotopy-based optimization framework described earlier.

The input specifications are obtained from a video of a human index finger to record the angular displacement of the phalanges. After processing the raw data (refer to [35] for more information), 21 design
positions are chosen (listed in Table 1) from a quadratic curve that fits the data as shown in Fig. 5. Beginning from the 25 start points of the general system solved earlier, a parameter homotopy is used to solve the system associated with the humanoid example in the Bertini software in about 1.5 s. All 25 paths converged successfully and those corresponding to physical linkage geometry are reported in Table 2 yielding the critical points of the design objective function.

4.1.1. Construction of a saddle graph

The saddle graph for this humanoid finger example is presented in Fig. 6. Each edge of this graph represents a continuous family of design candidates in terms of a parameter \( t \) between 0 and 1. The straight line edges shown are only representative. In other words, the connections shown are topologically consistent, however the true connections are 1-dimensional smooth manifolds embedded in a 5-dimensional space. Such connections cannot be visualized in their original space unlike in the case of Himmelblau’s function in Fig. 2b. It is observed that some of these edges offer design solutions that are very distinct from any of the minima. This is particularly interesting along the edges that map to the trivial global minimum solution such as 2-7, 3-7 and 6-7. While the trivial solution is practically useless by itself, the edges leading into them often produce excellent design candidates. One such design returning a very small objective value comparable with even all other local minima barring the trivial solution is shown in Fig. 8. It is a snapshot from a design tool we developed in Wolfram Mathematica to visualize the continuous family of design candidates that occur along the connecting edges of the graph. The design shown is along the edge connecting #6 to the trivial solution #7 (refer to Table 2).

The construction of such a design interface is made possible by the computation of all critical points (including all saddle points) of the optimization problem via homotopy continuation. Auxiliary considerations are taken up at this stage to find practical designs along the edges of this graph. These auxiliary

---

Table 1: Design specification for the function generation problem of a human finger motion.

<table>
<thead>
<tr>
<th>( \mu )</th>
<th>( \psi )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0245</td>
</tr>
<tr>
<td>-0.060</td>
<td>0.0532</td>
</tr>
<tr>
<td>-0.120</td>
<td>0.0860</td>
</tr>
<tr>
<td>-0.180</td>
<td>0.123</td>
</tr>
<tr>
<td>-0.240</td>
<td>0.164</td>
</tr>
<tr>
<td>-0.300</td>
<td>0.210</td>
</tr>
<tr>
<td>-0.360</td>
<td>0.259</td>
</tr>
<tr>
<td>-0.420</td>
<td>0.313</td>
</tr>
<tr>
<td>-0.480</td>
<td>0.371</td>
</tr>
<tr>
<td>-0.540</td>
<td>0.433</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \mu )</th>
<th>( \psi )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.660</td>
<td>0.499</td>
</tr>
<tr>
<td>-0.720</td>
<td>0.569</td>
</tr>
<tr>
<td>-0.780</td>
<td>0.644</td>
</tr>
<tr>
<td>-0.840</td>
<td>0.723</td>
</tr>
<tr>
<td>-0.900</td>
<td>0.806</td>
</tr>
<tr>
<td>-0.960</td>
<td>0.893</td>
</tr>
<tr>
<td>-1.020</td>
<td>0.984</td>
</tr>
<tr>
<td>-1.080</td>
<td>1.080</td>
</tr>
<tr>
<td>-1.140</td>
<td>1.179</td>
</tr>
<tr>
<td>-1.200</td>
<td>1.283</td>
</tr>
</tbody>
</table>
Figure 6: Saddle graph for the function generation optimization to design a humanoid finger. Edges may be traversed continuously to peruse linkage design candidates as they correspond to gradient descent paths in 5D design space.
Table 2: Design candidates for the function generation problem of a human finger data.
Figure 7: Auxiliary considerations visualized in the saddle graph for the design of a humanoid finger.

Figure 8: A feasible design found by analyzing the saddle graph. This design is neither a minimum nor a saddle point, but lies in a 1-dimensional manifold connecting the trivial global minimum #7 and a saddle point #6, found by solving (1).
### 4.2. Hummingbird flapping motion

The second example is taken from [51] which considers the design of a flapping wing mechanism wherein a Watt six-bar linkage was designed to obtain large amplitude of flapping motion about $180^\circ$ for a full-cycle. Refer to doi:10.7274/r781wd40q72 for supplementary resources showcasing a design interface.
of the crank motion. We attempt to design a four-bar mechanism that performs a similar motion as the six-bar mechanism designed in [51]. The design specification obtained from [51] is shown in Table 3. For this problem, the parameter homotopy yielded 5 real solutions which are listed in Table 4.

The saddle graph for this design problem is shown in Fig. 9. In this case, the index 2 saddle connects to the same minimum upon descent along either direction of the eigenvector with the most negative eigenvalue. As in the earlier case, the regions around the trivial solution lead to the most useful set of design candidates accounting for auxiliary considerations as shown in Fig. 10. In this example, the branch defect index plot is generated by using a tolerance of $5^\circ$ in the output value of the function generated in order to identify the primary branch. While no useful solution can be found among the critical points themselves, the path connecting them are found to lead to useful designs, e.g., the path connecting #1 and #5 contains useful designs such as the one shown in Fig. 11.

5. Discussions

For the design of mechanisms, we provide a new methodology to find feasible designs accounting for the primary motion requirement as well as auxiliary considerations such as kinematic defects, packaging and design sensitivity. In this context, we offer a qualitative comparison with Pareto fronts generated through
Figure 10: Auxiliary considerations visualized in the saddle graph for the design of a four-bar to generate the flapping motion of a hummingbird.

Figure 11: A feasible design for creating a flapping motion. This design lies in a 1-dimensional manifold connecting the global minimum #5 and a saddle point #1. It is defect-free and exhibits full-cycle of motion as desired to create a flapping motion.
a multi-objective optimization technique. Firstly, our framework can handle many auxiliary considerations that do not need to be computed quickly. For instance, auxiliary considerations such as the sensitivity index as we define it are computationally expensive. Hence, any conventional multi-objective optimization technique that relies on an iterative algorithm will be time consuming in this regard. Further, multi-objective optimization techniques rely on sophisticated search-based evolutionary algorithms such as [52] since these algorithms do not require gradient information of the objectives. These evolutionary algorithms are stochastic in nature, and depend on hyper parameters such as population size, mutation, and recombination probability. In comparison, the construction of saddle graphs is deterministic in nature. Saddle graphs provides a visual interface to interact with a continuous representation of the design space that competes favorably with Pareto fronts [53] and other trade-off charts such as a “snowflake” plot [25]. Snowflake charts are well-suited to visualize many objectives for a handful of designs, but are ill-suited to visualize many objectives over continuous families of designs.

Further, there is no guarantee in these search based algorithms of ensuring that the feasible design space is sufficiently explored. The authors of [53] note that these design spaces are highly sensitive to the design variables. It underscores the key advantage of the construction of saddle graphs in that the directed gradient-descent paths from saddle points identify the “best” regions of this sensitive design space with respect to the structural error objective. This is made possible by the computation of all the saddle points facilitated by numerical polynomial continuation technique. This makes our design methodology arguably superior when compared with other multi-objective optimization frameworks.

An open research question begs how to scale our work to larger problems, such as the path generation of four-bars. The path generation problem of four-bars and other design problems in six-bars are significantly harder to solve compared to the four-bar function generation example discussed here. The challenge lies in solving higher degree polynomial systems to compute all the critical points of the objective function. For example, the root counts of such systems are on the order of tens of thousands or higher. Parameter homotopy computations of these high degree systems typically suffer from significant numerical failures, which stifles the design process. Development of efficient numerical continuation algorithms that handle such stiff systems without requiring high numerical precision would improve the scalability of our design process.

6. Conclusion

In this paper, an optimization framework to design mechanisms via homotopy continuation is proposed. As the loop closure equations in rigid-body mechanisms are polynomial in nature, a sum of squares of the residue in these equations forms an unconstrained minimization problem of a polynomial objective. The necessary conditions of optimality leads into a system of polynomial equations. Numerical polynomial continuation enables the computation of all the critical points of such systems including the saddle points. The mountain pass theorem guarantees the existence of saddle points in between two minima, forming natural connections in the design space. These connections can be computed via a numerical integration routine starting from the saddle points leading to continuous families of design solutions to the problem at hand that can be represented as a saddle graph. These graphs serve as a platform for building real-time design tools where the continuous families of solutions can be sifted through for auxiliary requirements such as defect-free mechanisms, favorable mechanical advantage characteristics, packaging considerations, etc. This is demonstrated in the context of function generation of four-bar mechanisms. Two different case studies are shown: the design of a humanoid finger mechanism and the design of a mechanism to create flapping motion. These examples showcase the utility of saddle graphs in helping the designer to design mechanisms computationally in a fast and reliable manner.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant Nos. CMMI-2041789, CMMI-2144732, and CCF-1812746. We acknowledge the valuable critique from anonymous referees which aided in improving the quality of the paper.
Appendix A. Equations of optimal four-bar function generation

The following equations are the expanded form of (5) described in Section 3.1. Here, the design variables are \( r_m, s, t, u, v \) and the coefficients are \( c_1, c_2, ..., c_{40} \).

\[
2c_{40}r_m + c_{37}s + c_{34}t + c_{33}u + c_{38}su + c_{36}tu + c_{32}v + c_{38}sv + c_{35}tv = 0;
\]

\[
c_{37}r_m + 2c_{26}s + c_{29}t + c_{17}u + c_{39}r_m u + 2c_{29}su + c_{23}tu + c_{19}u^2 + 2c_{31}su^2 + c_{25}tu^2 + c_{15}v + c_{38}r_m v
\]

\[
+ 2c_{27}sv + c_{21}tv + c_{18}uw + 2c_{30}suw + c_{44}tuw + c_{16}v^2 + 2c_{28}sv^2 + c_{22}tv^2 = 0;
\]

\[
c_{34}r_m + c_{29}s + 2c_{28}t + c_{6}u + c_{36}r_m u + c_{23}su + 2c_{12}tu + c_{8}u^2 + 2c_{25}su^2 + 2c_{14}tu^2 + c_{4}v + c_{35}r_m v
\]

\[
+ c_{21}sv + 2c_{10}tv + c_{7}uv + 2c_{24}suw + c_{13}tuw + c_{5}v^2 + 2c_{22}sv^2 + 2c_{11}tv^2 = 0;
\]

\[
c_{33}r_m + c_{17}s + c_{39}r_m s + c_{29}s^2 + c_{6}t + c_{36}r_m t + c_{23}st + c_{12}t^2 + 2c_{3}u + 2c_{19}su + 2c_{31}s^2 u
\]

\[
+ c_{28}tu + 2c_{25}stu + 2c_{44}t^2 u + c_{2}v + c_{18}sv + c_{40}s^2 v + c_{7}tv + c_{24}sv + c_{13}t^2 v = 0;
\]

\[
c_{32}r_m + c_{15}s + c_{38}r_m s + c_{27}s^2 + c_{4}t + c_{35}r_m t + c_{21}st + c_{10}t^2 + c_{2}u + c_{18}su + c_{30}s^2 u + c_{7}tu
\]

\[
+ c_{24}stu + c_{13}t^2 u + 2c_{1}v + 2c_{16}sv + 2c_{28}s^2 v + 2c_{5}tv + 2c_{23}sv + 2c_{11}t^2 v = 0,
\]

where the coefficients are \( c_1, c_2, ..., c_{40} \) are functions of the design specifications \( (\mu_j, \psi_j) \) for \( j = 1, 2, ..., N \).

For example, \( c_1 = \frac{1}{2} \sum_{j=1}^{N} \sin^2 \mu_j \), \( c_2 = -\sum_{j=1}^{N} \sin \mu_j \cos \mu_j \). Due to the algebraic relations among the coefficients, the root count of the system (25) is less than the BKK bound (33 which includes the trivial solution at the origin) which would be the case were the coefficients algebraically independent.

References


