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

Aravind Baskar, Mark Plecnik, and Jonathan D. Hauenstein[1]

*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.

[1]Corresponding 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.

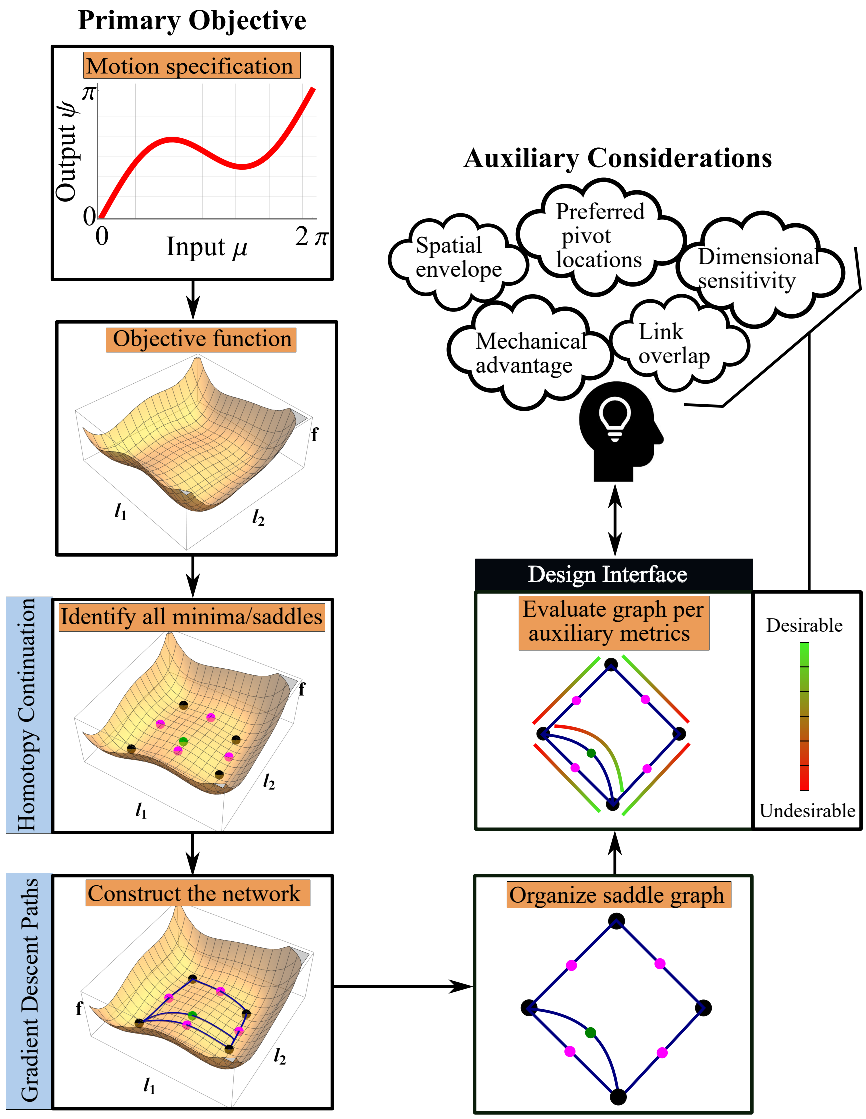



Figure 1: Process flow diagram of homotopy-based optimization.

### 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 [4, 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 $f(\mathbf{d})$ is the polynomial objective function of the optimization problem where $\mathbf{d}$ is the set of $n$ design variables. Let $\mathbf{d}^*$ denote a saddle point of index $k$ satisfying the first-order necessary condition for optimality, namely, $f_{\mathbf{d}}(\mathbf{d}^*) = \mathbf{0}$ where $f_{\mathbf{d}}$ is the gradient vector of $f$ with respect to $\mathbf{d}$. Let $f_{\mathbf{dd}}$ be the Hessian matrix of $f$ with respect to $\mathbf{d}$. Then, the eigenvalues of $f_{\mathbf{dd}}(\mathbf{d}^*)$ 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 $f_{\mathbf{dd}}(\mathbf{d}^*)$. 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:

$$\frac{d\mathbf{d}}{dt} = -f_{\mathbf{d}}, \quad \mathbf{d}(0) = \mathbf{d}^*, \quad \frac{d\mathbf{d}}{dt}(0) = s \cdot \mathbf{e}_j, \tag{1}$$

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., $\lim_{\|\mathbf{d}\|\to\infty} f(\mathbf{d}) = \infty$, 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 [18]. 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.

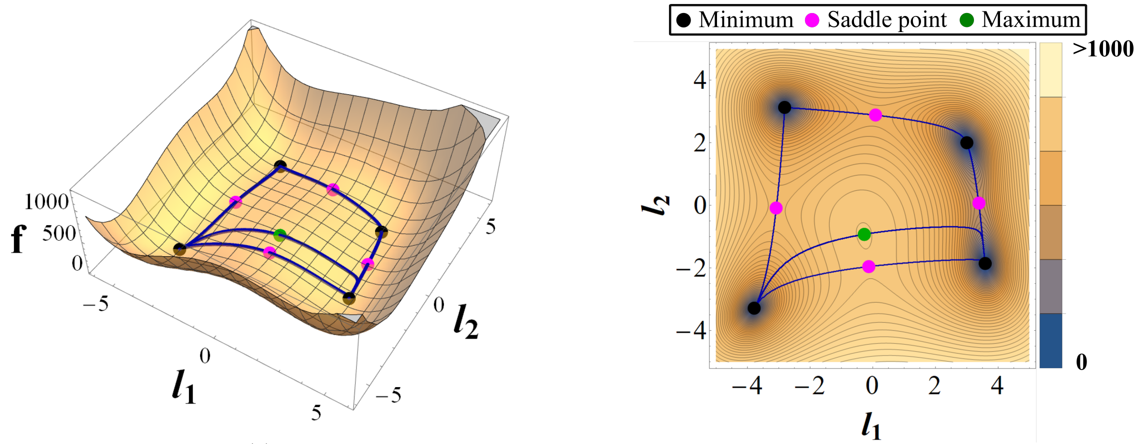
*2.2.1. Example of a saddle graph*

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




(a)

(b)

Figure 2: Illustration of the saddle graph for Himmelblau's function: (a) A 3-dimensional plot of Himmelblau's function with its saddle graph; (b) A 2-dimensional contour plot of Himmelblau's function with its saddle graph.

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.

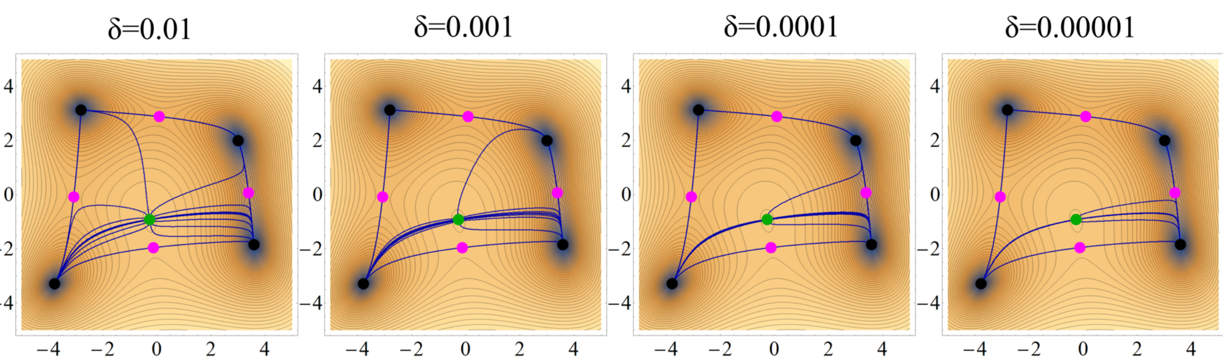



Figure 3: Evolution of finitely many descent paths starting from the index 2 saddle against step length $\delta$ of the initial perturbation.

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.

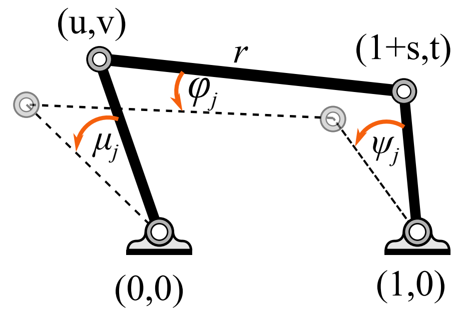



Figure 4: Schematic of the function generation problem.

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:

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

where $\mathbf{R}[\alpha]$ 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:

$$\eta_j := r_m + (su+tv)\cos(\mu_j - \psi_j) + (tu - sv)\sin(\mu_j - \psi_j) - s\cos\psi_j + t\sin\psi_j + u\cos\mu_j - v\sin\mu_j = 0. \tag{3}$$

The above condition is associated with the design position $j$. The number of variables in the design set is 5, namely, $\mathbf{d} = \{r_m, s, t, u, v\}$. 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:

$$f = \frac{1}{2}\sum_{j=1}^{N} \eta_j^2. \tag{4}$$

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 (3) 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, $\mathbf{d} = \{r_m, s, t, u, v\}$. 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:

$$f_{\mathbf{d}} = \sum_{j=1}^{N} \eta_j \frac{\partial \eta_j}{\partial \mathbf{d}} = \mathbf{0}. \tag{5}$$

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 $\{\{r_m, s, t\}, \{u, v\}\}$. There also exists a sharper Bernshtein-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 $\{\{r_m, s, t\}, \{u, v\}\}$ 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

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| $\mu$ | 0 | -0.060 | -0.120 | -0.180 | -0.240 | -0.300 | -0.360 | -0.420 | -0.480 | -0.540 | -0.600 |
| $\psi$ | 0 | 0.0245 | 0.0532 | 0.0860 | 0.123 | 0.164 | 0.210 | 0.259 | 0.313 | 0.371 | 0.433 |
| | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | |
| $\mu$ | -0.660 | -0.720 | -0.780 | -0.840 | -0.900 | -0.960 | -1.020 | -1.080 | -1.140 | -1.200 | |
| $\psi$ | 0.499 | 0.569 | 0.644 | 0.723 | 0.806 | 0.893 | 0.984 | 1.080 | 1.179 | 1.283 | |

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

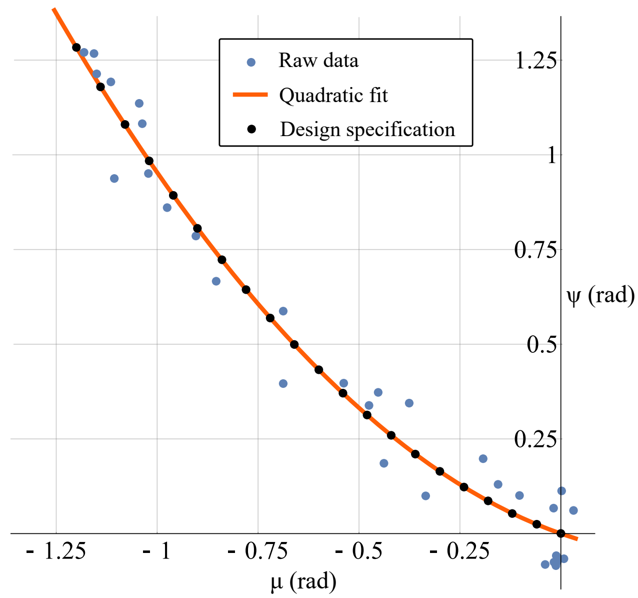



Figure 5: Data processing to arrive at the design specification for the function generation problem of a human finger motion.

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 [41] 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

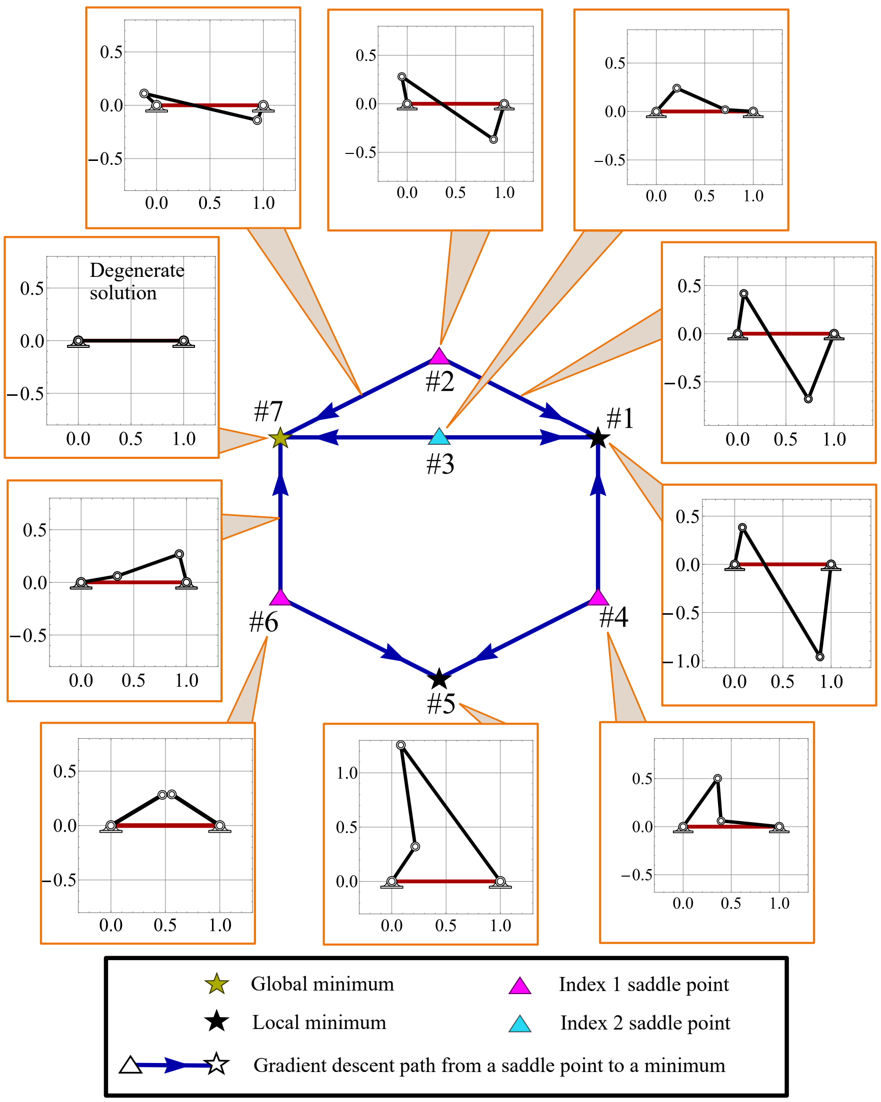



Figure 6: Saddle graph for the function generation optimization to design a humanoid finger. Edges may be traversed continuously to peruse linkage design candidates as they correspond to gradient descent paths in 5D design space.

| # | Design parameters (m) $r$ | $s$ | $t$ | $u$ | $v$ | Type | Saddle index | Max. error[†] (rad) | Obj. $f$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.553 | −0.118 | −0.948 | 0.079 | 0.384 | Min. | 0 | 0.010 | 0.00003 |
| 2 | 1.146 | −0.103 | −0.369 | −0.055 | 0.280 | Saddle | 1 | 0.055 | 0.00012 |
| 3 | 0.546 | −0.289 | 0.028 | 0.153 | 0.281 | Saddle | 2 | N/A | 0.00448 |
| 4 | 0.443 | −0.607 | 0.049 | 0.359 | 0.502 | Saddle | 1 | 0.025 | 0.00007 |
| 5 | 0.945 | −0.913 | 1.259 | 0.216 | 0.321 | Min. | 0 | N/A | 0.00003 |
| 6 | 0.087 | −0.428 | 0.307 | 0.447 | 0.320 | Saddle | 1 | N/A | 0.00005 |
| 7 | 1 | 0 | 0 | 0 | 0 | Min.[*] | 0 | N/A | 0 |

∗ Global minimum.

† N/A denotes a defective design.

Table 2: Design candidates for the function generation problem of a human finger data.

considerations provide a justification for the design candidate presented in Fig. 8.

#### *4.1.2. Auxiliary considerations of the design process*

The discussion on saddle graphs so far has primary considered only the objective function value of the optimization. The computed edges represent paths between saddle points and adjoining minima in a way that minimizes the objective function. However, as mentioned earlier, there may be other considerations that are very pertinent to the design process. The following discusses a few which are shown in Fig. 7.

a) **Branch/circuit defect index** $k_b$: Linkages found through kinematic synthesis are often plagued by branch and circuit defects [42]. Engineers have proposed strategies in literature to analyze and design mechanisms free of these defects [43, 44, 45]. These defects occur as a consequence of the fact that parallel mechanisms admit multiple assembly modes for the same input, e.g., elbow-up and elbow-down configurations in the case of four-bar mechanisms. Hence, in some cases, two branches of configurations occur separated by a *singular* configuration. In the case of a branch defect, the synthesized motion is divided by a singularity. In the case of a circuit defect, the synthesized motion is divided onto disconnected configuration assembly modes. It is useful to evaluate the extent that such defects occur on the saddle graph itself. To do so, we determine which branch or circuit each approximate function point belongs to, then determine which branch or circuit possesses the most design points. The percentage of function points belonging to this branch or circuit is used as a metric of desirability. For instance, if all of the design specifications occur on the same branch, then the candidate is very desirable ($k_b = 1$). It is possible that a design position does not belong to either branch. Since this is an approximate synthesis procedure, no design position is expected to occur on any of the branches with zero residue. We pick a small error tolerance (say 2.5°) in the output value of the function generated to determine the percentage of design positions close to each branch and pick the maximum as the index.

b) **Polygon diagonal index** $k_p$: Another useful characteristic to look at corresponds with the overall sizing of the mechanism. The size of a four-bar mechanism is based on its four pivots, two moving and two fixed. A natural choice of characteristic length is the largest diagonal (including the sides) of the quadrilateral defined by these pivots over the desired range of motion of the mechanism. Unlike the branch defect index $k_b$, the polygon diagonal index $k_p$ is a relative index scaled to lie between 0 and 1 where 1 corresponds with the desirable scenario of having the smallest diagonal length among all the designs that constitute the saddle graph. Due to the relative nature of this measure, a mechanism with $k_p = 0$ may still be useful.

c) **Maximum length index** $k_l$: Maximum length of the mechanism is also worth considering when it comes to sizing. This measure is not dependent on the range of motion but only the link lengths. In reference to the schematic in Fig. 4, this can be expressed mathematically as:

$$\max(1, \sqrt{u^2 + v^2}, r, \sqrt{s^2 + t^2}). \tag{6}$$

This also is a relative index scaled to fit between 0 and 1 where 1 corresponds with the mechanism having the smallest maximum length among all the designs that constitute the saddle graph.

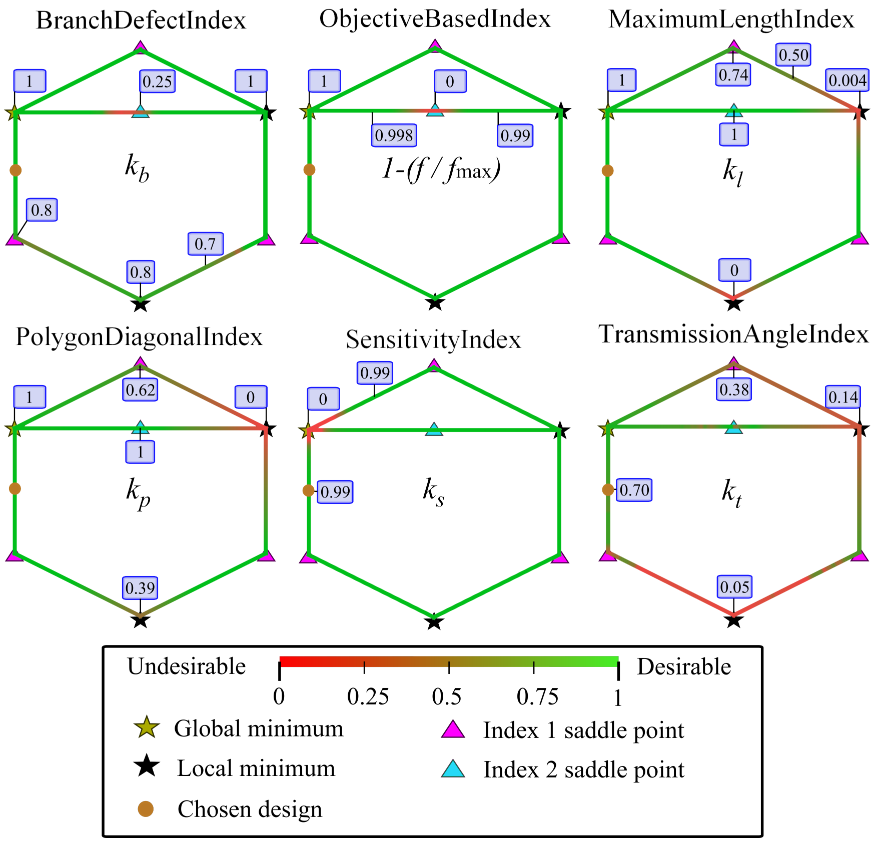
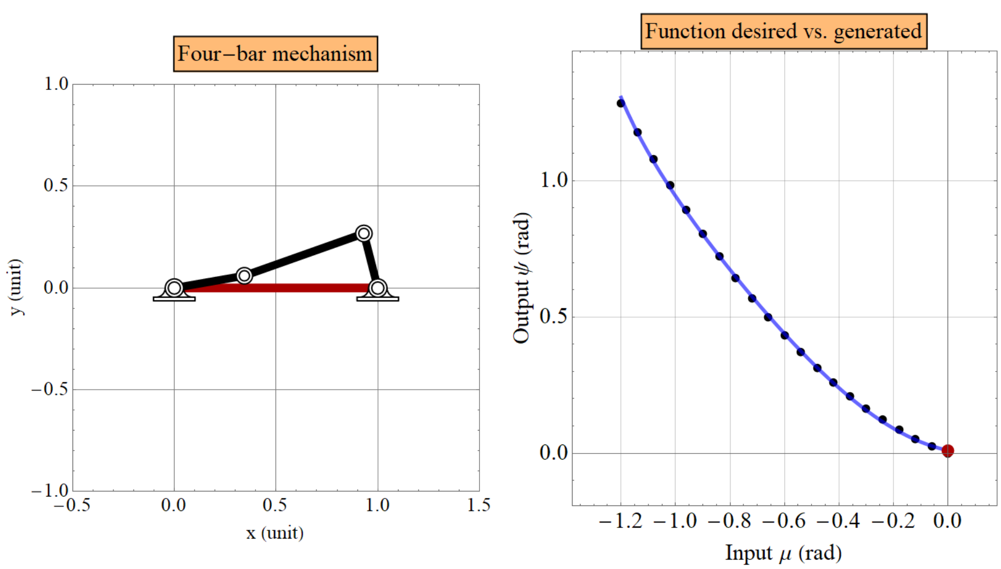



Figure 7: Auxiliary considerations visualized in the saddle graph for the design of a humanoid finger.




| | |
|---|---|
| Obj. value | $1.15\times10^{-5}$ |
| Gradient $L_\infty$ norm | $3.02\times10^{-5}$ |

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).

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| $\mu$ | 0 | $\frac{\pi}{10}$ | $\frac{\pi}{5}$ | $\frac{3\pi}{10}$ | $\frac{2\pi}{5}$ | $\frac{\pi}{2}$ | $\frac{3\pi}{5}$ | $\frac{7\pi}{10}$ | $\frac{4\pi}{5}$ | $\frac{9\pi}{10}$ | $\pi$ |
| $\psi$ | -0.613 | -0.999 | -1.212 | -1.241 | -1.100 | -0.800 | -0.400 | 0.003 | 0.356 | 0.644 | 0.866 |
| | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | |
| $\mu$ | $\frac{11\pi}{10}$ | $\frac{6\pi}{5}$ | $\frac{13\pi}{10}$ | $\frac{7\pi}{5}$ | $\frac{3\pi}{2}$ | $\frac{8\pi}{5}$ | $\frac{17\pi}{10}$ | $\frac{9\pi}{5}$ | $\frac{19\pi}{10}$ | $2\pi$ | |
| $\psi$ | 1.019 | 1.102 | 1.114 | 1.061 | 0.947 | 0.774 | 0.536 | 0.225 | -0.167 | -0.613 | |

Table 3: Design specification for a mechanism to mimic hummingbird flapping motion.

d) **Transmission angle index** $k_t$: The transmission angle is defined as the angle between the coupler and the follower link of a four-bar mechanism [46], which appears frequently in the relevant literature [47, 48]. A transmission angle of $\pm\frac{\pi}{2}$ is the most desirable while 0 or $\pi$ is undesirable in terms of input-output velocity transmission. This essentially provides a measurement on the proximity of a mechanism to a singular configuration. The index $k_t$ itself is computed by finding the minimum value of the *sine* of the transmission angle over the range of motion of the design candidate. It is also a relative index that is scaled to take values between 0 and 1 where 1 is the most desirable.

e) **Sensitivity index** $k_s$: From a designer's standpoint, a mechanism is desirable if the output motion is less sensitive to small deviations in link dimensions. This makes the mechanism robust to manufacturing errors. There are several ways of defining the sensitivity of linkages to deviations in link dimensions, e.g., [49, 50]. Following [24], a direct way of measuring the sensitivity of a design is by perturbing the design variables $\{r, s, t, u, v\}$ within an assumed manufacturing tolerance of 0.00012 mm (standard deviation), that corresponds to CNC machining tolerance. A sample size of 100 perturbations is generated for each design via a multi-normal distribution and the mean percentage deviation of the objective function over the sample is considered. For illustration, if $f_0$ is the objective value of a design $\boldsymbol{d}$ and $f_k, k = 1, 2, ..., 100$ are the objective values of the perturbed designs $\boldsymbol{d}_k, k = 1, 2, ..., 100$, respectively, then the sensitivity index is given by:

$$\frac{1}{100}\sum_{k=1}^{100}\frac{|f_k - f_0|}{f_0}.$$

This is then mapped to a linear scale such that $k_s = 0$ corresponds to the most sensitive design and $k_s = 1$ represents the least sensitive design.

In addition to these indices, the objective function value $f$ associated with the saddle graph is also shown in Fig. 7 via a normalized index $1 - \left(\frac{f}{f_{\max}}\right)$. Note that the region around the index 2 saddle shows the maximum objective value and the rest of the graph is comparable on this metric alone. Each of these sub-plots can be associated with the saddle graph. These sub-plots are generated based on a discretization scheme of fixed step length of 0.025 along each edge of the saddle graph. The computation time required for creating these sub-plots is about 10 s. It can be readily identified from this visualization that the edge $6 - 7$ provides the best overall traits. Note that while the objective function decreases in value monotonically as we traverse from #6 to #7, there is a trade-off in the fact that the mechanism degenerates with the two proximal links collapsing to zero-length at the global minimum #7, in the neighbourhood of which the sensitivity index is poor. This justifies the choice of the design shown in Fig. 8. Effective implementation of all the numerical computation of this allows for the quick development of a graphical user interface in the Wolfram Mathematica[2] environment. It may be noted that the list of performance indices and their definitions may not apply generally to all design scenarios and they may be modified based on the application at hand. Nevertheless, other considerations can be easily accommodated within this framework.

*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° for a full-cycle

[2]Refer to doi:10.7274/r781wd40q72 for supplementary resources showcasing a design interface.

| # | Design parameters (m) | | | | | Type | Saddle index | Obj. |
|---|---|---|---|---|---|---|---|---|
| | $r$ | $s$ | $t$ | $u$ | $v$ | | | $f$ |
| 1 | 0.674 | −0.963 | 0.308 | 0.644 | 0.173 | Saddle | 1 | 0.117554 |
| 2 | 0.682 | −0.907 | −0.196 | −0.207 | 0.477 | Saddle | 1 | 0.119302 |
| 3 | 0.633 | −0.751 | 0.085 | 0.277 | 0.121 | Saddle | 2 | 0.170992 |
| 4 | 0.665 | −1.157 | 0.033 | 0.459 | 0.362 | Min. | 0 | 0.0907026 |
| 5 | 1 | 0 | 0 | 0 | 0 | Min.* | 0 | 0 |

$*$ Global minimum.

Table 4: Real critical points for the design problem to create a flapping motion.

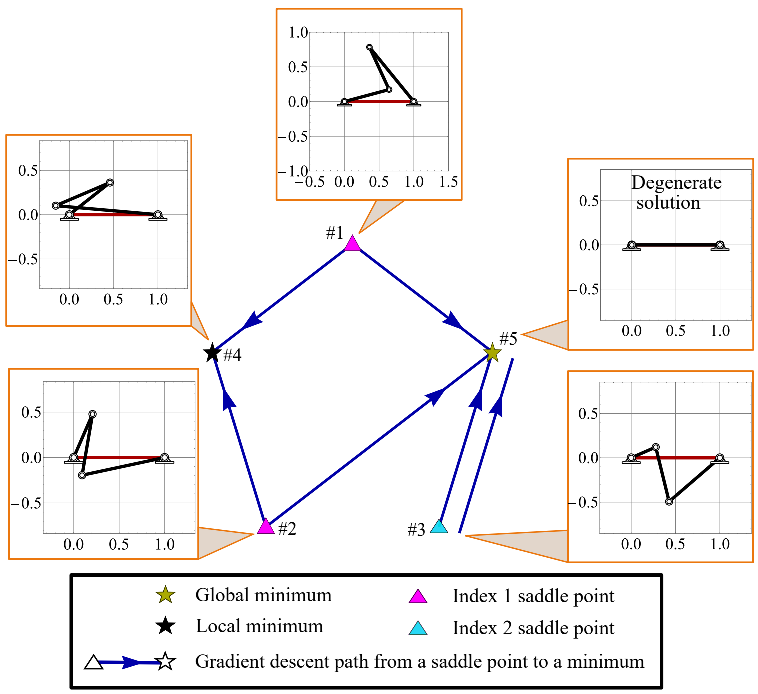



Figure 9: Saddle graph for the function generation optimization to design a mechanism that creates a flapping motion.

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° 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

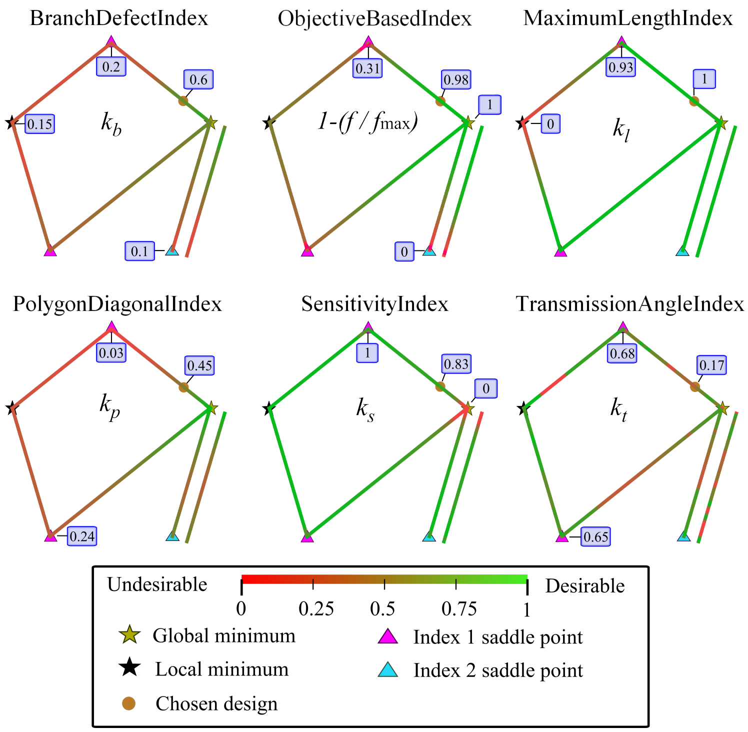



Figure 10: Auxiliary considerations visualized in the saddle graph for the design of a four-bar to generate the flapping motion of a hummingbird.

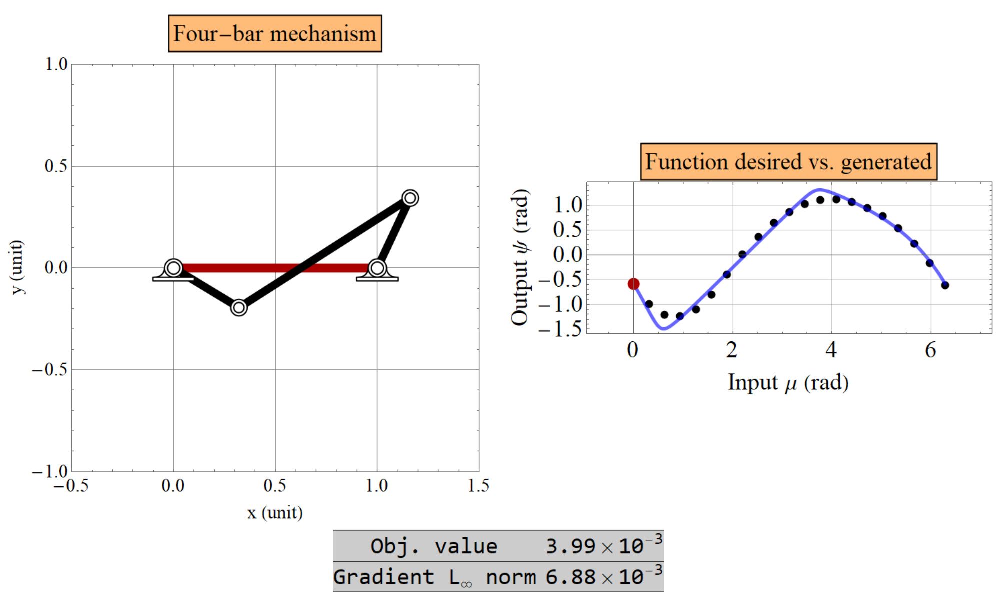



Figure 11: A feasible design for creating a flapping motion. This design lies in a 1-dimensional manifold connecting the global minimum #5 and a saddle point #1. It is defect-free and exhibits full-cycle of motion as desired to create a flapping motion.

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 $c_1, c_2, ..., c_{40}$.

$$
\begin{aligned}
2c_{40}r_m + c_{37}s + c_{34}t + c_{33}u + c_{39}su + c_{36}tu + c_{32}v + c_{38}sv + c_{35}tv = 0;\\
c_{37}r_m + 2c_{26}s + c_{20}t + c_{17}u + c_{39}r_m u + 2c_{29}su + c_{23}tu + c_{19}u^2 + 2c_{31}su^2 + c_{25}tu^2 + c_{15}v + c_{38}r_m v\\
+2c_{27}sv + c_{21}tv + c_{18}uv + 2c_{30}suv + c_{24}tuv + c_{16}v^2 + 2c_{28}sv^2 + c_{22}tv^2 = 0;\\
c_{34}r_m + c_{20}s + 2c_9t + c_6u + c_{36}r_m u + c_{23}su + 2c_{12}tu + c_8u^2 + c_{25}su^2 + 2c_{14}tu^2 + c_4v + c_{35}r_m v\\
+c_{21}sv + 2c_{10}tv + c_7uv + c_{24}suv + 2c_{13}tuv + c_5v^2 + c_{22}sv^2 + 2c_{11}tv^2 = 0;\\
c_{33}r_m + c_{17}s + c_{39}r_m s + c_{29}s^2 + c_6t + c_{36}r_m t + c_{23}st + c_{12}t^2 + 2c_3u + 2c_{19}su + 2c_{31}s^2u\\
+2c_8tu + 2c_{25}stu + 2c_{14}t^2u + c_2v + c_{18}sv + c_{30}s^2v + c_7tv + c_{24}stv + c_{13}t^2v = 0;\\
c_{32}r_m + c_{15}s + c_{38}r_m s + c_{27}s^2 + c_4t + c_{35}r_m t + c_{21}st + c_{10}t^2 + c_2u + c_{18}su + c_{30}s^2u + c_7tu\\
+c_{24}stu + c_{13}t^2u + 2c_1v + 2c_{16}sv + 2c_{28}s^2v + 2c_5tv + 2c_{22}stv + 2c_{11}t^2v = 0,
\end{aligned}
$$

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


[1] B. Roth, F. Freudenstein, Synthesis of path-generating mechanisms by numerical methods, Journal of Engineering for Industry 85 (3) (1963) 298–304.

[2] C. Wampler, A. Morgan, A. Sommese, Complete solution of the nine-point path synthesis problem for four-bar linkages, Journal of Mechanical Design 114 (1) (1992) 153–159.

[3] A. K. Dhingra, J. C. Cheng, D. Kohli, Synthesis of six-link, slider-crank and four-link mechanisms for function, path and motion generation using homotopy with m-homogenization, Journal of Mechanical Design 116 (4) (1994) 1122–1131.

[4] M. M. Plecnik, R. S. Fearing, Finding only finite roots to large kinematic synthesis systems, Journal of Mechanisms and Robotics 9 (2) (2017) 021005.

[5] A. Baskar, S. Bandyopadhyay, An algorithm to compute the finite roots of large systems of polynomial equations arising in kinematic synthesis, Mechanism and Machine Theory 133 (2019) 493–513.

[6] M. M. Plecnik, J. M. McCarthy, Computational design of Stephenson II six-bar function generators for 11 accuracy points, Journal of Mechanisms and Robotics 8 (1) (2016) 011017.

[7] M. M. Plecnik, J. M. McCarthy, Kinematic synthesis of Stephenson III six-bar function generators, Mechanism and Machine Theory 97 (2016) 112–126.

[8] S. Krishnamurty, D. A. Turcic, Optimal synthesis of mechanisms using nonlinear goal programming techniques, Mechanism and Machine Theory 27 (5) (1992) 599–612.

[9] S. Deshpande, A. Purwar, A task-driven approach to optimal synthesis of planar four-bar linkages for extended Burmester problem, Journal of Mechanisms and Robotics 9 (6) (2017) 061005.

[10] I. Ullah, S. Kota, Optimal synthesis of mechanisms for path generation using Fourier descriptors and global search methods, Journal of Mechanical Design 119 (4) (1997) 504–510.

[11] P. Shiakolas, D. Koladiya, J. Kebrle, On the optimum synthesis of six-bar linkages using differential evolution and the geometric centroid of precision positions technique, Mechanism and Machine Theory 40 (3) (2005) 319–335.

[12] J. Cabrera, A. Ortiz, F. Nadal, J. Castillo, An evolutionary algorithm for path synthesis of mechanisms, Mechanism and Machine Theory 46 (2) (2011) 127–141.

[13] S. Deshpande, A. Purwar, A machine learning approach to kinematic synthesis of defect-free planar four-bar linkages, Journal of Computing and Information Science in Engineering 19 (2) (2019) 021004.

[14] A.-X. Liu, T.-L. Yang, Finding all solutions to unconstrained nonlinear optimization for approximate synthesis of planar linkages using continuation method, Journal of Mechanical Design 121 (3) (1999) 368–374.

[15] A. Baskar, C. Liu, M. M. Plecnik, J. D. Hauenstein, Designing rotary linkages for polar motions, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, September 27- October 1, Prague, Czech Republic, 2021.

[16] J. Milnor, Morse theory, Annals of Mathematics Studies, No. 51, Princeton University Press, Princeton, N.J., 1963, based on lecture notes by M. Spivak and R. Wells.

[17] Y. Matsumoto, An introduction to Morse theory, Vol. 208 of Translations of Mathematical Monographs, American Mathematical Society, Providence, RI, 2002, translated from the 1997 Japanese original by Kiki Hudson and Masahico Saito, Iwanami Series in Modern Mathematics.

[18] L. Čomić, L. D. Floriani, L. Papaleo, Morse-Smale decompositions for modeling terrain knowledge, in: International Conference on Spatial Information Theory, Springer, 2005, pp. 426–444.

[19] A. N. Al-Khateeb, J. M. Powers, S. Paolucci, A. J. Sommese, J. A. Diller, J. D. Hauenstein, J. D. Mengers, One-dimensional slow invariant manifolds for spatially homogenous reactive systems, The Journal of Chemical Physics 131 (2) (2009) 024118.

[20] G. Nawratil, Snappability and singularity-distance of pin-jointed body-bar frameworks, Mechanism and Machine Theory 167 (2022) 104510.

[21] R. Singh, H. Chaudhary, A. K. Singh, A loop-by-loop defect rectification procedure for optimal synthesis of Stephenson III path generators, Meccanica 54 (11) (2019) 1869–1888.

[22] W.-M. Hwang, Y.-J. Chen, Defect-free synthesis of Stephenson-III motion generators, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 222 (12) (2008) 2485–2494.

[23] A. Baskar, M. Plecnik, Synthesis of six-bar timed curve generators of Stephenson-type using random monodromy loops, Journal of Mechanisms and Robotics 13 (1) (2021) 011005.

[24] A. Baskar, M. Plecnik, Synthesis of Watt-type timed curve generators and selection from continuous cognate spaces, Journal of Mechanisms and Robotics 13 (5) (2021) 051003.

[25] E. Sandgren, A multi-objective design tree approach for the optimization of mechanisms, Mechanism and Machine Theory 25 (3) (1990) 257–272.

[26] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical Programming 106 (1) (2006) 25–57.

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

[28] B. Huber, B. Sturmfels, A polyhedral method for solving sparse polynomial systems, Mathematics of computation 64 (212) (1995) 1541–1555.

[29] J. D. Hauenstein, A. J. Sommese, C. W. Wampler, Regeneration homotopies for solving systems of polynomials, Math. Comp. 80 (273) (2011) 345–377.

[30] T. Duff, C. Hill, A. Jensen, K. Lee, A. Leykin, J. Sommars, Solving polynomial systems via homotopy continuation and monodromy, IMA Journal of Numerical Analysis 39 (3) (2019) 1421–1446.

[31] J. D. Hauenstein, L. Oeding, G. Ottaviani, A. J. Sommese, Homotopy techniques for tensor decomposition and perfect identifiability, Journal für die reine und angewandte Mathematik (Crelles Journal) 2019 (753) (2019) 1–22.

[32] J. D. Hauenstein, S. N. Sherman, Using monodromy to statistically estimate the number of solutions, in: 2nd IMA Conference on Mathematics of Robotics (2020), Vol. 21, Springer Proceedings in Advanced Robotics, 2022, pp. 37–46.

[33] D. Kincaid, E. W. Cheney, Numerical analysis: mathematics of scientific computing, American Mathematical Soc., 2009.

[34] E. Rimon, J. Burdick, The Mechanics of Robot Grasping, Cambridge University Press, 2019.

[35] A. Baskar, M. M. Plecnik, Computing all solutions to a discretization-invariant formulation for optimal mechanism design, in: 2021 International Conference on Robotics and Automation, May 30- June 5, Xi'an, China, 2021.

[36] G. Sandor, A. Erdman, Advanced Mechanical Design: Analysis and Synthesis, Vol. 2, Prentice-Hall, INC., Englewood Cliffs, New Jersey, 1984.

[37] C. W. Wampler, Isotropic coordinates, circularity, and Bézout numbers: planar kinematics from a new perspective, in: Proceedings of ASME Design Engineering Technical Conference and Computers in Engineering Conference, Irvine, California, August, 18-22, 1996.

[38] D. J. Bates, J. D. Hauenstein, A. J. Sommese, C. W. Wampler, Numerically solving polynomial systems with Bertini, SIAM, Philadelphia, 2013.

[39] A. P. Morgan, A. J. Sommese, Coefficient-parameter polynomial continuation, Appl. Math. Comput. 29 (2, part II) (1989) 123–160.

[40] D. J. Bates, J. D. Hauenstein, A. J. Sommese, C. W. Wampler, Bertini: Software for numerical algebraic geometry, Available at `bertini.nd.edu`.

[41] Wolfram Research Inc., Mathematica, Version 12.1.0.0, Champaign, IL, 2021.

[42] S. S. Balli, S. Chand, Defects in link mechanisms and solution rectification, Mechanism and Machine Theory 37 (9) (2002) 851–876.

[43] Q. Shen, W.-T. Lee, K. Russell, On adjustable planar four-bar motion generation with order, branch and circuit defect rectification, Journal of Mechanisms and Robotics 7 (3) (2015) 034501.

[44] K. Gupta, A. Beloiu, Branch and circuit defect elimination in spherical four-bar linkages, Mechanism and Machine Theory 33 (5) (1998) 491–504.

[45] S. Deshpande, A. Purwar, A machine learning approach to kinematic synthesis of defect-free planar four-bar linkages, Journal of Computing and Information Science in Engineering 19 (2).

[46] H. Alt, Der übertragungswinkel und seine bedeutung für das konstruieren periodischer getriebe, Werkstattstechnik 26 (4) (1932) 61–64.

[47] L.-W. Tsai, Design of drag-link mechanisms with minimax transmission angle deviation, Journal of Mechanisms, Transmissions, and Automation in Design 105 (4) (1983) 686–690.

[48] P. J. Martin, K. Russell, R. S. Sodhi, On mechanism design optimization for motion generation, Mechanism and Machine Theory 42 (10) (2007) 1251–1263.

[49] M.-Y. Lee, A. Erdman, S. Faik, A generalized performance sensitivity synthesis methodology for four-bar mechanisms, Mechanism and Machine Theory 34 (7) (1999) 1127–1139.

[50] S. J. Lee, B. J. Gilmore, The determination of the probabilistic properties of velocities and accelerations in kinematic chains with uncertainty, Vol. 15th Design Automation Conference: Volume 2 —

Design Optimization of International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, 1989, pp. 225–232.

[51] M. Keennon, K. Klingebiel, H. Won, Development of the nano hummingbird: A tailless flapping wing micro air vehicle, in: 50th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, 2012, p. 588.

[52] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.

[53] M. Khorshidi, M. Soheilypour, M. Peyro, A. Atai, M. Shariat Panahi, Optimal design of four-bar mechanisms using a hybrid multi-objective GA with adaptive local search, Mechanism and Machine Theory 46 (10) (2011) 1453–1465.