Robust parameter synthesis for planar higher pair mechanical systems

Min-Ho Kyung and Elisha Sacks *

Media Division, Ajou University, Suwon, 443-749, South Korea
CS Department, Purdue University, West Lafayette, IN 47907, USA

Abstract

We present a parameter synthesis algorithm for planar, higher pair mechanical systems. The input is a parametric model of a mechanical system (part shapes and configurations) with nominal values and tolerance intervals for the parameters. The output is revised parameter ranges that guarantee correct kinematic function for all system variations. Nominal values are changed when possible and tolerance intervals are shrunk as a last resort. The algorithm consists of a three-step cycle that detects and eliminates system variations with incorrect kinematic function. The first step finds points in parameter space whose kinematic variation is maximal. The maximums of the higher pairs are derived by contact zone construction and are then combined into system maximums. The second step tests the points for correct kinematic function using configuration space matching and kinematic simulation. The third step adjusts the parameter ranges to exclude the points that fail the test. The cycle repeats until every point exhibits correct function. We demonstrate the algorithm on five real-world examples.

Key words: Parameter synthesis, higher pair, kinematics, configuration space.

1 Introduction

We present a parameter synthesis algorithm for planar, higher pair mechanical systems. Parameter synthesis is a central part of kinematic synthesis, which is the task of designing a mechanical system that performs a specified kinematic function. Kinematic synthesis is an iterative process in which designers

* Corresponding author, phone: 1-765-494-9026; fax: 1-765-494-0739
Email addresses: kyung@ajou.ac.kr (Min-Ho Kyung), eps@cs.purdue.edu (Elisha Sacks).

select a design concept, construct a parametric model, and assign parameter ranges in the form of nominal values plus tolerance intervals. The goal of parameter synthesis is to derive parameter ranges that guarantee correct kinematic function at a minimal cost. Overly tight tolerances can necessitate expensive manufacturing processes, whereas overly loose tolerances can lead to unreliable products.

The kinematic function of a system is the coupling between its part motions due to contacts between pairs of parts. A lower pair has a fixed coupling that can be modeled as a permanent contact between two surfaces. For example, a revolute pair is modeled as a cylinder that rotates in a cylindrical hole. A higher pair imposes multiple couplings due to contacts between pairs of part features. For example, gear teeth consist of involute patches whose contacts change as the gears rotate. The system transforms driving motions into outputs via sequences of part contacts.

Manufacturing variation causes the actual system to deviate from the nominal design, which causes incorrect kinematic function. One type of incorrect function is excessive deviation from the nominal part motion, such as a cam/follower that deviates by 5% from its prescribed path. A second type is a failure mode due to an unintended part contact, such as gears that jam because two pairs of teeth drive in opposite directions. System variation is modeled by generalizing the nominal design to a parametric family of designs. The part shapes and configurations are given in terms of symbolic parameters, such as lengths and angles, whose nominal values specify the nominal system. The allowable manufacturing variation is specified as tolerance intervals around the nominal parameter values. The parameter synthesis task is to assign parameter ranges that prevent incorrect kinematic function.

We formulate parameter synthesis as parameter space search [1]. The parameter space is an n-dimensional Euclidean space whose ith axis measures the ith design parameter. A point in parameter space represents a design instance. The parameter ranges define a hyper-box in parameter space that is centered at the nominal instance and that contains the allowable system variations. The synthesis task is to construct the largest possible hyper-box that is free of failure instances. We employ the *robust design* strategy [2,3] of first varying nominal values, which does not increase cost, and of tightening tolerances only as a last resort.

We present a parameter synthesis algorithm for planar, higher pair systems. We address these systems because they are common in applications, yet receive little attention in prior parameter synthesis research. The input to our algorithm is a parametric model with an initial parameter hyper-box. The algorithm searches the hyper-box for points that cause incorrect function. It excludes these points by revising the hyper-box. Following the robust paradigm,

the hyper-box is moved when possible and is shrunk as a last resort.

We cannot guarantee that all incorrect points are found because a complete search of the parameter hyper-box is prohibitively slow. Typical systems have tens to hundreds of parameters. Tiny steps are required because the kinematic function can vary suddenly or even discontinuously. Instead we search the portion of the parameter hyper-box where problems are most likely. We repeat the search/revision cycle until no incorrect points are found.

Incorrect function is most likely at points that maximize the variation of one or more contacts. Each contact generates a part of the kinematic function. A maximal variation in one contact represents a maximal deviation in its part of the kinematic function. A maximal variation in two or more contacts represents a maximal likelihood for unintended interactions among them. These heuristic arguments are the rationale for our parameter space search algorithm.

The paper is organized as follows. Section 2 reviews prior work. Section 3 specifies the input to our algorithm. Section 4 defines kinematic function for higher pair systems. Section 5 describe our algorithm and Section 6 validates it on five real-world examples. The validation shows that the algorithm finds and corrects incorrect kinematic function with a moderate number of iterations. The paper concludes with a discussion of our results.

2 Prior work

Algorithmic support for parameter synthesis is limited. Commercial packages, such as CATIA and IDEAS, support construction and visualization of parametric designs. Prior research provides synthesis algorithms for linkages [4,5] and cams [6,7], but does not address general higher pair systems. Most parameter synthesis research focuses on tolerance analysis algorithms [8]. The analysis falls into three increasingly general categories: small displacement analysis, large displacement analysis, and contact change analysis.

Small displacement analysis, also referred to as tolerance chain or stack-up analysis, is the most common. It consists of identifying a critical dimensional parameter (a gap, clearance, or play), building a tolerance chain based on part configurations and contacts, and determining the parameter variability range using vectors, tensors, or matrix transforms [9,10]. Recent research explores stack-up analysis with limited contact changes [11–13].

Large displacement analysis has been thoroughly studied in mechanical engineering [4]. It consists of defining kinematic relations between parts with a fixed contact topology, typically a linkage, and studying their kinematic

variation [14]. Commercial computer-aided tolerancing systems include this capability for planar and spatial mechanisms [15]. The kinematic variation is computed by linearization, which can be inaccurate, or by Monte Carlo simulation, which can be slow. Glancy and Chase [16] describe a hybrid algorithm that computes the first two derivatives of the system function with respect to the tolerance variables, calculates the first four moments of the system function, and fits an empirical variation distribution.

Large displacement analysis is inappropriate for higher pair systems with many contact changes, such as the examples in this paper. The user must enumerate the contact sequences, analyze them with the software, compose the results, and detect failures. We [17,18] developed the only general kinematic tolerance analysis algorithm for higher pair planar systems.

Tolerance synthesis Robust parameter synthesis is related to tolerance synthesis, which is the task of computing optimal tolerance intervals for a given nominal design. Prior work on tolerance synthesis in mechanical systems dates back to 1970, and proposes a variety optimization and modeling criteria [8]. Most work concentrates on modeling the cost-tolerance relations and the formulation of the optimization problem. The modeling limitations of these systems are the same as those for tolerance analysis. In addition, cost and manufacturing processes must be modeled. Dong and Gary [19] survey the issues involved in automating cost tolerance modeling. A general framework, the feasibility space approach, is proposed by Turner [20,21]. However, he assumes that the functional model is given, which is impractical for most mechanical systems. Several researchers develop functional model derivation and optimization algorithms for planar linkages [22,23], and some commercial CAT systems provide this capability [15]. We are unaware of prior work on general tolerance synthesis algorithms for higher pair systems.

Our prior work Parameter synthesis is the latest step in our ongoing research on algorithmic mechanical design with configuration spaces. Prior research addresses kinematic analysis [24], kinematic simulation [25], kinematic tolerance analysis [17], and kinematic synthesis [26]. That research provides several analysis tools, described below, that are employed by our parameter synthesis algorithm. However, neither our prior work nor any other published work gives a parameter synthesis algorithm for higher pair systems. The closest prior work is our kinematic synthesis algorithm, which interactively revises a nominal parametric design to eliminate kinematic failures. The designer suggests changes in kinematic function and the program achieves them by parameter space search. The current algorithm goes beyond that algorithm in two ways: it works without user assistance and it simultaneously revises the nominal design and the tolerances.

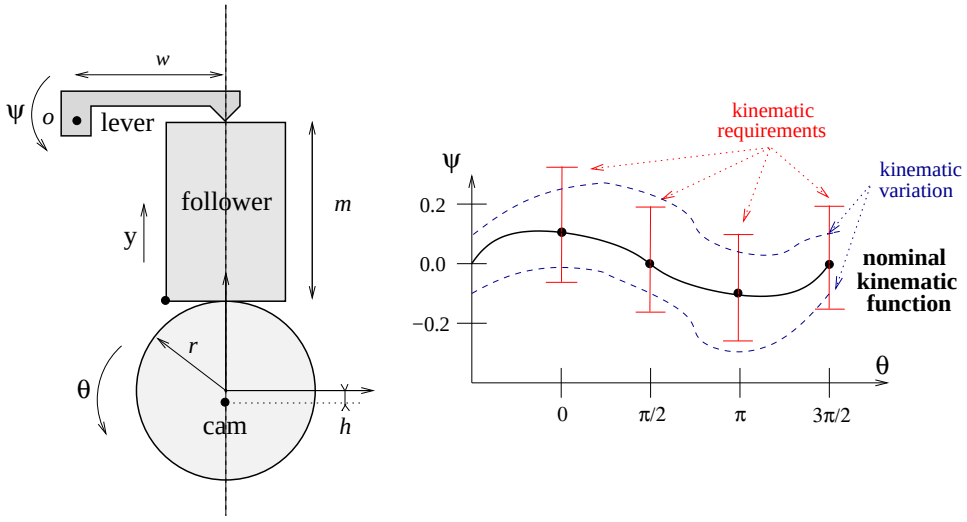


Fig. 1. Cam/follower/lever mechanism.

3 Input specification

The input to our algorithm is a parametric model of a planar mechanical system with initial tolerance ranges. The part shapes are simple loops of line and circle segments that are specified in part coordinate frames. A line segment, ab, is specified by the four shape parameters $a = (x_a, y_a)$ and $b = (x_b, y_b)$. A circle segment is specified by five shape parameters: its center $o = (x_o, y_o)$, radius, start angle, and end angle. A part has three configuration parameters, (x, y, θ), that represent its position and orientation in the system coordinate frame. Each part has one degree of freedom: translation along a coordinate axis or rotation around its reference point. The configuration parameter that corresponds to this motion is called the part motion variable. This class of inputs contains the vast majority of mechanisms [27].

Fig. 1 shows a sample input: a rotating cam that pushes a follower up and down that raises and lowers a rotating lever. The part coordinate frames are marked with dots. The cam is a circle whose center is $(0, h)$ and whose radius is r. The follower is a rectangle whose width is 0.75 cm and whose height is m. The lever is a polygon parameterized by its arm length w. The cam rotates by angle θ around $(0, 0)$. The follower translates vertically with y equal to the height of its reference point. The lever rotates by angle ψ around o. The system motion variables are $\mathbf{p} = (\theta, y, \psi)$. The input format appears below.

name: cam
origin: $(0, 0)$
motion: rotation
shape:
1. circle: (r, h) to $(-r, h)$
   center: $(0, h)$ radius: r
2. circle: $(-r, h)$ to (r, h)
   center: $(0, h)$ radius: r

name: follower
origin: $(0.375, 0)$
motion: vertical translation
shape:
1. line: $(0, 0)$ to $(0.75, 0)$
2. line: $(0.75, 0)$ to $(0.75, m)$
3. line: $(0.75, m)$ to $(0, m)$
4. line: $(0, m)$ to $(0, 0)$

The shape parameters and the non-motion configurations parameters comprise the design parameters, u_1-u_n, which form the parameter vector $\mathbf{u}$. The nominal value of u_i is $\overline{u}_i$. In our example, $\mathbf{u} = (r, h, m, w, x_l, y_l)$ and $\overline{\mathbf{u}} = (1.0, 0.1, 1.0, 1.0, -1, 2)$. Parameter u_i is constrained to the tolerance interval $[\overline{u}_i - s_i, \overline{u}_i + t_i]$ with $s_i, t_i \geq 0$. The elements s_i, t_i form the vectors $\mathbf{s}, \mathbf{t}$. A parameter vector $\mathbf{v}$ is feasible when $\overline{u}_i - s_i \leq v_i \leq \overline{u}_i + t_i$ for all i. The set of feasible vectors comprises the parameter hyper-box. Our task is to revise the initial hyper-box to eliminate points with incorrect kinematic function.

Our models are well suited to kinematic design research because they are simple, yet are flexible enough to model realistic design spaces for complex mechanical systems. Other types of models are common in engineering practice. We discuss how the two main ones relate to our models.

Variational models A richer class of parametric models, called variational models, is common in shape design. A variational model defines part shapes and configurations via geometric constraints. Example definitions are the point where two given lines intersect, the line through a given point and parallel to a given line, and the circle through two given points with a given radius. The designer specifies the constraints via a graphical interface and a solver constructs the corresponding segments. For example, we can define a line segment with endpoints $t = (1, 0)$ and $h = (x_h, y_h)$, using the variational constraints that points $a = (q_1, q_2)$ and $b = (q_3, q_4)$ lie on the segment. The values q_1–q_4 are specified by the designer. The solver formulates the equations

$$(a - t) \times (h - t) = 0$$
$$(b - t) \times (h - t) = 0$$

and solves for h.

Our parameter synthesis algorithm works with variational models. We employ parametric models in which the shape and configuration parameters are given as algebraic expressions in a separate set of design parameters. The only effect of this generalization is that we must use the implicit function theorem to

compute the derivatives of shape and configuration parameters with respect to design parameters, whereas the chain rule suffices for our original models. We do not employ this capability in this paper, so we refer the reader to our cited work for a full description. The algebraic input format also allows designers to specify couplings between segment tolerances.

GDT models GDT tolerance specifications are common in mechanical design. Part variation is specified by tolerance envelopes around nominal shapes. The semantics are only partially formalized [28]. It is straightforward, but tedious for a trained person to convert GDT specifications into our parametric ones. An automated translation tool would make our algorithm more accessible to the design community. Constructing such a tool is a research task in its own right.

4 Kinematic function

The next step is to define kinematic function for higher pair systems. We present the standard definition for fixed contact systems, generalize it to higher pairs, and characterize correct and incorrect kinematic function. The material in this section is a review of prior work that underlies the parameter synthesis algorithm. We explain and illustrate the main concepts; the technical details appear in our cited journal articles.

4.1 Fixed contact systems

The kinematic function of a fixed contact system is a functional relationship, $C(\mathbf{p}) = 0$ among its motion variables. The system configuration at any point in the work cycle is determined by the values of these variables. The kinematic function of a parametric system, $C(\mathbf{p}, \mathbf{u}) = 0$, is a functional relationship among its motion variables and its design parameters. The nominal function is $C(\mathbf{p}, \overline{\mathbf{u}}) = 0$. In our example (Fig. 1), the nominal kinematic function projects to the bold curve in the graph on the right.

A kinematic variation is the kinematic function, $C(\mathbf{p}, \mathbf{v}) = 0$, generated by a point $\mathbf{v}$ in the parameter hyper-box. The set of kinematic variations forms a band around the nominal kinematic function. A kinematic variation is incorrect when its distance from the nominal function exceeds a threshold. The error metric can specify the Euclidean distance between the curves at sample points and can also bound the distance between their derivatives.

In our example, the dashed lines bound the maximum kinematic variation for the tolerances intervals $s_i, t_i = 0.02$ mm. The error metric is that the lower/upper variation in ψ is at most 0.15/0.2 radians at the system configurations $(0, 2.1, 0.1)$, $(\pi/2, 2.0, 0.0)$, $(\pi, 1.9, -0.1)$, and $(3\pi/2, 2.0, 0.0)$. These bounds appear as vertical intervals around the nominal ψ values. The tolerance ranges allow incorrect kinematic function: the upper variation is too large at $\pi/2$ and the lower variation is too large at π.

4.2 Higher pairs

The kinematic function of a higher pair, a/b, consists of multiple functions in their motion variables, m_a and m_b. A contact between part features f_a and f_b generates a function in the manner described above, but this function only applies in the portion of (m_a, m_b) space where f_a and f_b are in contact. The a/b kinematic function consists of the sequence of these functions that occur throughout the work cycle.

We represent higher pair kinematic function with configuration spaces [24]. The configuration space of a pair is a two-dimensional manifold whose coordinates are the two part motion variables. Points in configuration space correspond to configurations of the pair. The configuration space partitions into blocked space where the parts overlap, free space where they are separate, and contact space where they touch. Free and blocked space are open sets whose common boundary is contact space. Contact space is a closed set comprised of subsets that represent contacts between part features.

We illustrate these concepts on a Geneva pair comprised of a driver and a wheel (Fig. 2). The Geneva pair is a canonical example that we employ in several prior publications. We reuse it here because it shows the key concepts of our research in a simple setting. The driver consists of a driving pin and a locking arc segment mounted on a cylindrical base. The wheel consists of four locking arc segments and four slots. The wheel rotates around axis A and the driver rotates around axis B. Each driver rotation causes an intermittent wheel motion with four drive periods where the driver pin engages the wheel slots and with four dwell periods where the driver locking arc engages the wheel locking arcs.

The configuration space coordinates are the part orientations θ and ω in radians. The pair is displayed in configuration $(0, 0)$, which is marked with a dot. Blocked space is the gray region, contact space is the black curves, and free space is the channel between the curves. Free space forms a single channel that wraps around the horizontal and vertical boundaries, since the configuration at $\pm\pi$ coincide. The defining equations of the channel boundary curves express

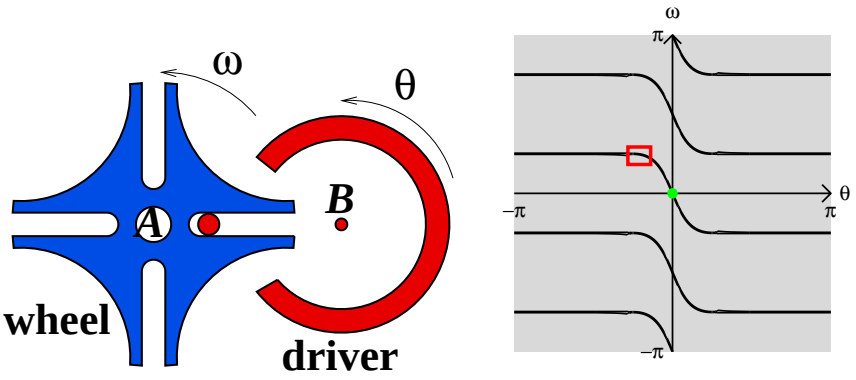


Fig. 2. Geneva pair and its configuration space.

the coupling between the part orientations. The horizontal segments represent contacts between the locking arcs, which hold the wheel stationary. The diagonal segments represent contacts between the pin and the slots, which rotate the wheel. The contact sequences of the pair are the configuration space paths in free and contact space. In a typical sequence, the driver rotates clockwise (decreasing θ) and alternately drives the wheel counterclockwise with the pin (increasing ω) and locks it with the arcs (contact ω).

4.3 Kinematic variation

We [17] model kinematic variation by generalizing configuration space to parametric parts with tolerances. Kinematic variation occurs in contact space. Each feasible parameter space point generates its own contact space. The union of these spaces over the parameter hyper-box defines a band around the nominal contact space, called a contact zone, that bounds the worst-case kinematic variation of the pair. The contact zone is the subset of the configuration space where contacts can occur for some feasible point.

Fig. 8a shows the contact zone that our algorithm generates for a parametric model of the Geneva pair. The zone is a detail of the portion of the configuration space in the box in Fig. 2. This portion is the interface between a horizontal and a diagonal channel where the driver pin leaves a wheel slot and the locking arcs engage. The two dark gray bands that surround the channel boundary curves are the contact zone. The white region between the bands is the subset of the nominal free space that is free for all feasible points.

4.4 Incorrect kinematic function

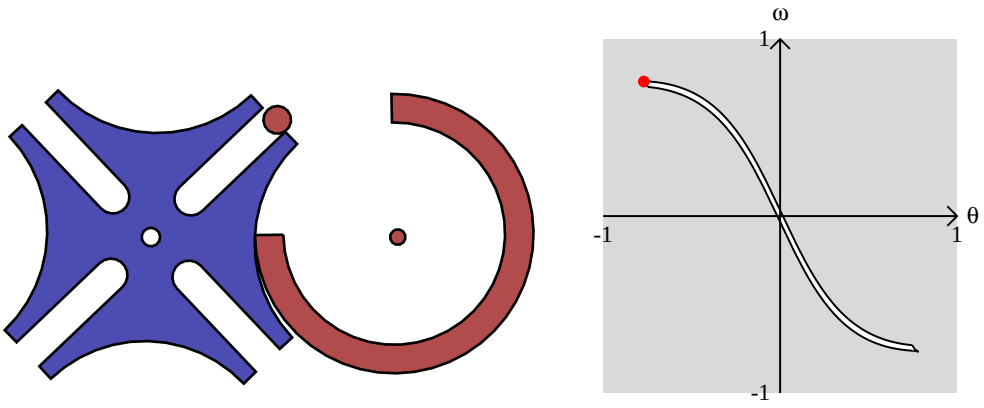
Our algorithm detects and corrects two types of incorrect kinematic function due to parameter variation. The first type, excessive deviation from the nominal part motion, occurs in all types of mechanical systems. The second type, failure modes due to unintended part contacts, is unique to higher pair systems. In both cases, the nominal kinematic function is compared to that of a candidate feasible point.

Motion deviation The motion deviation test compares the nominal and candidate motion paths of the driven parts using kinematic simulation [25]. The simulator steps the driving part through its work cycle and propagates the motion through the system. The driving motion and the step are specified as input. The output is a nominal and a candidate configuration for each driven part at each driver configuration. The motion variation of a candidate configuration is the minimal distance to a nominal configuration. The motion variation of the candidate is the maximum over these minimal distances. A candidate passes the motion test when the variation is below a specified limit. Section 6.5 presents an example system that initially fails this test.

Failure modes The Geneva contact zone (Fig. 8a) reveals a possible failure mode. The lower and upper bands overlap near where the horizontal and diagonal channels meet. Some feasible point might yield a configuration space in which the lower and upper contact curves intersect, block the channel, and cause the mechanism to jam. Fig. 3 illustrates this failure mode. But contact zone overlap does not guarantee a faulty feasible point. The contact zone is a conservative estimate of kinematic variation that ignores dependencies among contacts due to shared parameters. Hence, there might be no point that moves both curves into the overlap region.

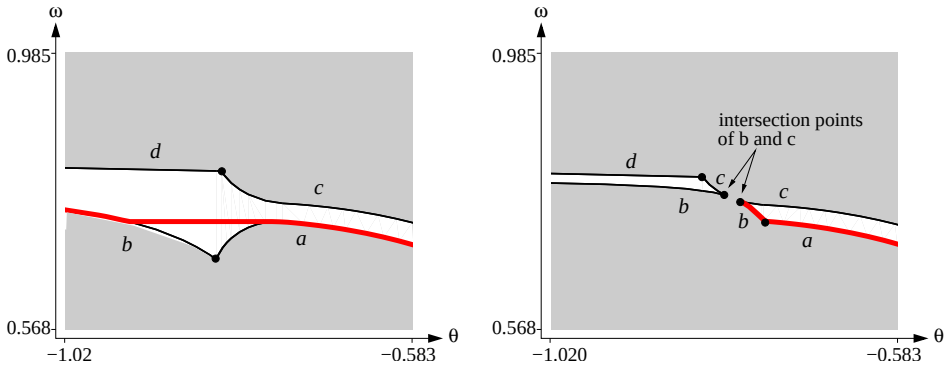
The failure mode test matches the nominal and candidate contact spaces. The pairs match when the two spaces have the same structure: they have the same number of components and each component in the first space matches a unique component in the second space. Two components match when they consist of equivalent curves in the same cyclic order. Two curves are equivalent when they are generated by the same pair of part features. A failed match indicates a difference between the nominal and the candidate kinematic functions, which we interpret conservatively as a failure mode. The matching algorithm is described elsewhere [26].

Fig. 4 illustrates matching on the nominal (Fig. 2) and jammed (Fig. 3) Geneva configuration spaces. The match fails because the nominal contact space has



(a) (b)

Fig. 3. Geneva failure: (a) jamming configuration; (b) configuration space.



(a) (b)

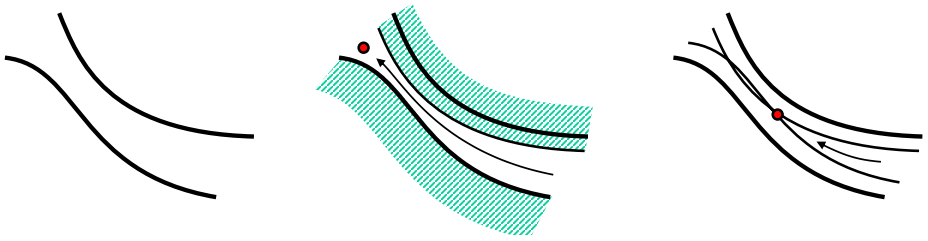
Fig. 4. Mismatch between (a) nominal and (b) jammed Geneva spaces.

two components, whereas the jammed space has one. The structural mismatch is that curves b and c are disjoint in the nominal space, but intersect in the jammed space.

5 Parameter synthesis algorithm

The parameter synthesis algorithm is summarized below. The algorithm consists of three main steps: candidate selection, candidate testing, and parameter range revision to exclude the candidates that fail the test. The cycle repeats until every candidate passes the test. The user can also specify a maximum number of iterations. The input and the two tests have already been described. This section describes candidate selection and parameter revision.

Input: parametric system, initial parameter ranges.
1. Select candidate feasible points.



(a) (b) (c)

Fig. 5. Candidate computation: (a) nominal kinematics; (b) one maximal variation; (c) two maximal variations.

2. Test candidates for motion deviation and for failure modes.
3. If all candidates pass, return parameter ranges.
4. Revise parameter ranges.
5. Go to step 1.

Output: revised parameter ranges.

5.1 Candidate selection

The candidates are selected in two steps. Step 1 finds feasible points whose contact spaces intersect the boundaries of the system contact zones. Each point maximizes the kinematic variation in a single feature/feature contact of a single higher pair. Step 2 forms the candidates from unions of step 1 points. The candidates maximize the simultaneous kinematic variation of multiple contacts, which is where incorrect function is likeliest to occur and is hardest to detect.

Fig. 5 illustrates the two steps. Part a shows the configuration space of a nominal pair with a free space channel. Part b shows the configuration space for a step 1 point that maximizes the variation of the upper channel boundary; the thin black curve is the new upper boundary and the lower boundary is unchanged. The channel remains open, so the kinematic function is qualitatively correct. Part c shows the configuration space for a step 2 point that maximizes the variation of both channel boundaries. The boundaries overlap, which indicates incorrect kinematic function. A similar failure occurs in Fig. 3 and is detected in Fig. 4.

5.1.1 Step 1

Each higher pair is processed separately. The contact space of a pair consists of contact curves that represent contact between feature pairs. Each curve is processed separately. The computation is described in our kinematic tolerance

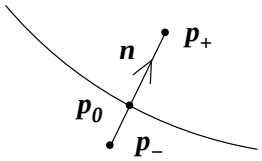


Fig. 6. Contact zone computation: nominal point $\mathbf{p}_0$ generates contact zone boundary points $\mathbf{p}_-$ and $\mathbf{p}_+$.

analysis paper [17]. We summarize it here for completeness.

A curve has a parametric equation $C(\mathbf{p}, \mathbf{u}) = 0$ where $\mathbf{p}$ denotes the configuration space coordinates, for example $\mathbf{p} = (\theta, \omega)$ in the Geneva pair. There is one type of equation for every combination of features and motions, such as rotating circle/translating line. For example, the driver/wheel locking arc equation is $(B + R_\theta m - A - R_\omega n)^2 = (r - s)^2$ where B, A are the centers of rotation, m, n are the arc centers in part coordinates, R_θ, R_ω are rotation operators, and r, s are the arc radii. The equation states that the distance between the arc centers equals the difference of their radii.

The nominal curve, $C(\mathbf{p}, \overline{\mathbf{u}})$, is discretized to a given accuracy (10^{-5} in the paper). A maximal $\mathbf{u}$ is computed for each discretization point, $\mathbf{p}_0$, as follows. The kinematic variation occurs along the normal vector, $\mathbf{n}$, to the contact space (Fig. 6). It has the form $\mathbf{p}_0 + k\mathbf{n}$ with k a function of $\mathbf{u}$. The desired $\mathbf{u}$ values maximize/minimize k to yield points $\mathbf{p}_+$/ $\mathbf{p}_-$ on the upper/lower boundaries of the contact zone. They are computed by solving a nonstandard optimization problem with a custom algorithm.

If the parameter u_i does not appear in the contact equation $C(\mathbf{p}, \mathbf{u}) = 0$, its value does not effect the output $\mathbf{u}$ values. We say that u_i is free for the contact. For each output point, step 1 reports the free parameters and sets them to their nominal values.

5.1.2 Step 2

Step 2 of the candidate selection algorithm appears below. The input, s, is the step 1 output; the output, t, is the candidate feasible points. The output consists of all unions of compatible points of s. Two points, $\mathbf{u}$ and $\mathbf{v}$, are compatible when for every i either u_i is free, v_i is free, or $u_i = v_i$. Their union, $\mathbf{w} = \mathbf{u} \cup \mathbf{v}$, is defined as follows: w_i is free when u_i and v_i are free, $w_i = u_i$ when v_i is free, and $w_i = v_i$ otherwise.

Input: list, s, of step 1 points.
1. $t \leftarrow s$
2. $f \leftarrow 0$
3. for each s_i in s

4. for each t_j in t
5. if s_i and t_j are compatible
6. $d \leftarrow s_i \cup t_j$
7. if d is not in t
8. $f \leftarrow 1$
9. add d to t
10. if $f = 1$ go to Step 2
11. return t

Output: list, t, of candidate feasible points.

Line 1 of the algorithm initializes t to s. The first iteration of lines 3–9 adds to t every compatible pair of s points, $s_i \cup t_j = s_i \cup s_j$. The second iteration adds every compatible triple of s points, since $\cup$ is associative and commutative. The kth iteration adds every compatible k tuple. The iteration ends when no new compatible tuples can be formed.

We illustrate the algorithm on a simple, artificial input

$$s = \{s_1, s_2, s_3, s_4\} = \{(1, 2^*, 3^*), (5^*, 4, 3^*), (7, 2, 3), (5^*, 4, 8)\}$$

where x^* denotes that x is free. The compatible pairs are $s_{12} = s_1 \cup s_2 = (1, 4, 3^*)$, $s_{14} = s_1 \cup s_4 = (1, 4, 8)$, and $s_{24} = s_2 \cup s_4 = (5^*, 4, 8)$. The only compatible triple is $s_{124} = s_{12} \cup s_4 = (1, 4, 8)$, which is already in t, so the iteration ends.

We accelerate the algorithm in two ways. We allow t points to contain at most two s points per higher pair. Using three is pointless: the associated configuration space is two-dimensional, so three contact curves cannot intersect (hence cannot interact), except for rare degenerate cases. At step 7, we do not add d to t if it is within ϵ of any member of t in the max norm. In our examples, setting ϵ to 1% of the initial tolerance interval eliminates 75% of the candidates, yet never misses one with incorrect kinematic function.

5.2 Parameter range revision

The parameter synthesis algorithm revises the parameter ranges to eliminate the incorrect kinematic function revealed by the candidates. The revision consists of two steps. Step 1 computes points that need to be excluded from the parameter hyper-box. Step 2 excludes them by modifying $\overline{\mathbf{u}}$ when possible and by shrinking tolerance intervals otherwise.

5.2.1 Step 1

Each failed candidate is processed separately to yield a *critical* point that needs to be excluded from the parameter hyper-box. First, consider a system with one parameter, u, and with failed candidate v. The same failure occurs in a neighborhood of v because the system depends continuously u. Continuity implies the existence of a critical parameter value, c, where the failure first occurs: the kinematic function is correct on $[\overline{u}, c)$ and is incorrect on $(c, v]$. Parameter revision needs to exclude c to eliminate the failure; excluding v alone leaves the failure interval in the parameter range. In higher dimensions, $\mathbf{c}$ is the first failure point on the line segment $[\overline{\mathbf{u}}, \mathbf{v}]$, as shown in Fig. 7a–b.

The critical point is found by bisection search, as shown below. The initial interval is $[\overline{\mathbf{u}}, \mathbf{v}]$. At each iteration, the midpoint parameter point is tested. If the kinematic function is correct the lower limit is replaced; otherwise the upper limit is replaced. The iteration ends when the interval width reaches ϵ (10^{-5} in the paper).

Input: $\overline{\mathbf{u}}, \mathbf{v}, \epsilon$.
1. $s \leftarrow 0, t \leftarrow 1$
2. $m \leftarrow 0.5(s+t), \mathbf{w} \leftarrow \overline{\mathbf{u}} + m(\mathbf{v} - \overline{\mathbf{u}})$
3. if $t - s < \epsilon$ return $\mathbf{w}$
4. if $\mathbf{w}$ is correct $s \leftarrow m$ else $t \leftarrow m$
5. go to Step 2

Output: critical point.

5.2.2 Step 2

Step 2 of the parameter revision algorithm modifies the parameter hyper-box to exclude the critical points. The hyper-box is centered at $\overline{\mathbf{u}}$ and its width in the ith dimension is the tolerance interval of $\mathbf{u}_i$. We aim to move the hyper-box minimally without shrinking the intervals, which means changing $\overline{\mathbf{u}}$ minimally without decreasing any s_i or t_i. If this proves impossible, we aim for an optimal combination of motion and shrinking. Finding the optimal revision is computationally intractable, so we employ a greedy algorithm to obtain a good revision quickly.

The optimal revision for a single critical point, $\mathbf{v}$, is easy to compute. Find the perpendicular distance, d, from $\mathbf{v}$ to a hyper-box face along its inward normal, $\mathbf{n}$. The normal is a unit vector whose ith element is ± 1 and whose other elements are 0. Incrementing $\overline{\mathbf{u}}$ by $d\mathbf{n}$ excludes $\mathbf{v}$ from the hyper-box (Fig. 7c). The face that minimizes d yields the optimal revision. The second closest face yields the next best revision and so on.

We exclude multiple critical points by combining their optimal revisions. Two

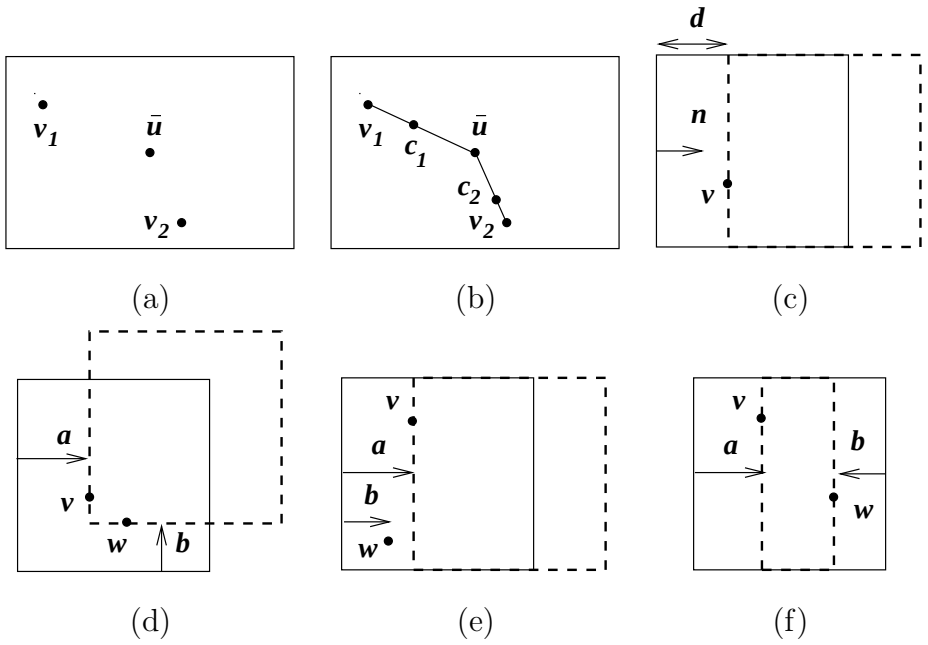


Fig. 7. Parameter revision: (a) failed candidates; (b) critical points; (c) optimal revision for one point; (d-e) revision for compatible points; (f) revision for incompatible points. Solid/dashed boxes are initial/revised parameter hyper-boxes.

revisions, $\mathbf{a}$ and $\mathbf{b}$, can be combined when they are compatible: $a_i b_i \geq 0$ for all i. Their combination, $\mathbf{c} = \mathbf{a} \cup \mathbf{b}$, is defined as $c_i = \max(a_i, b_i)$ when $a_i, b_i \geq 0$ and as $c_i = \min(a_i, b_i)$ otherwise. Fig. 7d–e show two ways that compatible points combine. Fig. 7f shows how an incompatible pair is handled by shrinking the parameter hyper-box, as explained below. The following theorems establish the relevant properties of combination.

Theorem 5.1 *If* $\mathbf{a}$ *excludes* $\mathbf{v}$ *and* $\mathbf{a}$ *is compatible with* $\mathbf{b}$*,* $\mathbf{a} \cup \mathbf{b}$ *excludes* $\mathbf{v}$*.*

Proof. There exists some $\mathbf{v}_i$ that $\mathbf{a}$ moves out of its tolerance interval, since otherwise $\mathbf{a}$ would not exclude $\mathbf{v}$. Suppose $\mathbf{a}$ moves it out the right side, so $v_i > \overline{u}_i + a_i + t_i$. Let $\mathbf{c} = \mathbf{a} \cup \mathbf{b}$. We have $c_i = \max(a_i, b_i) \geq a_i$, which implies $v_i > \overline{u}_i + c_i + t_i$, so $\mathbf{c}$ moves v_i out the right side. The proof is similar when $\mathbf{a}$ moves $\mathbf{v}_i$ out the left side of its tolerance interval. □

Theorem 5.2 *If* $\mathbf{a}$ *excludes* $\mathbf{v}$*,* $\mathbf{b}$ *excludes* $\mathbf{w}$*, and* $\mathbf{a}$ *is compatible with* $\mathbf{b}$*, then* $\mathbf{a} \cup \mathbf{b}$ *excludes* $\mathbf{v}$ *and* $\mathbf{w}$*.*

Proof. Follows from Theorem 5.1 and the symmetry of compatibility and of the $\cup$ operator. □

The parameter revision algorithm is shown below. The revision, $\mathbf{a}$, is initialized to the zero vector. Each critical point, $\mathbf{w}$, is processed in turn. If possible, $\mathbf{a}$ is updated to $\mathbf{a} \cup \mathbf{b}$ with $\mathbf{b}$ the smallest compatible $\mathbf{w}$ revision. Otherwise, a tolerance interval is shrunk to exclude $\mathbf{w}$. We select the tolerance interval

$[\overline{u}_i - s_i, \overline{u}_i + t_i]$ whose left or right endpoint minimizes the distance to w_i and replace it with $[w_i, \overline{u}_i + t_i]$ or with $[\overline{u}_i - s_i, w_i]$ accordingly (Fig. 7f). We set $\overline{u}_i$ to the midpoint of the revised interval.

Input: $\overline{\mathbf{u}}, \mathbf{s}, \mathbf{t}$, critical points.
1. initialize **a** to zero vector
2. for each critical point **w**
3. if **w** in the parameter hyper-box
4. sort the **w** revisions in increasing d order
5. for each revision **b**
6. if **a** is compatible with **b**
7. $\mathbf{a} \leftarrow \mathbf{a} \cup \mathbf{b}$
8. go to Step 2
9. shrink parameter hyper-box to exclude **w**

Output: revised parameter hyper-box.

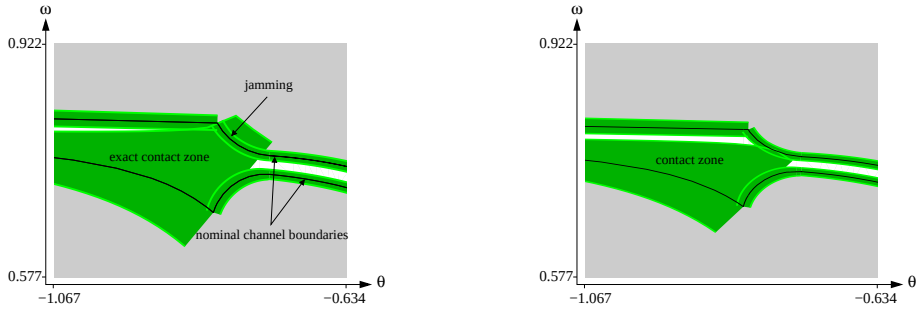
The algorithm output depends on the order in which the critical points are processed. We employ the heuristic of processing them in decreasing d order where d is the distance to the closest face of the initial parameter hyper-box. In our test cases, the algorithm never generates a suboptimal revision.

6 Results

We demonstrate the parameter synthesis algorithm on two common higher pairs from the engineering literature, on one custom pair, and on two systems comprised of three custom pairs apiece. In prior work, we have used these five examples to demonstrate kinematic analysis (nominal and toleranced) with configuration spaces. Here, we move beyond analysis to synthesis. Our synthesis algorithm detects and corrects intermittent failure modes in all five examples. It never shrinks a tolerance interval, so the revised nominal values are in the initial tolerance intervals. Yet these small changes eliminate hundreds of incorrect kinematic functions to produce robust designs. We use symmetric initial tolerance intervals for simplicity; the algorithm handles general intervals.

6.1 Geneva pair

The first example is a Geneva pair (Fig. 2) from an encyclopedia of mechanisms [29]. The pair has 26 design parameters. We assigned initial tolerances of ± 0.02 mm to the length parameters (line segment coordinates, circle center coordinates, circle radii) and $\pm 1^\circ$ to the angle parameters (circle segment start



(a) (b)

Fig. 8. Detail of Geneva pair contact zone: (a) initial; (b) final.

and end angles). The contact zone shows possible jamming (Fig. 8a).

The first iteration of the parameter synthesis algorithm is as follows. Step 1 of candidate selection (Sec. 5.1.1) finds 44 maximal parameter sets and step 2 (Sec. 5.1.2) generates 137 candidates. Candidate testing finds that 21 of these points have incorrect kinematic function; this is the jamming mode shown in Fig. 8a. Parameter revision finds the corresponding 21 critical points (Sec. 5.2.1) and excludes them by moving the parameter hyper-box (Sec. 5.2.2).

There are three more iterations in which five, two, and one incorrect points are found. The algorithm terminates after the fifth iteration because every candidate has correct kinematic function. The maximum change in any nominal parameter value is 8% of its tolerance interval. The contact zone for the output parameter hyper-box (Fig. 8b) is overly conservative. The lower and upper channel boundaries overlap slightly, but there is no feasible point that realizes both boundary variations.

The 11 parameters that change are described below. We work in a coordinate system whose origin is the driver center of rotation and whose x axis lies on the line between the driver and wheel centers. Parameter a is the x coordinate of the driver pin. Parameter b is the x coordinate of the driver locking arc center, c is its start and end angle, and d is its radius. Parameter e is the x coordinate of the wheel center. Parameters f and g are the slot width and depth. Parameter h is the vertical distance between the wheel center and the slot bottom. Parameters i and j are the x and y coordinates of the center of one locking arc and k is the radius. The other three arcs have centers $(i, -j)$, $(-i, j)$, and $(-i, -j)$.

name	initial value	final value	name	initial value	final value
a	5.451074	5.433074	g	4.235842	4.253842
b	-0.142998	-0.160998	h	1.235858	1.253858
c	0.798530	0.780530	i	5.471301	5.489301
d	4.505872	4.523872	j	5.471301	5.471154
e	8.032096	8.014096	k	4.401504	4.383504
f	1.052529	1.034529			

6.2 Reciprocating cam/follower pair

The second example is a reciprocating cam/follower pair (Fig. 9a) from an encyclopedia of mechanisms [29]. The cam rotates around an axis on the frame and the follower translates horizontally. When the cam rotates clockwise, its three fingers alternately push the upper and lower slanted segment of the follower right and left. Fig. 9b shows the configuration space: θ is the cam angle and y is the follower displacement. The red curve marks the motion path. The pair has 11 design parameters with initial tolerances of ± 0.02 mm.

The initial contact zone shows overlap (Fig. 10a). The parameter synthesis algorithm eliminates this overlap in 9 iterations (Fig. 10b). In each iteration, candidate selection finds 45–57 points. Candidate testing finds 15 incorrect points in the first iteration and 5 in the second. The maximum parameter change is 40% of the initial interval. The parameters that change are the cam finger length and width, and the vertex coordinates of the two triangular bulges on the inner follower profile.

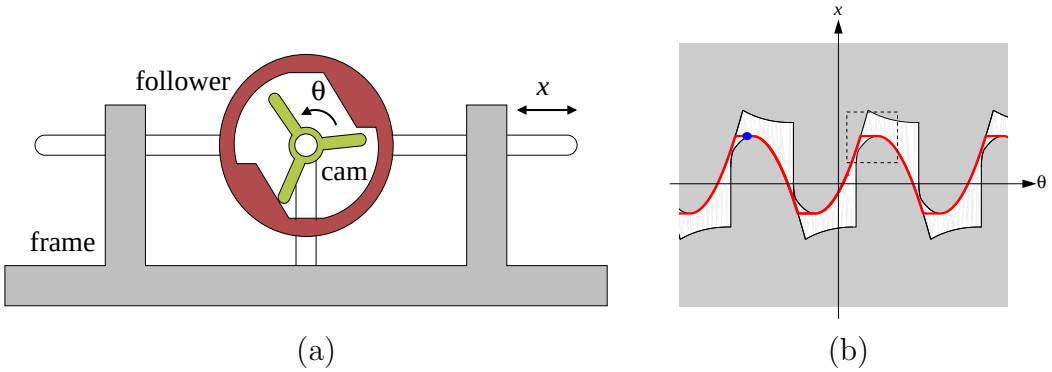
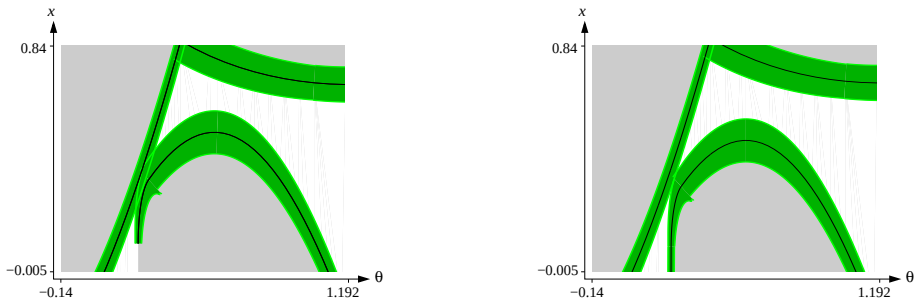


Fig. 9. (a) Reciprocating cam pair; (b) configuration space.



(a) (b)

Fig. 10. Detail of reciprocating cam pair contact zone: (a) initial; (b) final.

6.3 Optical filter pair

The third example is a custom cam/follower pair in an optical filter mechanism developed by Israel Aircraft Industries (Fig. 11a). The parts are attached to a fixed frame with pin joints. Rotating the cam counterclockwise causes its pin to engage the follower slot and drive the follower clockwise. The follower motion ends when the cam pin leaves the slot, at which point the follower filter covers the lens. As the cam continues to rotate, its locking arc aligns with the complementary follower arc and locks the follower in place. Rotating the cam clockwise returns the filter to the initial state. Fig. 11b shows the configuration space: θ is the cam angle, ω is the follower angle, and the free space structure is similar to that of the Geneva pair. The pair is parametrized by 17 parameters with tolerances of ± 0.02 mm. We derived the model and the initial tolerances from proprietary engineering drawings.

The initial contact zone shows overlap between the channel boundary curves (Fig. 12a). The synthesis algorithm eliminates this overlap in 19 iterations (Fig. 12b). In each iteration, candidate selection finds 75 maximal parameter sets that generate 700 points. Candidate testing finds 146 incorrect points in the first iteration, and this number decreases linearly to zero over the 19 iterations. The maximum parameter change is 0.0271, which moves one nominal value just outside its initial interval. Although the relative change is large, the kinematic variation is slight.

6.4 Camera shutter mechanism

The fourth example is a camera shutter mechanism comprised of a driver, a shutter, and a shutter lock (Fig. 13). The user advances the film (not shown), which engages the driver film wheel and rotates the driver counterclockwise. The shutter tip follows the driver cam profile, which rotates the shutter clock-

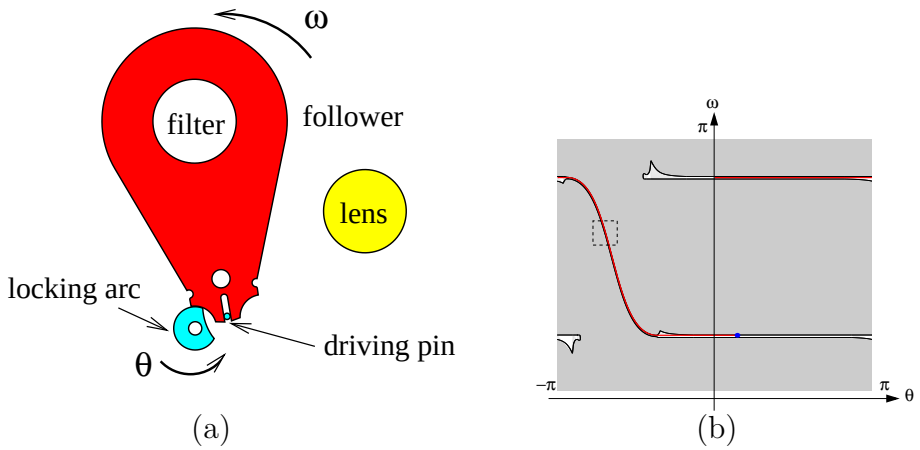


Fig. 11. (a) Cam/follower pair; (b) configuration space.

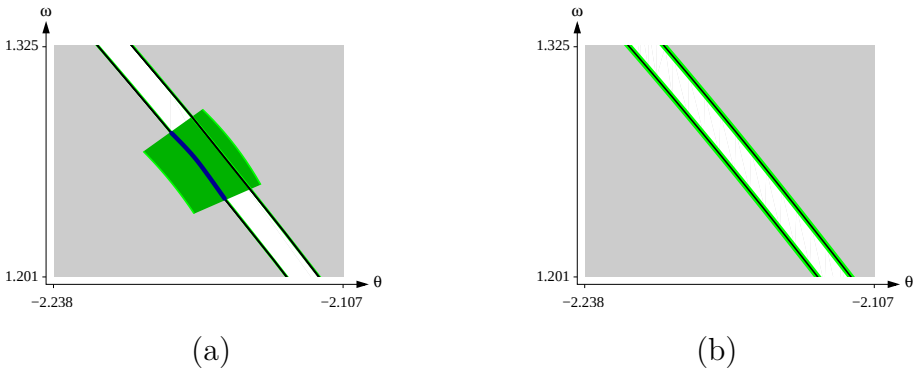


Fig. 12. Detail of cam/follower contact zone: (a) initial; (b) final.

wise, which extracts the shutter pin from the shutter lock slot. When the pin leaves the slot, a torsional spring rotates the shutter lock clockwise until its tip engages the driver slotted wheel. The mechanism is parametrized by 22 parameters with tolerances of ±0.09 mm. We constructed the model by reverse engineering a camera. The initial tolerances are an estimate of the variation of the molding process.

In the first synthesis iteration, candidate selection generates 1673 points. All the points pass the failure mode test, but 40 exhibit excessive motion variation. The shutter angle reaches a minimum value (furthest clockwise) that is too small for its pin to clear the shutter lock. The problem is fixed in seven iterations. The maximum change is in the y coordinate of the cam axis, which is moved 0.1577mm.

Fig. 14 shows the nominal and incorrect motion paths in the configuration spaces of the shutter/lock and driver/lock kinematic pairs. The nominal shutter/lock motion (the thick line) follows the contact space from right to left (1) as the shutter rotates clockwise, leaves the free-space mouth and drops vertically (2) when the shutter pin clears the shutter lock slot, and moves right and down (3) when the torsional spring rotates the shutter lock clock-

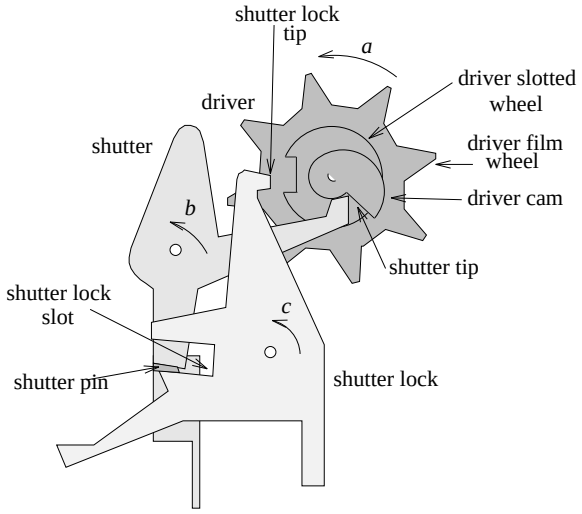


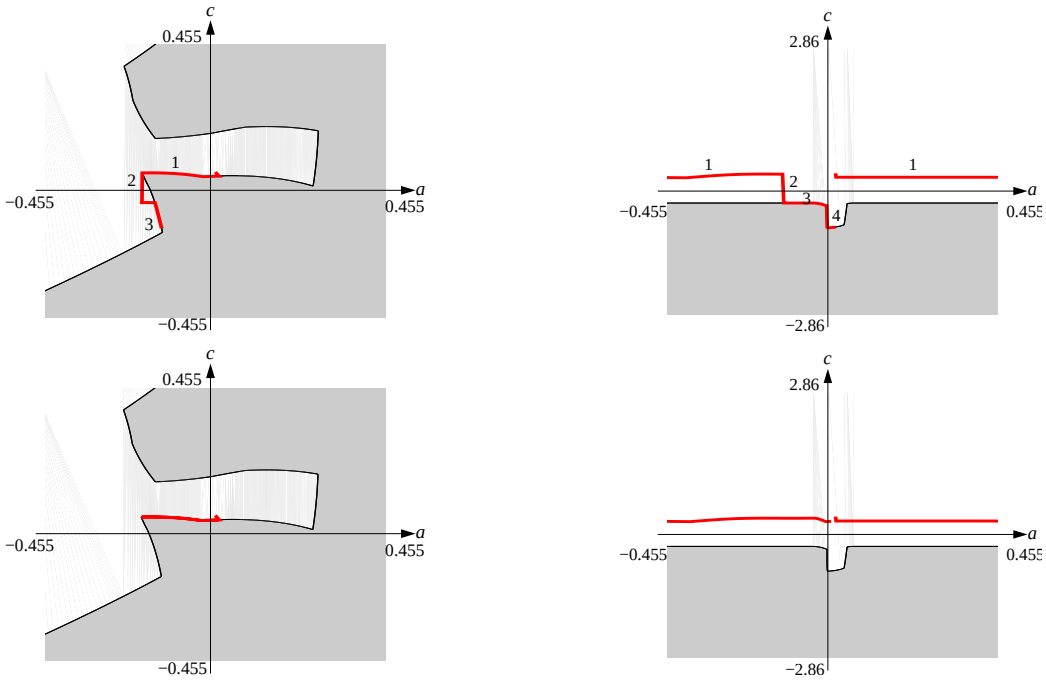
Fig. 13. Camera shutter mechanism.

wise. The incorrect motion follows the contact space from right to left, but does not reach the left side of the mouth, hence cannot drop vertically. The correct driver/lock motion path starts with a horizontal segment in free space (1) where the driver rotates counterclockwise and the lock is held away from the driver by the shutter. It continues with a vertical segment (2) where the shutter lock tip rotates into contact with the driver, a horizontal segment (3) where the driver rotates and the tip follows its circular profile, and a second vertical segment (4) where the tip engages the driver slot. The incorrect motion path consists solely of the first horizontal segment, since the shutter prevents the lock from rotating.

6.5 Gear selector mechanism

The fifth example is a gear selector mechanism that is part of an automatic transmission with tens of moving parts (Figure 15). The mechanism consists of a cam, a piston, a pin, and a valve body. The valve body is fixed to the frame. The pin is attached to the valve body by a flexible rod that acts as a torsional spring. The piston length is 108.9 cm, the tips of the triangular cavities in the cam bottom are 55.1–55.6 cm from its center, and the distance between the pin and its center of rotation is 92.3 cm. When the driver moves the gearshift (not shown), the cam rotates, which slides the piston along the valve body via a pin that engages a slot in its left side. When the driver releases the gearshift, the torsional spring rotates the pin clockwise until it engages between two adjacent cam teeth. There is one engagement position for each gear setting (1, 2, 3, D, N, R, P).

We obtained the nominal part geometry from Ford Werke AG, Cologne. We constructed a parametric model by adding variation parameters to the func-



shutter/lock driver/lock

Fig. 14. Nominal (top) and incorrect (bottom) motion paths.

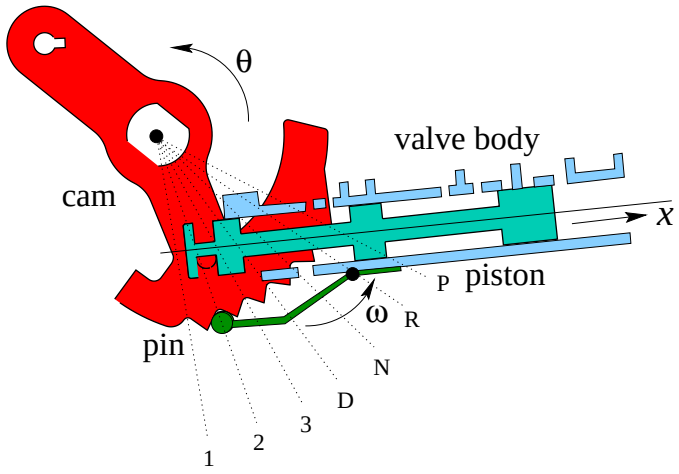
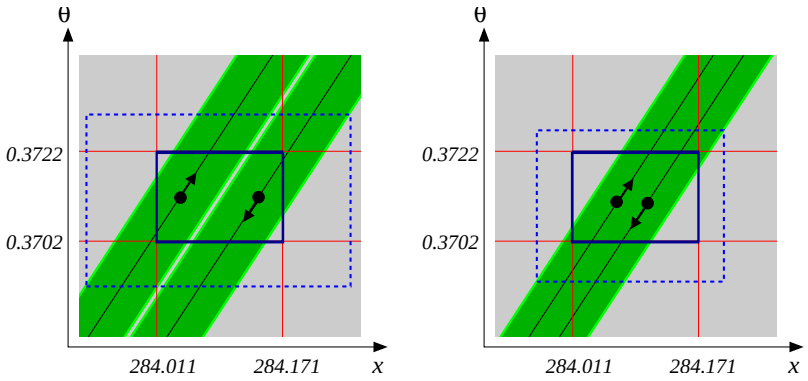


Fig. 15. Gear selector mechanism.

tional features of the parts. For the cam, we toleranced the line segments that form the tooth sides and the circular pin that engages the piston. For the piston, we toleranced the two vertical segments that are in contact with the cam pin. For the pin, we toleranced the single, circular feature. To account for uncertainties in the position of the rotation axes, we also toleranced the centers of rotation of the cam and the pin. We chose the piston as the reference part, so there was no need to tolerance its translation axis. There are a total of 34 parameters with tolerances of ± 0.02 cm.



(a) (b)

Fig. 16. Piston/cam contact zone: (a) initial; (b) final.

The tolerancing problem is to ensure that the piston closes the correct valves in gear setting 1. The other settings are analogous. Correct function is obtained by bounding the motion variation of the piston and the cam. Fig. 16a shows the initial piston/cam contact zone space. The nominal contact space is a diagonal channel bounded by two lines labeled with arrows that mark typical contact configurations. The motion path follows the left/right channel boundary when the cam rotates counterclockwise/clockwise. The allowable motion variation is 0.08 cm translation for the piston and 0.001 radians rotation for the cam (the solid box in the contact zone). The dashed box is a conservative bound on the motion variation that we obtain by composing the piston/cam and pin/cam contact zones [17]. The fact that the dashed box is larger than the solid box suggests possible motion variation failures.

The first iteration of the synthesis algorithm finds 331 motion variation failures out of 715 candidates. The number drops to 17 after 10 iterations. Another 149 iterations are required to obtain correct tolerances. Fig. 16b shows the contact zone for the output tolerances. Although the conservative bound (dashed box) still suggests possible motion variation failures, the synthesis algorithm rules them out. It also rules out the possible kinematic failure modes that are suggested by the overlap between the lower and upper channel boundaries.

7 Conclusion

We have presented a parameter synthesis algorithm for higher pair mechanical systems based on configuration spaces. The algorithm is given nominal values and tolerance intervals for the system design parameters. It searches the parameter space for revised nominals and tolerances that ensure correct kinematic function. The search is an iterative process in which candidate failure points are generated, tested, and excluded. Nominal parameter values are

changed when possible and tolerance intervals are shrunk otherwise. We have demonstrated the algorithm on a range of mechanisms.

Prior work does not address parameter synthesis for general higher pair systems. Our prior work provides analysis tools for higher pairs. Other prior work provides parameter synthesis algorithms for linkage mechanisms and for cams. We build on this work to obtain a practical parameter synthesis algorithm. The five examples show that it efficiently detects and corrects kinematic failures in complex mechanical systems.

We have implemented the algorithm for planar parts with fixed motion axes. The components that impose this restriction are contact zone construction and configuration space matching. Spatial pairs with fixed motion axes require a straightforward extension to the contact zone module. General planar pairs are harder because their configuration spaces are three-dimensional. General spatial pairs are far harder yet because their configuration spaces are six-dimensional. We expect that the first two cases are worth pursuing, whereas the third is impractical.

Mechanical design involves non kinematic constraints that influence parameter synthesis. Driving forces need to overcome friction and inertia. Structural elements need to resist bending and vibration. Non functional design goals can constrain functional parameters. Each type of constraint is handled separately: rigid-body simulation for dynamics, finite-element analysis for structures, and other types of optimization. Integrating this analysis with kinematic parameter synthesis could shorten the design cycle and improve product quality. Although pure kinematic analysis is less effective than an integrated analysis, it is worthwhile in designs where kinematics play a major role, which is the norm in applications.

Our algorithm can share parameters with other algorithms. The initial tolerances can express any constraint. As long as our algorithm only tightens tolerances, it maintains these constraints. Moving a nominal value outside its initial interval can invalidate a non kinematic constraint. We can forbid these revisions or we can flag them, so the designer can recheck the constraints with the final parameter values.

Another research direction is to integrate an objective function, which encodes design quality, into our synthesis algorithm. A first step would be to allow the algorithm to loosen tolerances, which is the main way to reduce cost. One strategy would be to expand the parameter box to touch the closest failure point outside it. Combining expansion and contraction raises convergence issues that would need to be researched.

An orthogonal research task is to enhance the configuration space matching algorithm. The current matching criterion is structural: same number of com-

ponents, same number of curves per component, and same contact per curve. A geometric matching algorithm would yield a richer understanding of the kinematic variations. One idea is to use computational geometry to identify kinematic features, for example Voronoi diagrams for channels.

Acknowledgments

This research was supported by NSF grants IIS-0082339 and CCR-0306214, by the Purdue Center for Computational Image Analysis and Scientific Visualization, and by the MIC (Ministry of Information and Communication), Korea, under the Information Technology Research Center support program supervised by the Institute of Information Technology Assessment.

References

[1] Panos Papalambros and Douglass Wilde. *Principles of Optimal Design*. Cambridge University Press, 1988.

[2] Ralf Schultheiss and Uwe Hinze. Detect the unexpected - how to find and avoid unexpected tolerance problems in mechanisms. In *Global consistency of tolerances, Proc. of the 6th CIRP Int. Seminar on Computer-Aided Tolerancing, F. van Houten and H. Kals eds., Kluwer*, 1999.

[3] Elisha Sacks, Leo Joskowicz, Ralf Schultheiss, and Min-Ho Kyung. Towards robust kinematic synthesis of mechanical systems. In Pierre Bourdet and Luc Mathieu, editors, *Geometric product specification and Verification: Integration of Functionality*, pages 135–144. Kluwer Academic Publishers, 2003.

[4] G. Erdman, Arthur. *Modern Kinematics: developments in the last forty years*. John Wiley and Sons, 1993.

[5] Rajan Ramaswamy. *Computer Tools for Preliminary Parametric Design*. PhD thesis, Massachussetts Institute of Technology, 1993.

[6] Jorge Angeles and Carlos Lopez-Cajun. *Optimization of Cam Mechanisms*. Kluwer Academic Publishers, Dordrecht, Boston, London, 1991.

[7] Max Gonzales-Palacios and Jorge Angeles. *Cam Synthesis*. Kluwer Academic Publishers, Dordrecht, Boston, London, 1993.

[8] K. W. Chase and A. R. Parkinson. A survey of research in the application of tolerance analysis to the design of mechanical assemblies. *Research in Engineering Design*, 3(1):23–37, 1991.

[9] André Clemént, Alain Rivière, Phillipe Serré, and Catherine Valade. The ttrs: 13 constraints for dimensioning and tolerancing. In *Proc. of the 5th CIRP Int. Seminar on Computer-Aided Tolerancing*, Toronto, 1997.

[10] Daniel Whitney, Olivier Gilbert, and Marek Jastrzebski. Representation of geometric variations using matrix transforms for statistical tolerance analysis. *Research in Engineering Design*, 6(4):191–210, 1994.

[11] Eric Ballot and Pierre Bourdet. A computation method for the consequences of geometric errors in mechanisms. In *Proc. of the 5th CIRP Int. Seminar on Computer-Aided Tolerancing*, Toronto, 1997.

[12] Jingliang Chen, Ken Goldberg, Mark Overmars, Dan Halperin, Karl Bohringer, and Yan Zhuang. Shape tolerance in feeding and fixturing. In P.K. Agarwal, L. E. Kavraki, and M. T. Mason, editors, *Robotics, The Algorithmic Perspective: 3rd Workshop on Algorithmic Foundations of Robotics (WAFR)*. A. K. Peters, 1998.

[13] Matsamoto Inui and Masashiro Miura. Configuration space based analysis of position uncertainties of parts in an assembly. In *Proc. of the 4th CIRP Int. Seminar on Computer-Aided Tolerancing*, 1995.

[14] Kenneth Chase, Spencer Magleby, and Charles Glancy. A comprehensive system for computer-aided tolerance analysis of 2d and 3d mechanical assemblies. In *Proc. of the 5th CIRP Int. Seminar on Computer-Aided Tolerancing*, Toronto, 1997.

[15] O.W. Solomons, F. van Houten, and H. Kals. Current status of cat systems. In *Proc. of the 5th CIRP Int. Seminar on Computer-Aided Tolerancing*, Toronto, 1997.

[16] Charles G. Glancy and Kenneth W. Chase. A second-order method for assembly tolerance analysis. In *Proceedings of the ASME Design Automation Conference*, 1999.

[17] Min-Ho Kyung and Elisha Sacks. Nonlinear kinematic tolerance analysis of planar mechanical systems. *Computer-Aided Design*, 35(10):901–911, 2003.

[18] Elisha Sacks and Leo Joskowicz. Parametric kinematic tolerance analysis of planar mechanisms. *Computer-Aided Design*, 29(5):333–342, 1997.

[19] Zuomin Dong and Gary Wang. automated cost modeling for tolerance synthesis using manufacturing process data, knowledge reasoning, and optimization. In *Fifth CIRP International Seminar on Computer Aided Tolerancing*, Toronto, 1997.

[20] James Guilford and Joshua Turner. Advanced analysis and synthesis of geometric tolerances. *Manufacturing Review*, 6(4):305–313, 1993.

[21] Victor Skowronski and Joshua Turner. Estimating gradients for statistical tolerance synthesis. *Computer-Aided Design*, 28(12):993–942, 1993.

[22] R.G. Fenton, W.L. Cleghorn, and Jing-fan Fu. Allocation of dimensional tolerances for multiple-loop planar mechanisms. *Journal of Mechanisms, Transmissions, and Automation in Design*, 111:465–470, 1989.

[23] Kwun-Lon Ting and Yufeng Long. Performance quality and tolerance sensitivity of mechanisms. *Journal of Mechanisms, Transmissions, and Automation in Design*, 118:114–150, 1996.

[24] Elisha Sacks and Leo Joskowicz. Computational kinematic analysis of higher pairs with multiple contacts. *Journal of Mechanical Design*, 117(2(A)):269–277, June 1995.

[25] Elisha Sacks and Leo Joskowicz. Automated modeling and kinematic simulation of mechanisms. *Computer-Aided Design*, 25(2):106–118, 1993.

[26] Min-Ho Kyung and Elisha Sacks. Parameter synthesis of higher kinematic pairs. *Computer-Aided Design*, 35:567–575, 2003.

[27] Leo Joskowicz and Elisha Sacks. Computational kinematics. *Artificial Intelligence*, 51(1-3):381–416, October 1991.

[28] The American Society of Mechanical Engineers, New York. *ASME Y14.5M-1994 Dimensioning and Tolerancing Standard*, 1994.

[29] I. Artobolevsky. *Mechanisms in Modern Engineering Design*, volume 1–4. MIR Publishers, Moscow, 1979. English translation.