

Kinematic Analysis of Spatial Fixed-Axis Higher Pairs Using Configuration Spaces

Ku-Jin Kim

The Graduate School of Information and Communication
Ajou University, South Korea

Elisha Sacks (corresponding author)

Computer Science Department

Purdue University

West Lafayette, IN 47907, USA

phone: 1-765-494-9026; fax: 1-765-494-0739

Leo Joskowicz

School of Computer Science and Engineering

The Hebrew University of Jerusalem

Jerusalem 91904, ISRAEL

Abstract

We present a kinematic analysis algorithm for spatial higher pairs whose parts rotate around or translate along fixed spatial axes. The part geometry is specified in a parametric boundary representation consisting of planar, cylindrical, and spherical patches bounded by line and circle segments. Kinematic analysis is performed by configuration space construction following the method that we developed for planar pairs. The configuration space of a pair is a complete encoding of its kinematics, including contact constraints, contact changes, and part motions. The algorithm constructs contact curves for all pairs of part features, computes the induced configuration space partition, and identifies the free space components. Spatial contact analysis is far harder than planar analysis because there are 72 types of contact versus 3. We have developed a systematic analysis technique and have used it to derive low-degree equations for all cases, which are readily solvable in closed form or numerically. We demonstrate the implemented algorithm on three design scenarios involving spatial pairs and planar pairs with axis misalignment.

To appear in *Computer Aided Design*.

Introduction

This paper presents a kinematic analysis algorithm for spatial higher pairs. Kinematic analysis determines the configurations (positions and orientations) at which two parts touch and the velocities at which they maintain contact. This is a core design task because contacts are the physical primitives that make mechanical systems out of collections of parts. Systems perform functions by transforming motions via part contacts. The shapes of the interacting parts impose constraints on their motions that largely determine the system function. Designers perform kinematic analysis to derive the evolving sequence of contacts and motion constraints. The results help them simulate the system function, find and correct design flaws, measure performance, and compare design alternatives.

Pairs are classified as lower or higher. A lower pair imposes a single contact constraint 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 shaft. A higher pair imposes multiple constraints due to contacts between pairs of part features (faces, edges, or vertices). For example, gear teeth consist of involute patches whose contacts change as the gears rotate. When two features touch, the relative velocity at the contact point must be tangent to both features. This constraint can be expressed as an algebraic equation in the part configurations whose solution varies smoothly as the contact point moves along the features. When the contact point shifts to another feature, a different contact constraint takes effect, perhaps causing a velocity discontinuity.

Higher pairs are common in mechanical design. Gears and cams are used in all types of mechanical systems. Ratchets, indexers, and other specialized pairs are used in low-torque precision mechanisms, such as sewing machines, copiers, cameras, and VCRs. Higher pairs are more versatile than lower pairs because they can realize multiple functions, and are usually cheaper, lighter, more compact, and more robust than actuators. In a survey of more than 2,500 mechanisms in an engineering encyclopedia, we found that 65% contain higher pairs [1]. When manufacturing variation and wear are taken into account, lower pairs must be analyzed as higher pairs, as in revolute joints with hub play.

Kinematic analysis of higher pairs is a computational bottleneck in mechanical design. The analyst needs to identify the configurations (if any) where each pair of features touch, the contact constraints, and the configurations where contacts change. For parts with tens of features, there are typically thousands of potential contacts, since every feature from one part can, in principle, be in contact with a feature of the other part. However, only a fraction of these potential contacts can occur, since most are prevented by other contacts.

In prior work, we developed a fast, robust kinematic analysis algorithm for planar higher pairs [2, 3]. A pair is planar if its part shapes consist of one or more extruded planar contour slices and the part motions are parallel to the slice plane. The algorithm covers parts whose contour features are line and circle segments. This class is simple enough to allow fast analysis, yet is broad enough to cover almost all planar applications. Prior research does not provide an algorithm for this analysis.

Our algorithm constructs a geometric representation of kinematics called a configuration space, which is widely used in robotics [4]. The configuration space of a pair is a complete encoding of its kinematics, including contact constraints, contact changes, and part motions. It encodes quantitative information, such as contact equations and contact change configurations, and qualitative information, such as failure modes. Industrial case studies [5, 6, 7] show that configuration space construction is an effective kinematic analysis method for planar higher pairs. Extensive testing shows that the implementation is fast and robust.

In this paper, we extend the kinematic analysis algorithm from planar pairs to spatial pairs. The parts are specified in a parametric boundary representation with planar, cylindrical, and spherical patches bounded by line and circle segments. Each part rotates around or translates along a fixed spatial axis. Most spatial systems fall into this class, according to our 2,500 mechanism survey [1] and according to our industrial experience. The main exceptions are precision gears and cams, which require higher-degree surfaces, or non fixed-axis parts, which are infrequent. The algorithm also supports spatial tolerancing of planar systems: planar pairs can be analyzed as spatial pairs to study axis misalignment due to manufacturing variation, assembly error, or wear.

Configuration space construction for spatial pairs is much harder than for planar pairs because of the increased number and geometric complexity of the part features. Instead of two types of features (line and circle), there are six types (plane, sphere, and cylinder patches, line and circle segments, and vertices). Hence, there are 21 spatial contact types (sphere/cylinder, cylinder/arc, vertex/plane, etc) versus 3 planar types (line/arc, arc/arc, line/line). The fixed-axis motion restriction splits each case into four cases. For example, the sphere/cylinder contact splits into rotating sphere/rotating cylinder, rotating sphere/translating cylinder, translating sphere/rotating cylinder, and translating sphere/translating cylinder. All told, there are 72 spatial cases versus 8 planar ones. To cope with this complexity, we have developed a systematic spatial analysis technique and have used it to derive low-degree equations for all cases. We solve most equations in closed form and the rest numerically.

The rest of the paper is organized as follows. The next section describes prior research. The following section illustrates the configuration space method of kinematic analysis on three engineering case studies. The next section outlines the configuration space construction algorithm. The following two sections describe the spatial contact curve construction step and illustrate it on three representative cases. The final section contains a discussion of our results and plans for future work.

Prior research

Table 1 summarizes prior kinematic analysis research. Most research focuses on efficient algorithms for linkages and manipulators, which are systems whose parts are connected by lower pairs and by simple higher pairs with permanent contacts [8, 9, 10]. Prior research on higher pairs addresses planar and spatial gears [12] and cams [11, 13] because of their practical importance. The

| Type | Dofs | Part geometry | Analysis |
|----------------------------|------|----------------------------|--------------------------------------|
| lower pair (linkages) | 1-3 | planar | Schiehlen [8], Erdman [9] |
| | | spatial | Haug [10] |
| planar higher pairs | 1 | cams, gears | Angeles[11], Litvin[12] |
| | | lines, circles, splines | Sacks-Joskowicz [2] |
| | 3 | lines, circles | Sacks [3] |
| spatial higher pairs | 1 | cams, gears | Angeles[13], Litvin[12] |
| | | planes, spheres, cylinders | <i>This paper</i> |
| | 6 | polyhedral | Donald [14] Joskowicz-Taylor [15] |
| | | general | <i>theoretical work</i> [4] |

Table 1: Prior research in kinematic analysis. The columns indicates the type of pair, the number of degrees of freedom per part, the geometry of the part boundary, and representative references of their analysis.

contacts are permanent or change in a known sequence, so the studies focus on contacts between pairs of complex surface features, such as involutes and helicoids.

We developed kinematic analysis algorithms for fixed-axis planar higher pairs [2] and for general planar pairs [3] based on configuration space construction. The algorithms handle contact changes and are fast and robust on engineering applications. Configuration space was first used as a computational tool in robot motion planning [4]. The literature describes configuration space construction algorithms for a polyhedral robot moving amidst fixed polyhedral obstacles [15, 16, 17]. Graphics research provides fast collision detection algorithms for polyhedra in support of simulation [18, 19, 20]. Although a limited form of kinematic analysis, collision detection does not address many aspects of mechanical design.

Design scenarios

We illustrate the configuration space method of kinematic analysis and its use in common design tasks with three examples: axis misalignment in planar dwell gears, backlash reduction in a spatial indexing pair, and jamming detection in a bevel gear mechanism.

Axis misalignment in planar dwell gears

The first example is a planar dwell gear pair (Figure 1a). The driver is a concentric arc segment mounted on a gear sector of diameter 240mm. The follower is a star-shaped disk mounted on a gear sector of diameter 100mm. The arc segment, disk, and gears are 20mm thick. Both parts rotate around their centers. As the driver rotates, the gear sector engages and turns the gear wheel.

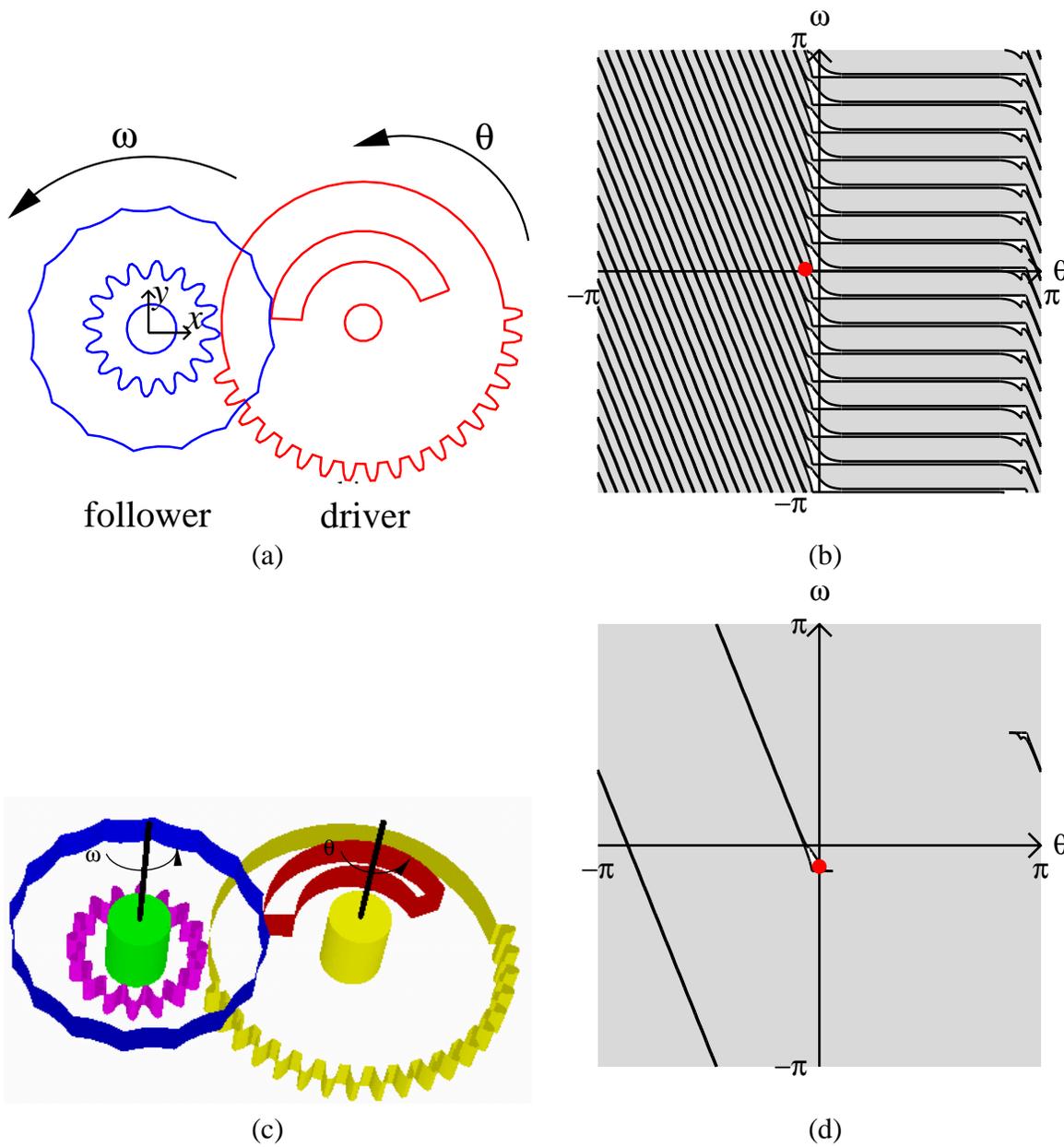


Figure 1: Dwell gears: (a) top view with no axis tilt, and; (b) configuration space; (c) perspective see-through view with two degree axis tilt, and ; (d) configuration space.

When the sector disengages, the concentric arc segment engages one of the 16 concave surfaces of the star-shaped disk, thereby locking it in a dwell position. A full rotation of the driver advances the follower by 1/16th of a rotation. For this example, we approximate each gear tooth involute boundary segment by two circle segments. The design task is to assess the sensitivity of the pair to axis misalignment.

We describe the kinematics of the pair within the configuration space representation. The configuration space of the pair is a two-dimensional manifold whose coordinates are the part degrees of freedom: θ is the driver orientation and ω is the follower orientation (Figure 1b). The dot marks the configuration displayed in Figure 1a. Configuration space partitions into blocked space where the parts overlap (the gray area), free space where they do not touch (the white area), and contact space where they touch without overlap (the black curves in between). Contact space partitions into contact curves that represent contacts between pairs of part features.

The free and contact spaces encode the pair kinematics. They consist of 16 channels, each of which has a horizontal segment and a slanted segment connected by openings along the lines $\theta = 0$ and $\theta = 3$ radians (344 degrees). The horizontal segments contain the configurations where the driver arc locks the follower disk, hence θ changes and ω is nearly constant. The slanted segments contain the configurations where the gears engage, hence ω is a nearly linear function of θ . The gap between the channel boundaries represents part play: the follower can rotate by the channel height when the driver stands still.

We perform a spatial analysis to assess the sensitivity of the pair to axis misalignment. The axis misalignment transforms the contacts between planar features into spatial ones. For example, circle/circle contact becomes cylinder/cylinder or cylinder/circle contact. Tilting the driver axis by two degrees around its y axis causes jamming (Figure 1c–d). Tilting the axis by the same amount around the x axis narrows the channels (reduces play), but does not produce jamming (a qualitative kinematic change).

Manual kinematic analysis of the dwell gears is impractical because of the large number of contacts. There are over 2,000 pairs of features, called potential contacts, that might generate contact curves. Only 156 of these generate actual contacts: 36 between a pair of gear teeth and 120 between the locking arc and a disk arc. The 36 tooth/tooth contacts give rise to $36 \times 16 \times 16 = 9,216$ contact curves for all the teeth, while the 120 arc/disk contacts give rise to $120 \times 16 = 1,920$ curves for all the disks. All these curves must be analyzed to construct the configuration space.

Backlash reduction in spatial indexing pair

The second example is a spatial indexing pair (Figure 2a). The driver (left) consists of two cylindrical plates of radius 100mm connected by a slanted crossover. It rotates around a vertical axis through its center and perpendicular to the cylindrical plates. The follower (right) is a gear of radius 100mm with 21 rectangular teeth that rotates around an axis through its center and perpendicular to its surface. The two axes are perpendicular. One full rotation of the driver advances the follower by one tooth. The crossover rotates the follower then the guides lock it for the remainder of the

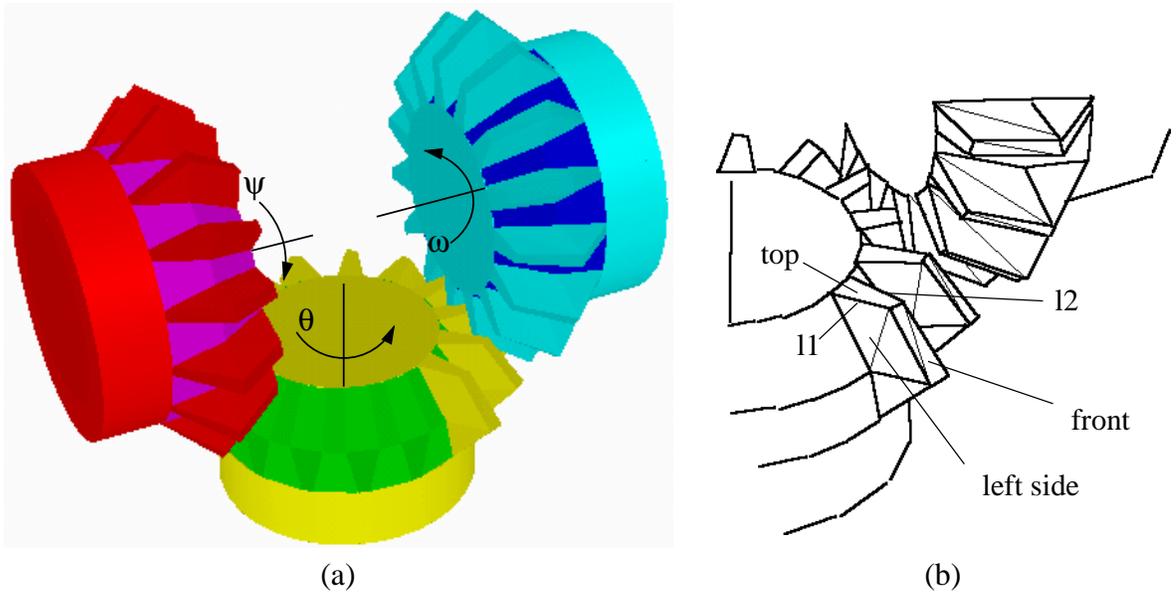


Figure 3: (a) Bevel gear mechanism; (b) detail of tooth geometry and gear engagement.

slanted channels. The driver engages the full gear at the left side of a channel, rotates it by 2.22 radians (127 degrees) in the channel, and breaks contact at the right side. The driver/left gear space is obtained by a horizontal shift of 180 degrees. The pair has 67 contacts out of 1,764 potential contacts.

The design task is to ensure that the gears never jam. They cannot jam while they are engaged because the channel sides are smooth. It remains to confirm that they always engage. Figure 4b shows the motion path due to counterclockwise rotation of the driver near the configuration where it engages a driven gear. Contact occurs on a curve that slopes right. The driving motion projects onto this curve to yield a coupled motion that follows the curve into the channel mouth. Hence, the gears cannot jam during engagement. Figure 4c shows a configuration space in which the contact curve is vertical, so the driving motion is blocked. This space comes from a faulty design (taken from a gear manual) in which the sides of the teeth have the same slope (features l_1 and l_2 in Figure 3b), whereas the redesigned teeth have different slopes. Next we assess the sensitivity of the kinematic function to axis misalignment. Configuration space construction shows that a two degree deviation greatly increases backlash, while a three degree deviation causes jamming.

Configuration space construction

Figure 5 shows our configuration space construction algorithm. The inputs are the geometry, motion axis, and degree of freedom (rotation or translation) for each part, and a configuration space accuracy. The part geometry is specified in a custom parametric boundary representation with

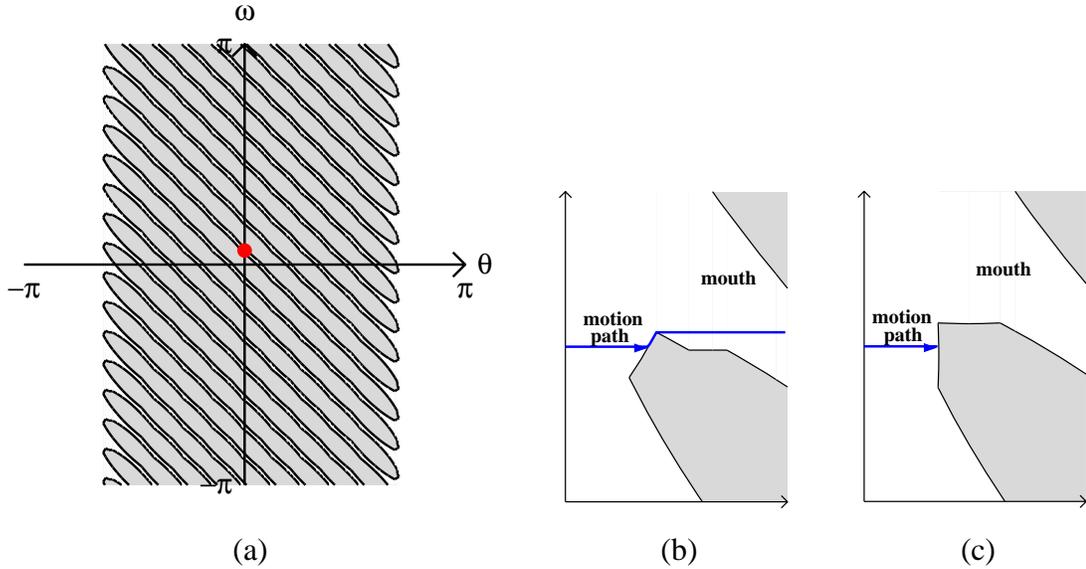


Figure 4: Driver/right gear: (a) configuration space; (b) detail of the configuration space with the motion path (thick line) showing correct engagement, and (c) blocking due to faulty design.

planar, cylindrical, and spherical patches and with line and circle segments as patch boundaries. A planar patch is the subset of a plane bounded by a rectangle. A cylindrical patch is the cross-product of a circle segment with a perpendicular interval. It is bounded by top and bottom circle segments and by two line segments that connect their start/end points. A spherical patch is the set of points on a sphere whose latitude and longitude lie in given intervals. It is bounded by circle segments.

Figure 6 illustrates the algorithm on the spatial indexer shown in Figure 2. Step 1 constructs contact curves for all pairs of part features (Figure 6a). It generates piecewise linear approximations of the curves that are accurate to the specified input accuracy. The contact curves for a pair of features are the configurations in which the features would touch if there were no other features to interfere. Step 2 computes the partition of the configuration space into connected components induced by the contact curves (Figure 6b). The partition is computed via a line sweep and is encoded in the standard winged-edge representation of computational geometry [21]. The elements of the partition are connected components of configuration space bounded by loops of contact curves. Each component lies wholly in free space or wholly in blocked space. Step 3 classifies the components and returns the free space ones. A component is blocked if the parts overlap at a single, arbitrarily chosen interior configuration.

The algorithm is not provably correct because its numerical computations and computational geometry can fail on degenerate inputs. We have used heuristics, software engineering, and extensive testing to make it robust.

Input: part geometry, motion axes and degree of freedom, accuracy.

1. Construct contact curves.
For each feature pair, do:
 - (a) Structure analysis
 - (b) Branch construction (point analysis)
 - (c) Branch pruning
2. Partition configuration space.
3. Classify components into free and blocked space.

Output: Free space components.

Figure 5: Configuration space construction algorithm.

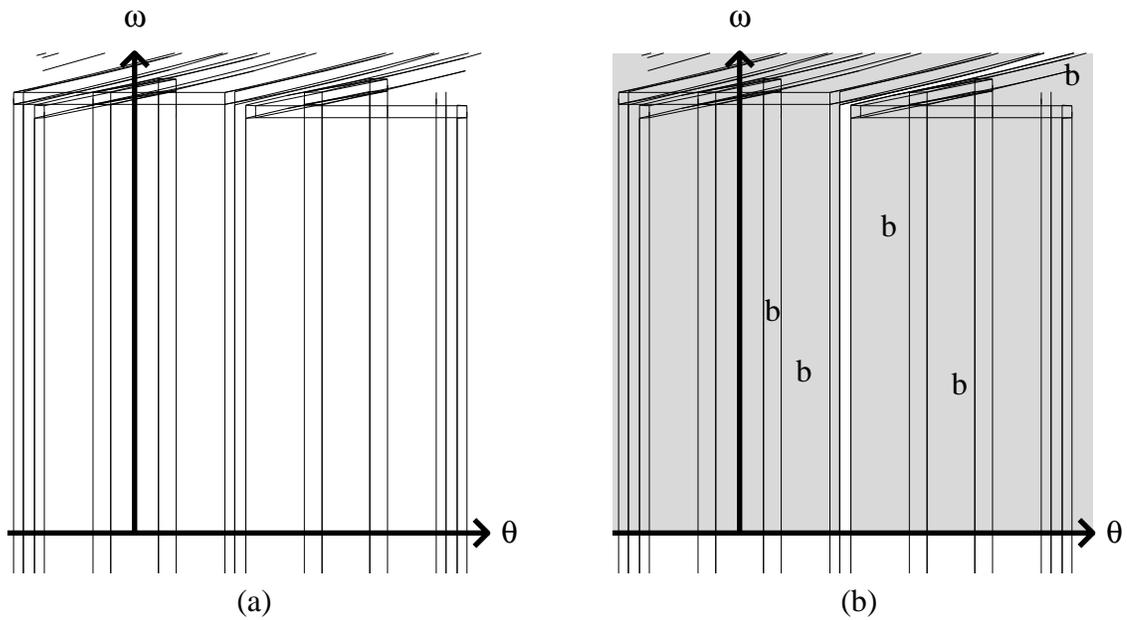


Figure 6: Detail of configuration space construction for spatial indexer: (a) contact curves; (b) partition with “b” on selected blocked components.

Contact curve construction

The next two sections describe contact curve construction. The partition and classification steps of configuration space construction are independent of the part geometry and are described in prior work [2]. The discussion applies to a pair of features. All pairs of part features are enumerated (the potential contacts) and the actual contact curves are returned.

Let a and b be features on two parts with configuration space coordinates u and v . Let $a(u)$ denote the space occupied by feature a in configuration u . The contact curves for a and b are the solution set of one equality and several inequalities in u and v . The equality holds when the underlying algebraic features touch. These are planes for planar patches, spheres for spherical patches, cylinders for cylindrical patches, lines for line segments, circles for circle segments, and vertices for themselves. The solution set, called the algebraic contact curve, is a plane curve. The inequalities hold at curve points where the contact point of the algebraic features lies on the actual features. The contact curves are constructed via structure analysis, branch construction, and branch pruning.

Structure analysis partitions the algebraic contact curve into simple pieces, called branches, that are monotone in u and whose interiors are disjoint (Figure 7). The u values where branches begin and end are called critical values. The critical values are computed by sweeping b along its motion path (a circle or line) and computing the u values where $a(u)$ is tangent to the swept volume. Tangency is a necessary condition for a u_0 to be critical. Otherwise, $a(u_0)$ intersects the swept volume transversely or is separated from it. If $a(u_0)$ intersects the swept volume, then $a(u)$ intersects it for all u in a neighborhood of u_0 . If $a(u_0)$ is disjoint from the swept volume, there are no (u_0, v) contact configurations. Either way, there are no u_0 curve endpoints. A tangent point might not be critical if the tangency is degenerate. This case is rare and does not effect the correctness of structure analysis. The only penalty is that one branch is split in two by a spurious criticality.

Branch construction approximates the branches to the input accuracy. The critical values are sorted and each interval between consecutive values is treated separately. On an interval, say $[u_1, u_2]$, we use Brent's method (a standard optimization algorithm [22]) to find the configuration u_m that maximizes the distance between an actual branch and the line segment between its endpoints. This distance bounds the error in approximating the branches with the segments. Figure 7 illustrates the computation on the first interval: the line segment is dashed, the maximum occurs on the upper branch, and the error at u_m is dotted. If the error is less than the input accuracy, the line segments are returned. Otherwise, the intervals $[u_1, u_m]$ and $[u_m, u_2]$ are approximated recursively and the results are concatenated. The method whereby the optimizer computes the branch v values at u_m , called point analysis, is described in the next section.

Branch pruning constructs the contact curves by pruning the portions of the approximate algebraic curves where the contact point lies off the actual features. At each curve configuration, we compute the contact points and the corresponding parameter values for the two features. A line segment is parameterized linearly. A circle segment is parameterized by its polar angle. A spher-

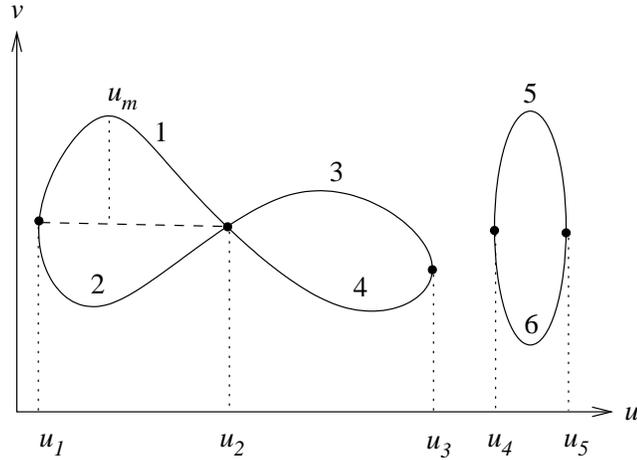


Figure 7: Example of an Algebraic contact curve with six branches and five critical values.

ical patch is parameterized with intervals of latitudes and longitudes. A cylindrical patch inherits the parameterizations of its axis and base. A planar patch is parameterized by height and width. A contact configuration is removed if either of its features is out of range. A feature is out of range if any of its parameters is out of range. The remaining configurations are the contact curves.

Each combination of feature and motion types requires a separate form of structure and point analysis. We have developed a systematic treatment that reduces these to canonical cases. Table 2 lists the cases for two rotating parts. The rotation/translation and translation/translation cases are very similar with simpler solutions. The feature types are cylinders, spheres, planes, and circles. Lines and vertices are modeled as zero-radius cylinders and spheres. Axis aligned cylinders are handled separately from general cylinders because they are common and have simpler solutions. Likewise, circles whose axis is parallel to the rotation axis are handled separately from general circles. All cases have been implemented with the exception of those involving contacts between two general cylinders or their boundary circles, which are rare.

Structure and point analysis are simplified by shrinking one feature and offsetting the other feature, so that the adjusted features have the same contact configurations as the originals. The analysis is further simplified by reducing most cases to planar geometry. Table 3 lists the computational geometry operations for two rotating parts. The other motion types are very similar with simpler solutions. Axis-aligned cylinders and circles are split into two cases each, depending on whether the rotation axis is parallel or perpendicular to the cylinder axis. Each operation intersects two curves, intersects a curve with a surface, or constructs the tangent to a curve at a point. The table lacks entries for those Table 2 cases that require nonstandard constructions. All the cases are implemented in closed form.

Each pair of feature types requires a separate form of branch pruning. The motion type is not taken into account, since the analysis occurs at a single configuration. All the cases are imple-

| feature 1 | feature 2 | structure analysis | point analysis |
|-------------|--|---|---|
| g. cylinder | g. cylinder a. cylinder sphere g. circle v. circle | numerical* numerical numerical numerical* numerical | numerical* closed-form (2) closed-form (4) numerical* numerical |
| a. cylinder | a. cylinder sphere g. circle v. circle | closed-form (2) closed-form (4) numerical* numerical | closed-form (2) closed-form (2) numerical* numerical |
| sphere | sphere v. circle | closed-form (4) closed-form (4) | closed-form (2) closed-form (2) |
| plane | sphere g. circle v. circle | closed-form(4) numerical* closed-form (4) | closed-form(2) numerical* closed-form (2) |
| g. circle | sphere g. circle v. circle | numerical numerical* numerical | closed-form(4) numerical* closed-form (2) |
| v. circle | v. circle | closed-form (2) | closed-form (2) |

Table 2: Coverage of structure and point analysis for rotation/rotation. The first two columns indicate the pair of features in contact: g. cylinder denotes general cylinders, a. cylinder axis aligned cylinders, g. circle general circles, and v circle circles whose axis is parallel to the rotation axis. The next two columns indicate the type of solution for structure and point analysis. The equation degree appears in parenthesis for closed form solutions. Unimplemented cases are starred.

| feature 1 | feature 2 | structure analysis | point analysis |
|-------------|---|--|--|
| v. cylinder | v. cylinder v. circle | circle/line intersection offset ellipse/circle intersection | circle/line intersection offset ellipse/circle intersection |
| h. cylinder | h. cylinder v. cylinder v. circle | circle tangent ellipse tangent | ellipse tangent circle/line intersection offset ellipse/circle intersection |
| sphere | v. cylinder h. cylinder sphere v. circle | ellipse/circle intersection plane/circle intersection torus/circle intersection torus/circle intersection or plane/circle intersection | circle/circle intersection circle tangent circle/circle intersection circle/circle intersection |
| plane | sphere v. circle | circle tangent circle tangent | plane/circle intersection plane/circle intersection |
| v. circle | v. circle | line/circle intersection or circle/circle intersection | circle/circle intersection |

Table 3: Computational geometry in structure and point analysis for rotation/rotation. The first two columns indicate the pair of features in contact: h. cylinder, v.cylinder, h. circle v. circle denote cylinders and circles whose axes are parallel or perpendicular to the rotation axis. The next two columns indicate the type of geometric operation.

mented in closed form.

Three representative cases

We describe structure and point analysis on three representative cases. The first illustrates the derivation in a relatively simple setting. The second is the most complicated contact that has a closed form solution. The third is the only implemented contact that lacks a closed form solution. We describe branch pruning on the first case; the other cases have the same form, but involve more complicated geometric constructions. A technical report [23] contains detailed descriptions of all the cases.

Rotating cylinder/translating sphere

Figure 8a shows a rotating cylinder/translating sphere contact. The cylinder has axis $\mathbf{p} + \alpha\mathbf{z}$ and rotates around the global z axis. Its degree of freedom is the angle θ between its x axis and the global x axis. The sphere translates parallel to the global y axis along direction \mathbf{v} , so its degree of freedom is the y coordinate of its center \mathbf{q} . The cylinder radius is r and the sphere radius is s . Cylinder and sphere configurations are written as $C(\theta)$ and $S(y)$.

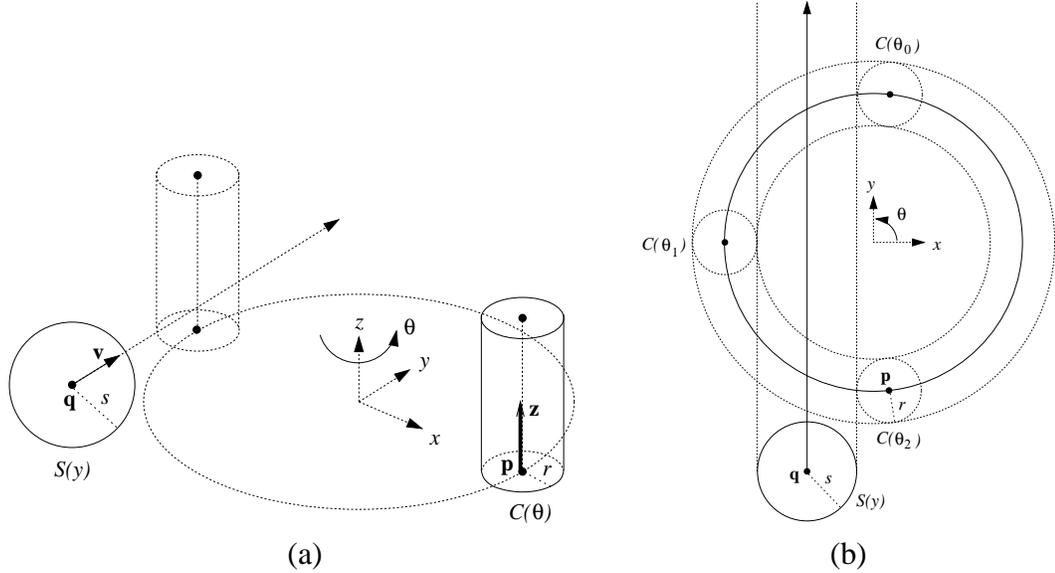


Figure 8: (a) spatial view of rotating cylinder/translating sphere; (b) projection onto xy plane. The cylinder is shown at several sample configurations.

We simplify the analysis by projecting onto the xy plane. The contact curves are unchanged because the cylinder axis is always perpendicular to this plane. The cylinder projects to a circle that rotates around the origin and the sphere projects to a circle that translates vertically (Figure 8b).

Figure 9 illustrates point analysis. We shrink the sphere (now a circle) to its center (x_0, y) with x_0 a known constant. We offset the cylinder (now a circle) by the sphere radius s and obtain the circle $(x - o_x)^2 + (y - o_y)^2 = (r + s)^2$ whose center is $\mathbf{o}(\theta) = (o_x, o_y)$. This feature adjustment preserves contacts by definition of the offset operation. As y varies, the sphere center traces the vertical line $x = x_0$. This line is intersected with the offset circle to compute y_a and y_b .

In structure analysis, we compute the θ values where the rotated cylinder, $C(\theta)$, is tangent to the volume swept by the sphere along its motion axis. Figure 8b shows the three solutions in the xy plane. The cylinder projects to a circle and the swept volume projects to the region between two vertical lines. Figure 10 shows the critical value computation. We shrink the cylinder to its center $\mathbf{o}(\theta)$, which generates a circle. We offset the vertical lines by the cylinder radius r to obtain $x = x_0 \pm r$. The cylinder is tangent to the swept sphere at the three θ values where the circle intersects the verticals.

In branch pruning, the contact point is $\mathbf{c} = \mathbf{p}(\theta) + r\mathbf{d}$ with \mathbf{d} a unit vector that points from $\mathbf{p}(\theta)$ to $\mathbf{q}(y)$. We compute $\mathbf{f} = \mathbf{q}(y) - \mathbf{p}(\theta)$, $\mathbf{e} = \mathbf{f} - (\mathbf{f} \cdot \mathbf{z})\mathbf{z}$, and $\mathbf{d} = \mathbf{e}/(r + s)$. The cylinder parameters are the cylindrical coordinates of \mathbf{c} in the cylinder frame and sphere parameters are the spherical coordinates of \mathbf{c} in the sphere frame. The four parameters must lie in the intervals that define the two patches.

The rotating cylinder/translating sphere analysis carries over to the other cylinder/sphere mo-

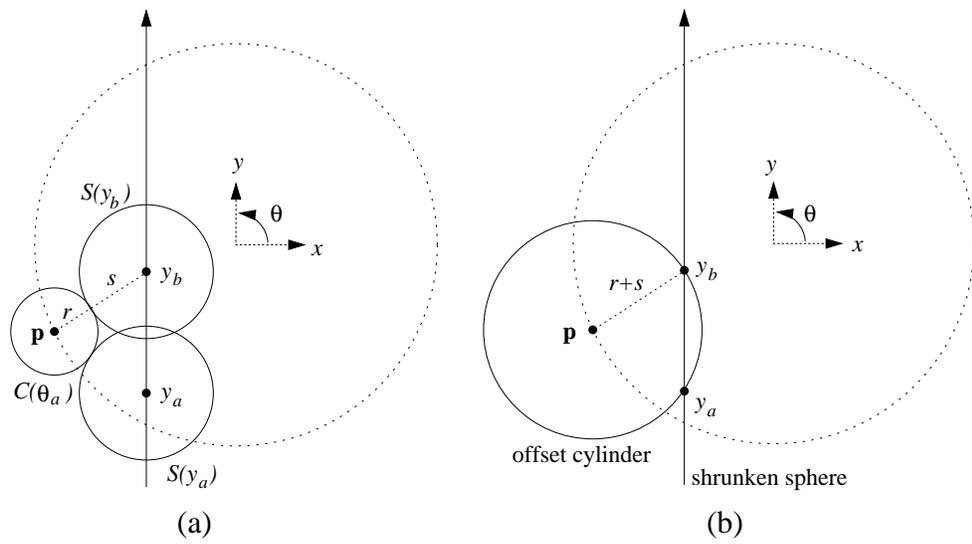


Figure 9: Point analysis: top view of (a) original features; (b) adjusted features.

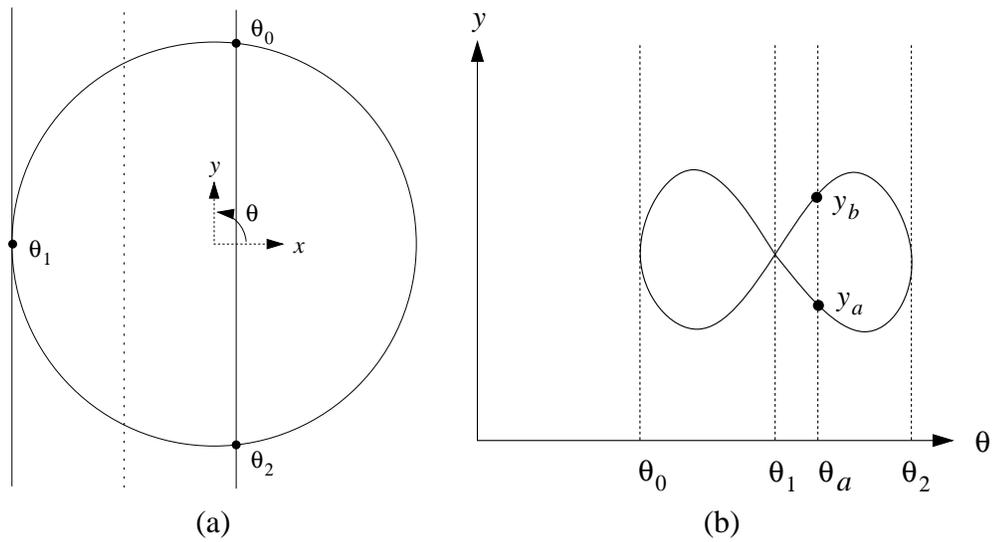


Figure 10: Structure analysis: (a) top view of adjusted features; (b) analysis results.

tions with small changes. Suppose the sphere rotates and the cylinder is unchanged. The only change in point analysis is that the projected offset cylinder is intersected with a circle (the motion path of the projected shrunken sphere) instead of with a line. The only change in structure analysis is that the cylinder center circle is intersected with two circles instead of with two lines, since the swept volume of the sphere is a torus instead of a rectangle. The only change in branch pruning is $\mathbf{q}(y)$ is replaced with $\mathbf{q}(\omega)$ where ω is the sphere angle. A similar analysis applies when the cylinder translates and the sphere rotates or when both features translate.

Rotating cylinder/rotating cylinder

Figure 11a shows a rotating cylinder/rotating cylinder contact. The first cylinder has axis $\mathbf{p} + \alpha\mathbf{u}$, has radius r , and rotates by θ around an axis parallel to \mathbf{u} . The second has axis $\mathbf{q} + \beta\mathbf{v}$, has radius s , and rotates by ω around an axis parallel to \mathbf{v} . Configurations of the two cylinders are written as $C_1(\theta)$ and as $C_2(\omega)$.

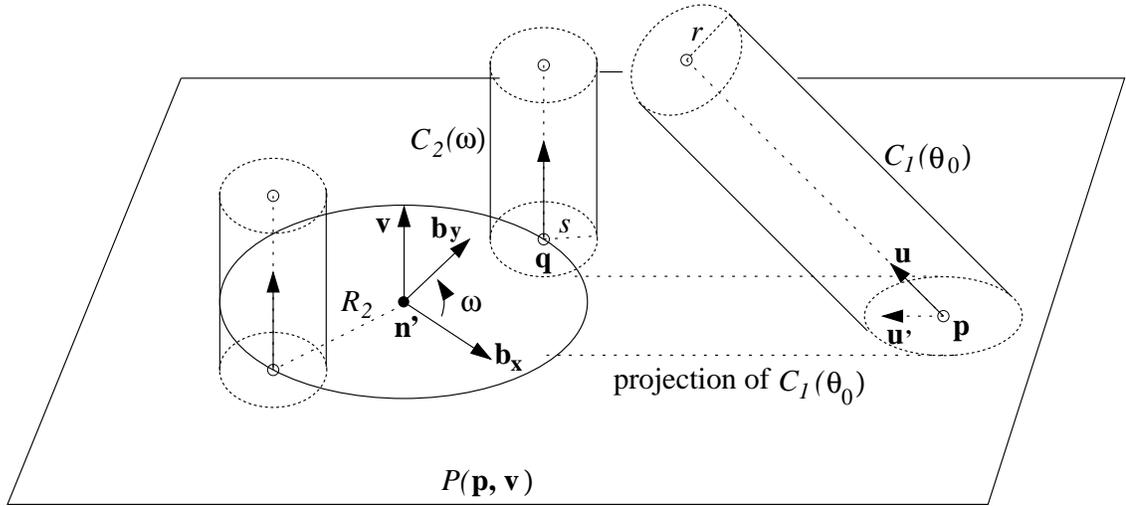
In point analysis, we project onto the plane through \mathbf{p} and perpendicular to \mathbf{v} , which preserves contacts as before. The cylinder $C_1(\theta_0)$ projects to a region bounded by two parallel lines, while $C_2(\omega)$ projects to a rotating circle (Figure 11b). We shrink the circle to its center \mathbf{q} and offset the lines by the circle radius s (Figure 11c). As before, this feature adjustment preserves contacts by definition of the offset operation. We compute ω by intersecting the motion circle of \mathbf{q} with the offset lines.

In structure analysis, we compute the θ values where $C_1(\theta)$ is tangent to the volume swept by $C_2(\omega)$ (Figure 12a). The swept volume is a hollow cylinder bounded by cylinders of radii $R_2 \pm s$ with R_2 the distance between the cylinder axis and the rotation axis. We project onto a plane perpendicular to \mathbf{u} : $C_1(\theta)$ projects to a rotating circle and the hollow cylinder projects to two regions bounded by four parallel lines (Figure 12b). We shrink the circle to its center, offset the lines by its radius s , and intersect the motion circle of the center with the offset lines (Figure 12c).

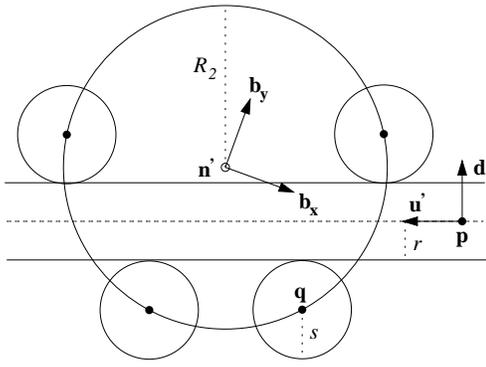
Rotating cylinder/rotating circle

Figure 13a shows a rotating cylinder/rotating circle contact. The cylinder has axis $\mathbf{p} + \alpha\mathbf{u}$, has radius r , and rotates by θ around an axis parallel to \mathbf{u} . The circle lies in the plane through \mathbf{n} and perpendicular to \mathbf{v} , has center \mathbf{c} and radius s , and rotates around the axis through \mathbf{p} and parallel to \mathbf{v} . Cylinder and circle configurations are written as $C(\theta)$ and $K(\omega)$.

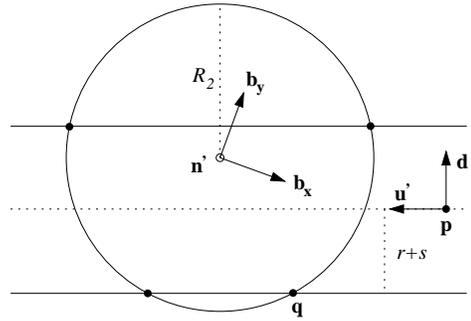
In point analysis, we intersect $C(\theta_0)$ with the circle plane to obtain an ellipse. The circle is tangent to the cylinder exactly when it is tangent to this ellipse. We shrink the circle to its center \mathbf{c} , offset the ellipse by s , and intersect the motion circle of \mathbf{c} with the offset. A closed form solution is impossible (equivalent to solving an irreducible equation of degree 8), so we use a custom numerical method, which is described in the technical report. In structure analysis, we sweep the circle to obtain an annulus, shrink the cylinder to its axis, and offset the annulus boundary circles by r . We project onto a plane perpendicular to \mathbf{u} : the cylinder axis projects to a rotating point and



(a)

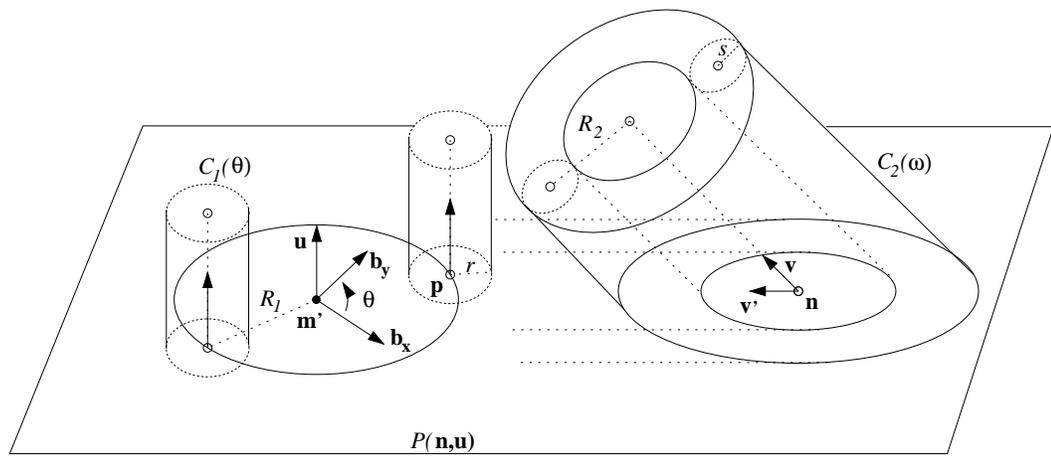


(b)

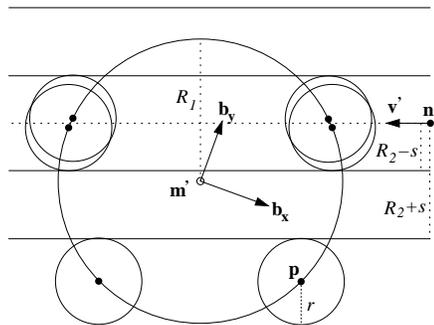


(c)

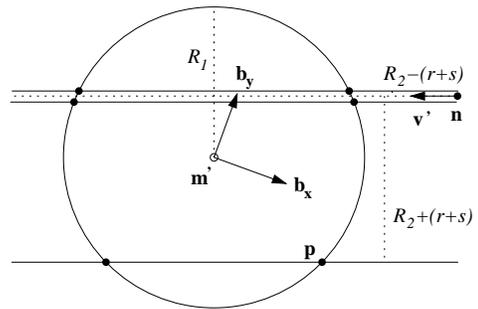
Figure 11: Point analysis: (a) spatial view of rotating cylinder/rotating cylinder; (b) projection onto xy plane; (c) adjusted features.



(a)



(b)



(c)

Figure 12: Structure analysis: (a) spatial view of rotating cylinder/swept cylinder; (b) projection onto xy plane; (c) adjusted features.

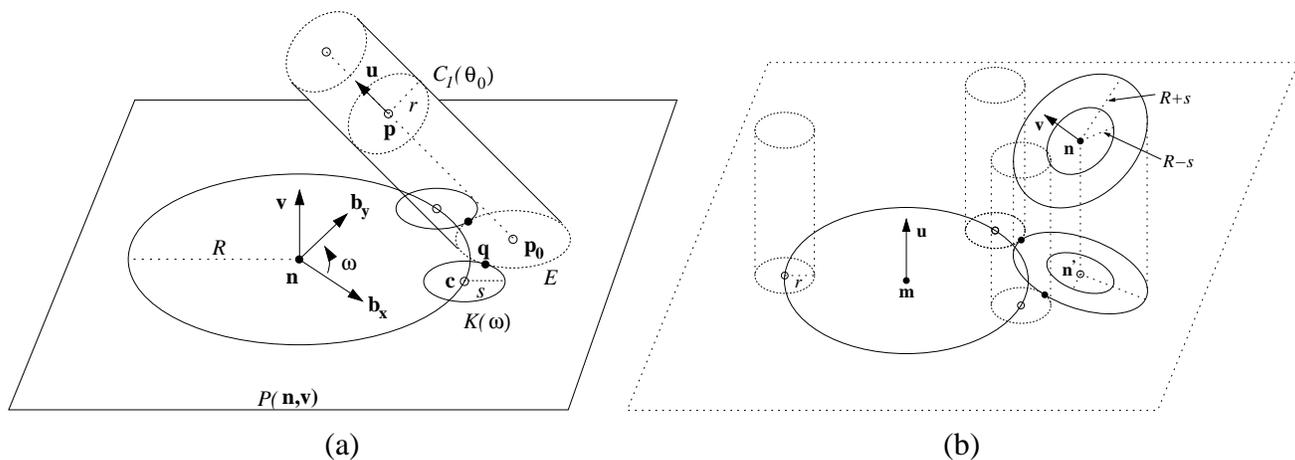


Figure 13: Rotating cylinder/rotating circle contact: (a) point analysis; (b) structure analysis.

the offset circles project to ellipses (Figure 13b). We intersect the motion circle with the ellipses by our numerical method to obtain the critical θ values.

Conclusion

We have presented a kinematic analysis algorithm for fixed-axis spatial higher pairs via configuration space construction. The input is the part geometry and degrees of freedom and the output is a geometric encoding of the part contacts and contact changes. We have demonstrated the algorithm on three mechanical design scenarios. The examples show that our approach frees designers from manual derivation of contact curves and contact changes, helps them understand complex part interactions, and facilitates function validation. We have implemented the algorithm and tested it on tens of examples with thousands of contacts, including planar and spatial Geneva pairs, parallel and orthogonal gears, and cam/follower pairs.

The design scenarios illustrate that complex mechanical parts can be modeled with our class of patches. In practice, we can model almost any part geometry to a desired accuracy with a moderate numbers of patches. Thus, our algorithm is broadly applicable to spatial higher pairs. In contrast, prior work on spatial pairs provides exact solutions for shapes that we approximate, but does not address the harder problem of contact changes.

There are several directions for future research. The first task is to develop design software for spatial pairs. In prior work, we developed software for dynamical simulation [24], kinematic tolerance analysis [25, 26], and parametric design [27, 28] of planar mechanical systems. The algorithms extend to spatial pairs without major changes. We need to validate the impact model for the dynamical simulator and to formulate sensitivity equations for the new contact equations.

The next task is to extend configuration space construction to other useful features, such as

helicoids and tori for helical gears and screws. We can approximate these features with planes, cylinders, and spheres or can develop new contact equations for which numerical solutions will probably be required. The ultimate challenge is to analyze general spatial pairs. Computing full configuration spaces for six degrees of freedom is impractical and often unnecessary. Instead, we plan to develop an incremental algorithm that constructs the portion of the configuration space that is relevant to the design problem.

Configuration space construction combines numerical computation with computational geometry on large datasets. This type of computation is prone to robustness problems due to the fundamental mismatch between floating point arithmetic and real geometry. We have addressed these problems with heuristics, software engineering, and extensive testing. A fast, provably robust algorithm is a topic for future research.

Acknowledgments

We thank Iddo Drori for his contribution to the initial research and implementation phase. Sacks was supported by NSF grants IIS-0082339 and CCR-9617600 and by the Purdue Center for Computational Image Analysis and Scientific Visualization. Sacks and Joskowicz were supported by a Ford University Research Grant, the Ford ADAPT 2000 project, and by grant 98/536 from the Israeli Academy of Science. Kim was supported by the above grants, by a Korea Science and Engineering Foundation Overseas Postdoctoral Fellowship, and by the Brain Korea 21 Project.

References

- [1] Leo Joskowicz and Elisha Sacks. Computational kinematics. *Artificial Intelligence*, 51(1-3):381–416, October 1991. reprinted in [29].
- [2] 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.
- [3] Elisha Sacks. Practical sliced configuration spaces for curved planar pairs. *International Journal of Robotics Research*, 18(1):59–63, January 1999.
- [4] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.
- [5] Elisha Sacks, Leo Joskowicz, Ralf Schultheiss, and Uwe Hinze. Computer-assisted kinematic tolerance analysis of a gear selector mechanism with the configuration space method. In *25th ASME Design Automation Conference*, Las Vegas, 1999.
- [6] Elisha Sacks and James Allen. Mems functional validation using the configuration space approach to simulation and analysis. In *Second International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors, and Actuators*, 1999.

- [7] Elisha Sacks and Steven M. Barnes. Computer-aided kinematic design of a torsional ratcheting actuator. In *Proceedings of the Fourth International Conference on Modeling and Simulation of Microsystems*, Hilton Head, SC, 2001.
- [8] W. Schiehlen. *Multibody systems handbook*. Springer-Verlag, Berlin, 1990.
- [9] G. Erdman, Arthur. *Modern Kinematics: developments in the last forty years*. John Wiley and Sons, 1993.
- [10] Edward J. Haug. *Computer-Aided Kinematics and Dynamics of Mechanical Systems*. Allyn and Bacon, Needham Heights, MA, 1989.
- [11] Jorge Angeles and Carlos Lopez-Cajun. *Optimization of Cam Mechanisms*. Kluwer Academic Publishers, Dordrecht, Boston, London, 1991.
- [12] Faydor L. Litvin. *Gear Geometry and Applied Theory*. Prentice Hall, New Jersey, 1994.
- [13] Max Gonzales-Palacios and Jorge Angeles. *Cam Synthesis*. Kluwer Academic Publishers, Dordrecht, Boston, London, 1993.
- [14] Bruce R. Donald. A search algorithm for motion planning with six degrees of freedom. *Artificial Intelligence*, 31(3):295–353, 1987.
- [15] L. Joskowicz and R. H. Taylor. Interference-free insertion of a solid body into a cavity: An algorithm and a medical application. *International Journal of Robotics Research*, 15(3):211–229, 1996.
- [16] F. Avnaim, J. D. Boissonnat, and B. Faverjon. A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. In *IEEE International Conference on Robotics and Automation*, 1988.
- [17] Randy C. Brost. *Analysis and planning of planar manipulation tasks*. PhD thesis, Carnegie-Mellon University, 1991. Available as Technical Report CMU-CS-91-149.
- [18] David Baraff. *Dynamic Simulation of Non-Penetrating Rigid Bodies*. PhD thesis, Cornell University, 1992.
- [19] Ming C. Lin, Dinesh Manocha, Jon Cohen, and Stefan Gottschalk. Collision detection: Algorithms and applications. In Jean-Paul Laumond and Mark Overmars, editors, *Algorithms for Robotic Motion and Manipulation*, pages 129–141. A. K. Peters, Boston, MA, 1997.
- [20] Brian Mirtich and John Canny. Impulse-based dynamic simulation. In Goldberg et al. [29].
- [21] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, editors. *Computational Geometry: Algorithms and Applications*. Springer, Berlin, 1997.

- [22] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, Cambridge, England, 1990.
- [23] Ku-Jim Kim, Elisha Sacks, and Leo Joskowicz. Kinematic analysis of spatial fixed-axes pairs using configuration spaces. Technical Report CSD-TR 99-036, Purdue University, 1999. Submitted for publication.
- [24] Elisha Sacks and Leo Joskowicz. Parametric kinematic tolerance analysis of planar mechanisms. *Computer-Aided Design*, 29(5):333–342, 1997.
- [25] Elisha Sacks and Leo Joskowicz. Dynamical simulation of planar systems with changing contacts using configuration spaces. *Journal of Mechanical Design*, 120(2):181–187, June 1998.
- [26] Elisha Sacks and Leo Joskowicz. Parametric kinematic tolerance analysis of general planar systems. *Computer-Aided Design*, 30(9):707–714, August 1998.
- [27] Min-Ho Kyung and Elisha Sacks. Computer-aided synthesis of higher pairs via configuration space manipulation. In *Proceedings of the International Conference on Robotics and Automation*, Seoul, 2001.
- [28] Min-Ho Kyung and Elisha Sacks. Parameter synthesis of higher kinematic pairs. Technical Report CSD-TR 01-020, Purdue University, 2001.
- [29] K. Goldberg, D. Halperin, J.C. Latombe, and R. Wilson, editors. *The Algorithmic Foundations of Robotics*. A. K. Peters, Boston, MA, 1995.