Comprehensive Kinematic Analysis of Higher Pairs with Two Degrees of Freedom

Elisha Sacks
Computer Science Department
Princeton University
Princeton, NJ 08544

Leo Joskowicz
IBM T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

April 26, 1993

Abstract

We present a comprehensive kinematic analysis program for higher pairs with two degrees of freedom, such as gears, cams, ratchets, and escapements. The program, called HIPAIR, derives the contact constraints and contact transitions of a pair by computing its configuration space. It generates the contact curves of the interacting features, finds their intersection points, and partitions the resulting space. It derives displacement, velocity, and acceleration plots by simulating driving motions in configuration space. HIPAIR handles intermittent, simultaneous, and degenerate contacts between planar and 2.5D parts. This class contains 90% of 2.5D pairs and 80% of all higher pairs according to our survey of 2500 mechanisms from an engineering encyclopedia. HIPAIR analyzes pairs of average shape complexity in under five seconds on a workstation. It facilitates the design and analysis of multipart mechanisms that contain higher pairs. It helps designers detect part interference, establish part clearances, identify functional surfaces, and find jamming and lock-up configurations.

Submitted to the *ASME Journal of Mechanical Design.*

1 Introduction

We present a comprehensive kinematic analysis program for higher kinematic pairs with two degrees of freedom, such as gears, cams, ratchets, and escapements. Higher pairs are rigid parts that interact via point or curve contacts. The contacts can be permanent or intermittent. In contrast, lower pairs are rigid parts that interact via permanent surface contacts, such as pin and screw joints. Higher pairs play an essential role in mechanism design, especially in precision mechanisms. The most common pairs, gears and cams, appear in all types of mechanisms. Others serve as path, function, and motion generators in low-speed, low-torque mechanisms, such as sewing machines, copiers, cameras, and VCRs. They are typically cheaper, lighter, more compact, and more robust than lower pair or actuator controlled alternatives.

Kinematic analysis determines the possible behaviors of moving parts consistent with their shapes and degrees of freedom. It computes the contact constraints and contact transitions of the parts and derives their displacements, velocities, and accelerations for given driving motions. Kinematic analysis supports mechanism design, validation, and toleranc-ing. It helps designers detect part interference, establish part clearances, identify functional surfaces, and find jamming and lock-up configurations. It complements dynamical simulation, which determines the actual behavior for specific driving motions, and provides the relevant contact information.

Kinematic analysis of lower pairs is straightforward because there are only six types of surface contact, each of which constrains the touching parts in a simple, known manner. For example, a pin joint constrains two parts to rotate around a common axis. Kinematic analysis of higher pairs is much harder because there are infinitely many types of contact, each of which constrains the touching parts in a different, generally complex way.

Previous mechanism analysis programs perform limited kinematic analysis of higher pairs. Multibody simulators [11, 16] require the user to specify the conditions under which the higher pair contacts occur and the motion relations that they impose. Cam analysis programs [4] handle a single, permanent contact without interference. Gear train analysis programs model intermittent contacts between meshed gear teeth as permanent contacts between rotating cylinders. Other programs [5, 19] analyze planar curves in permanent contact. Rigid body simulators [2, 6, 10] interleave local kinematic analysis with dynamical simulation. They enforce contact constraints at each simulation step by computing contact forces based on the shapes of the touching parts. They restrict surface contacts and simultaneous contacts, and handle only convex or polyhedral parts.

We automate the kinematic analysis of 2.5D higher pairs with two degree of freedom, including pairs with intermittent, simultaneous, and degenerate contacts. This class contains 90% of 2.5D pairs and 80% of all higher pairs according to our survey of 2500 mechanisms from an engineering encyclopedia [13]. The program, called HIPAIR, derives the contact constraints imposed by the interacting features of the pair, and the configurations at which

contact transitions occur. It simulates driving motions and plots the part displacements, velocities, and accelerations. We have tested the program on about 100 examples, including non-standard gears, escapements, ratchets, latches, and indexing pairs. It produces correct results for every pair to within a user-defined tolerance. It takes five seconds to analyze pairs of average shape complexity and one minute for pairs with several thousand contacts.

We perform kinematic analysis within the configuration space representation, which provides a concise, yet complete description of rigid body interaction. The configuration space of a kinematic pair is the space of configurations (positions and orientations) of its parts. It decomposes into blocked space where the parts overlap and into free space where they do not overlap. Only configurations in free space are physically realizable. The boundary between free and blocked space contains the configurations where the parts touch. We partition free space into an adjacency graph of regions. Each region is defined by equality and inequality constraints that represent possible feature contacts. Contact changes occur when the configuration shifts between regions.

The HIPAIR program computes configuration spaces for 2.5D higher pairs with two degrees of freedom. It generates the contact constraints for every pair of contacts, intersects them, and partitions the resulting space into regions. It simulates driving motions by tracing the configuration space paths that the parts traverse. It tests for part interference along user-specified motion paths by verifying that they lie entirely in free space, which guarantees that the parts do not interfere with each other.

We build upon previous research in robot motion planning [14] and in artificial intelligence. The robotics research provides configuration space computation algorithms for simple shapes, such as spheres and polyhedra. For example, Lozano-Pérez [15] and Brost [3] treat a polygonal body with three degrees of freedom amidst polygonal obstacles and Donald [8] treats a polyhedral body with six degrees of freedom amidst polyhedral obstacles. The artificial intelligence research shares our configuration space approach to kinematic analysis of mechanisms. For example, Faltings [9] treats planar, two degree of freedom pairs whose boundaries are line segments and arcs. We extend configuration space computation to general 2.5D shapes, including degenerate cases such as parallel lines and concentric circles.

The HIPAIR program is part of a kinematic analysis and simulation program for multipart mechanisms. The program generates the configuration space of a mechanism comprised of linkages and of higher pairs. It decomposes the mechanism into linkages and higher pair subassemblies, computes subassembly configuration spaces, and composes the results. It computes motion paths by simulating driving motions and external forces. We describe configuration space computation [13] and kinematic simulation [21] in previous publications.

2 Kinematic analysis of a three-finger cam

We illustrate HIPAIR on a simple example: a rotating three-finger cam that drives a reciprocating follower (Figure 1a). A full clockwise rotation of the cam drives the follower back and forth three times with short dwell periods. The cam fingers alternately push the upper vertical segment of the follower right and the lower vertical segment left. Turning the cam counterclockwise causes it to block when one of its fingers hits a horizontal segment of the follower.

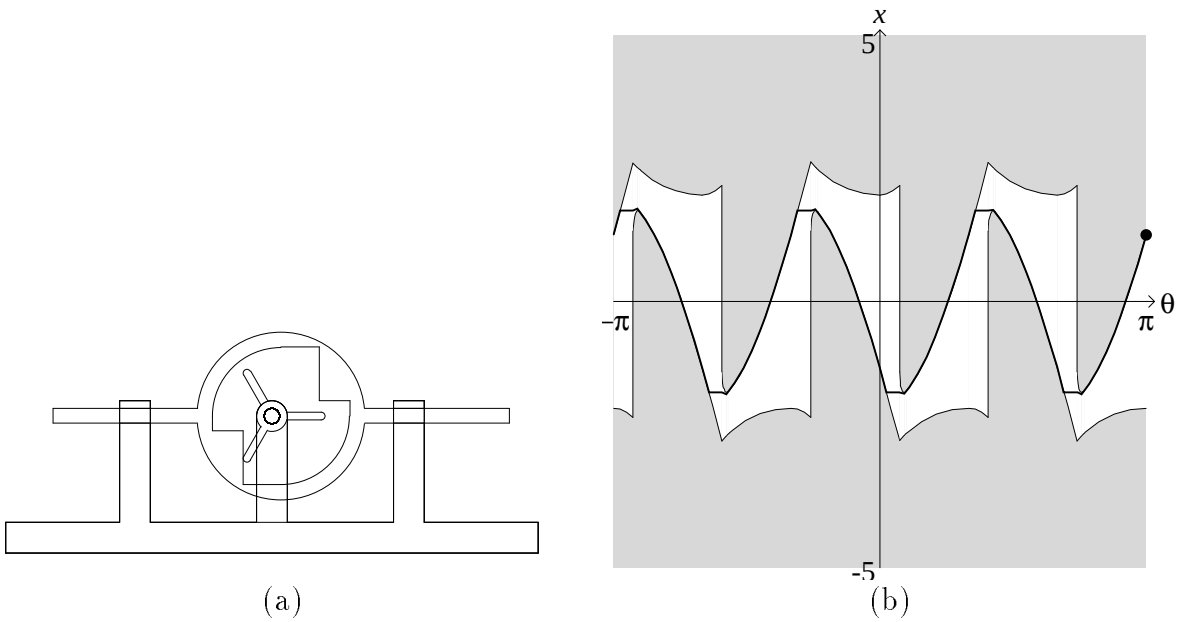
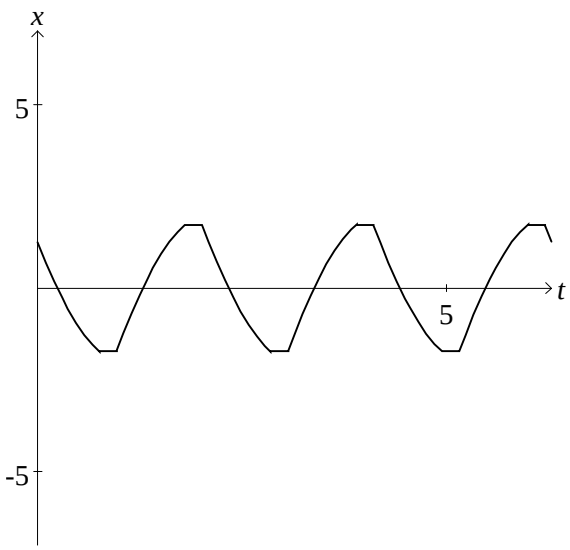


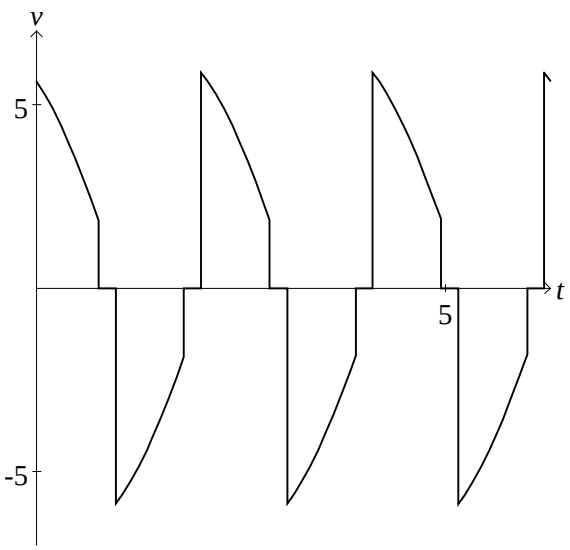
Figure 1: (a) The three-finger cam; (b) Configuration space with θ the cam orientation and x the follower displacement. Shading indicates blocked configurations. The thick line is the motion path. The filled circle marks the displayed configuration $(\pi, 1.25)$.

HIPAIR derives the behavior of the pair by computing its configuration space and simulating the rotation of the cam (Figure 1b). The configuration of the pair follows the thick line from right to left as the cam rotates clockwise, starting at configuration $(\pi, 1.25)$. The horizontal segments represent the dwell periods when the cam does not touch the follower. The rising and falling segments represent the periods when the cam pushes the follower right and left. A cam finger breaks contact with one side of the follower at the start of each dwell period. The next finger makes contact with the other side of the cam at the end of the dwell period. Figure 2 shows the displacement, velocity, and acceleration plots for this motion. HIPAIR computes the configuration space and the plots in two seconds on a workstation.

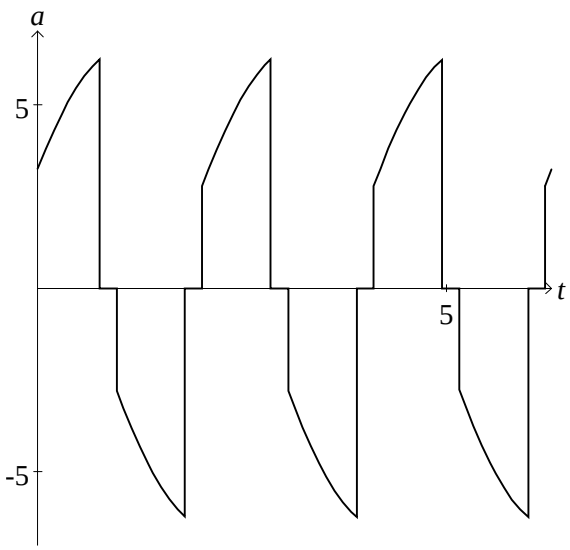
Configuration space computation supports tolerancing and design. For example, we test the sensitivity of the pair to variations in the length and angular spacing of the cam fingers



displacement



velocity



acceleration

Figure 2: The displacement, velocity, and acceleration of the follower when the cam rotates clockwise at $1 rad/s$.

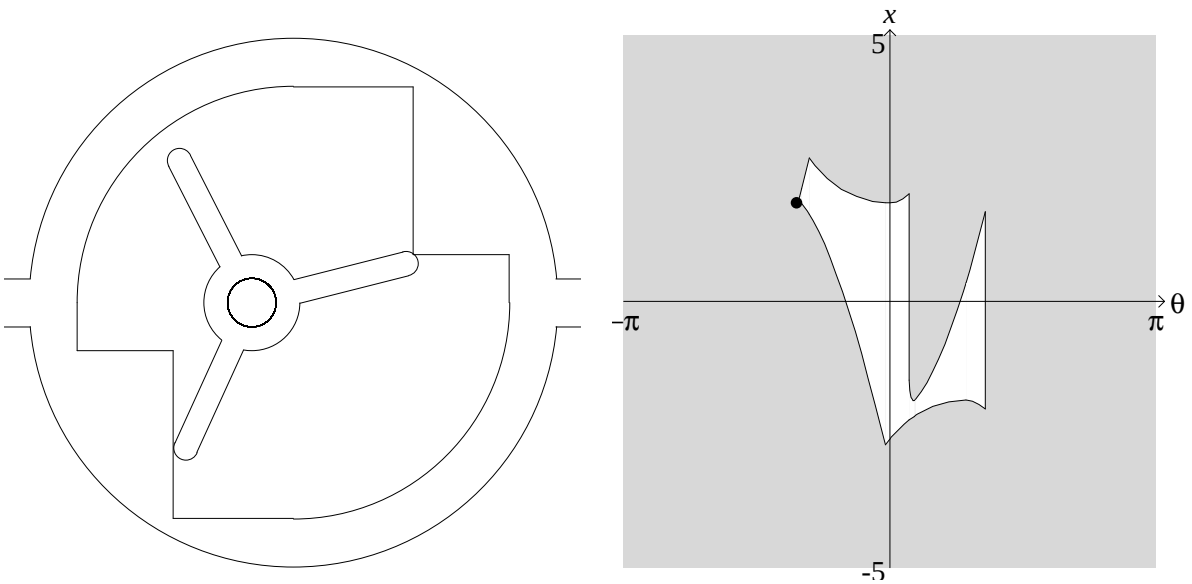


Figure 3: (a) Jammed cam; (b) Configuration space.

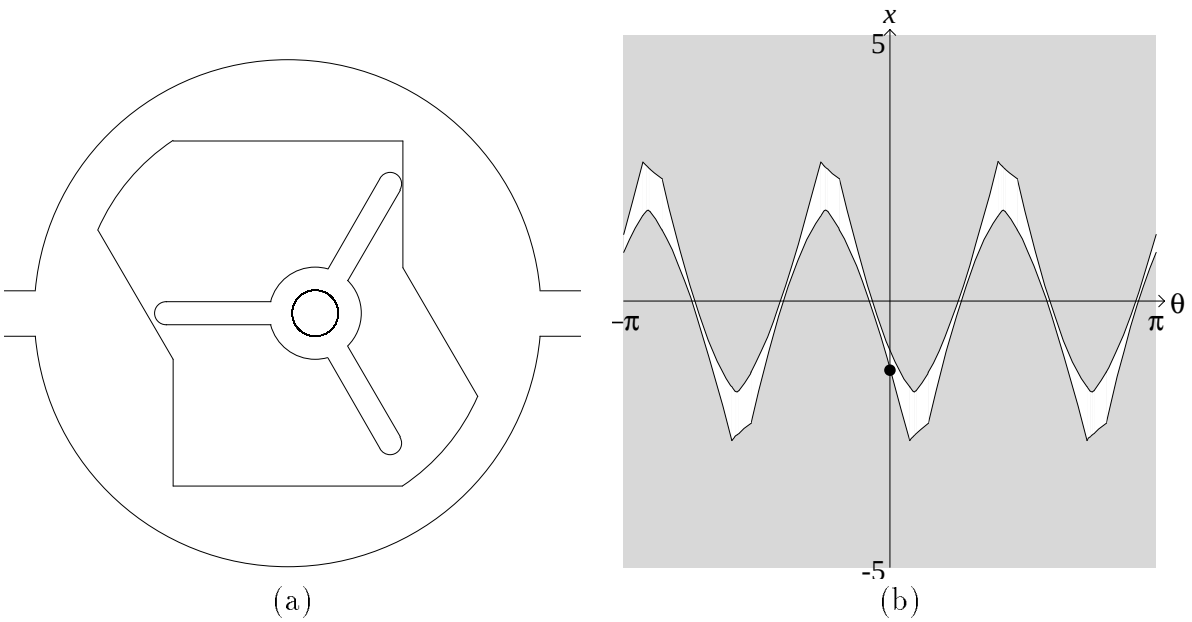


Figure 4: (a) Bidirectional cam; (b) Configuration space.

by computing configuration spaces for several variations. We find that a simultaneous 2.5% increase in length and 8° deviation in one finger produce jamming. The free space shrinks into a closed region that permits only partial rotation (Figure 3). This situation, which involves simultaneous variations in two independent parameters, cannot be detected with traditional, one-parameter tolerancing techniques. Suppose now we want to make the cam pair bidirectional. We observe that the vertical segments of the original configuration space, which represent contacts between the cam fingers and the horizontal segments of the follower, prevent clockwise rotation (Figure 1b). We obtain a symmetrical configuration space by replacing the horizontal segments with slanted segments (Figure 4).

3 The HIPAIR program

We now describe the HIPAIR program in detail. HIPAIR automates the kinematic analysis of 2.5D higher pairs with two degrees of freedom. Each part consists of one or more slices parallel to a reference plane. Each slice is formed by extruding a closed boundary parallel to the reference plane. The boundary consists of line segments, circular arcs, and parametric curves. Each part has one degree of freedom, meaning that its configuration lies on a curve in configuration space. The most common cases are translation or rotation relative to a fixed spatial axis.

Figure 5 shows the analysis algorithm. The inputs are the part shapes, degrees of freedom, and initial configuration of the parts, a tolerance, and a driving motion. The shapes are specified in boundary representation with respect to a local coordinate frames. The degrees of freedom are the motion parameters and the motion paths of the local coordinate frames. In the fixed axis case, the motion path is the axis of motion and the motion parameter is the displacement along or rotation around that axis. In the general case, the motion parameter is a variable s that ranges over $[0, 1]$ and the motion path is a parametric curve $(x(s), y(s), \theta(s))$. The driving motion is a mapping from time to the motion parameter of one part. The program outputs are the configuration space and the displacement, velocity, and acceleration plots. The results are correct to within the tolerance.

3.1 Contact curves

Given two parts, HIPAIR enumerates the pairs of interacting features and computes their contact curves. It formulates contact equations that express the condition that the features touch without overlap. The motion parameters and the feature shapes determine their functional form. HIPAIR solves the contact equations to obtain the contact curves. The solutions define the motion parameter of one part as a function of the motion parameter of the other part over a range of values.

We classify a pair as planar when the parts move in a shared reference plane, as do the

Input: part shapes, degrees of freedom, initial configuration, tolerance, driving motion

1. formulate contact curves
2. compute contact regions
3. simulate driving motion

Output: configuration space and displacement, velocity, and acceleration plots

Figure 5: Kinematic analysis algorithm.

three-finger cam and the follower. Planar contacts between two slices are uniform along the axis orthogonal to the reference plane. HIPAIR derives the contact curves by projecting each pair of slices onto the reference plane, computing the planar contacts between the pair boundaries, and combining the results. For example, the three-finger cam has two slices, the propeller and a drive shaft (not shown), and the follower has one slice. The propeller interacts with the follower, but the drive shaft does not, so the only contacts are the planar contacts between the profiles of the propeller and the follower.

We classify a pair as orthogonal when the orthogonal axis of one part lies in the reference plane of the other. For example, Figure 6 shows an orthogonal gear pair. The driver (top) rotates around an axis normal to the page with motion parameter θ and the driven gear (bottom) rotates around a vertical axis parallel to the page with motion parameter ω. The configuration space consists of a narrow channel with wavy sides, which implies coupled rotation with slight backlash and chatter.

HIPAIR computes closed-form solutions for planar and orthogonal contacts between points, line segments, and circular arcs that move along fixed axes. It computes numerical solutions for all other contacts. We prefer closed-form solutions when they are available even though the numerical approach always applies in principle. The numerical approach cannot handle degenerate contact equations or multiple contacts, such as an s-shape that touchs a vertical wall. The user must specify an initial contact configuration. Solving contact equations numerically takes roughly 100 times longer than discretizing a closed-form solution. On the other hand, closed-form solutions are generally difficult or impossible to derive. Previous work [14] provides the generic functional forms for line and point contacts, but not the other contacts, the degenerate cases, or the ranges. We obtain the best of both approaches by solving the simple, common cases in closed-form and solving the rest numerically.

We have compiled a table of closed-form contact curves indexed by the shapes and motion types of the features. For planar contacts, the 4 motion types (combinations of rotation and translation) and 9 contact types (combinations of points, lines, and arcs) define 36 contact curves. We reduce the 36 cases to 8 cases by exploiting symmetry and by treating

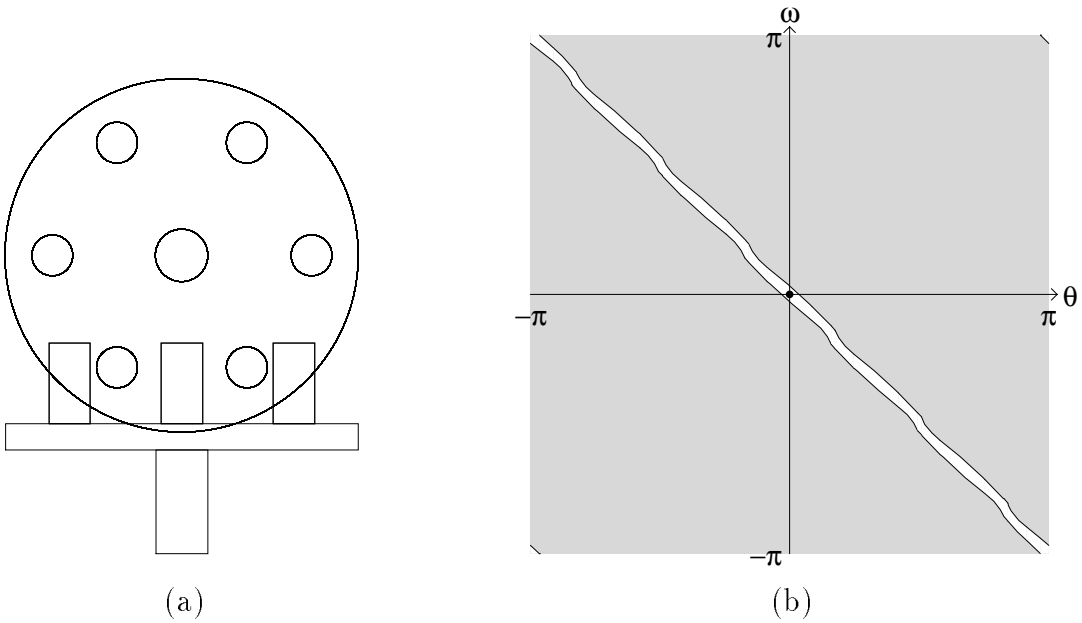


Figure 6: (a) Orthogonal gears; (b) Configuration space.

points as arcs of radius zero. Seven cases represent nonlinear contact curves: translating line/translating arc, translating arc/translating arc, rotating arc/translating arc, rotating arc/translating line, rotating line/translating arc, rotating arc/rotating arc, and rotating arc/rotating line. The eight case represents linear contact curves: translating line/translating line and degenerate cases such as concentric rotating arcs. For orthogonal contacts, we obtain 5 nonlinear curves for rotation/rotation, which produces line, arc, planar face, and cylindrical face contacts. We obtain linear curves for translation/translation and translation/rotation motion types, which produces planar and cylindrical contacts.

For example, the planar contact equation for a rotating arc and a translating line relates the orientation, θ, of the arc frame to the horizontal offset, x, of the line frame (Figure 7). It states that the line is tangent to the arc, that is that the distance from the arc center to the line equals the arc radius. The equation is

$$ax_o + by_o - ax + c - r = 0 \tag{1}$$

with $[a, b]$ a unit vector normal to the line and pointing into its interior, $o = (x_o, y_o)$ the arc center, and $c = -ax_l - b(y_l + y)$ a constant determined by the line. We relate θ to x by substituting $x_o = x_f + d\cos(\theta+\alpha)$ and $y_o = y_f + d\sin(\theta+\alpha)$ into the equation, where (x_f, y_f) are the global coordinates of F's origin. The closed-form solution is $x = k_1\cos(\theta + k_2) + k_3$ where the k's depend on l, m, o, and r. The range of the solution is those x values for which the contact point lies on the arc (between p and q) and on the line (between l and

m). It is empty when the line segment lies entirely above or below the region spanned by the arc. Otherwise, it contains one or two intervals whose endpoints depend on the feature parameters.

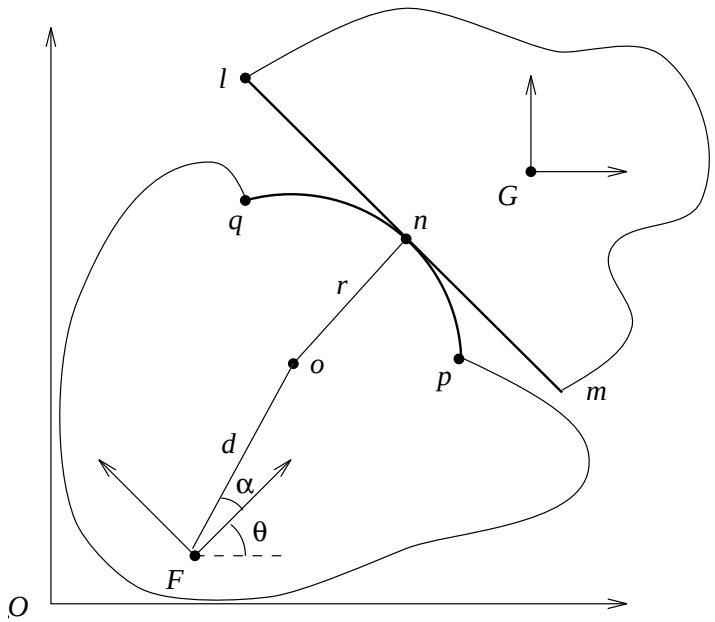


Figure 7: Contact configuration for rotating arc pq and translating line segment lm with F and G the local coordinate frames of the rotating and translating parts, n the contact point, d the distance from F to the center, o, of the arc, and O the global coordinate frame.

HIPAIR handles all other contacts by standard numerical methods. The equations for planar, parametric features that move along fixed axes are

$$\begin{aligned} \mathrm{Trans}(F) \circ \mathrm{Rot}(\theta)\, p(u) &= \mathrm{Trans}(G) \circ \mathrm{Rot}(\omega)\, q(v) \\ \mathrm{Rot}(\theta)\, p'(u) &= k\, \mathrm{Rot}(\omega)\, q'(v) \end{aligned}$$

with $p(u)$ and $q(v)$ the parametric curves, F and G the positions of the part reference frames, θ and ω the orientations of the part reference frames, k a variable, Rot the rotation operator, and Trans the translation operator (Figure 8). The first equation states that the features touch and the second states that they are tangent at the contact point. One of the quantities x_f, y_f, θ is the motion parameter, s, of the first feature, while the other two are fixed. Likewise, one of the quantities x_g, y_g, ω is the motion parameter, t, of the second feature, while the other two are fixed. We express the contact equations as four scalar equations in u, v, k, s, and t. We eliminate k from the last two equations and solve the resulting system for t as a function of s. We obtain the same equations for general motions, except that the motion parameters range over $[0, 1]$ and that the coordinates are

parametric functions $x_f(s), y_f(s), \theta(s)$ and $x_g(t), y_g(t), \omega(t)$. HIPAIR formulates the contact equations with the BOUNDER computer algebra package [20] and solves them with the AUTO continuation package [7].

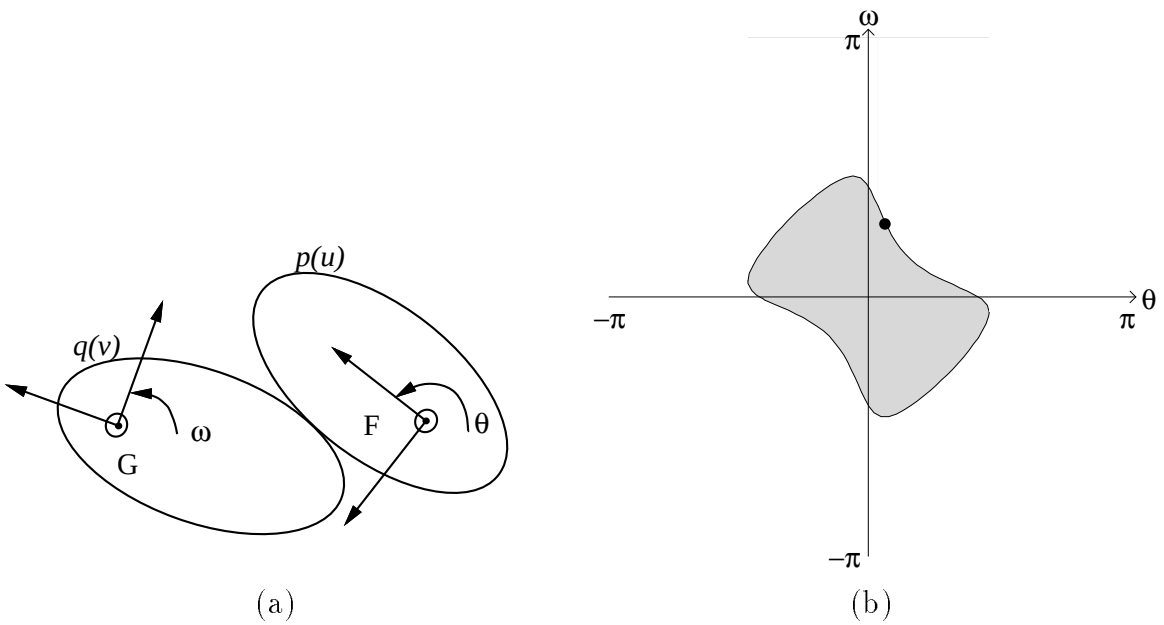


Figure 8: (a) Parametric ellipses pair; (b) Configuration space.

3.2 Contact regions

HIPAIR computes the contact regions of a pair from its contact curves. The curves partition the configuration space into connected components. The component that contains the initial configuration forms the reachable free space, while the others form the blocked space. The component boundaries are closed chains of segments whose endpoints are contact curve intersection points. HIPAIR splits the contact curves at their intersection points, groups the resulting segments into regions, and links the regions into an adjacency graph. It retrieves the reachable regions by depth first search of the adjacency graph starting from the region that contains the initial configuration. Figure 9 illustrates these operations.

The algorithm handles all curve shapes, including vertical lines and cusps, and all configurations, including identical curves, overlapping curves, and groups of curves that intersect at a single point. It produces correct results whenever the minimum distance between contact curve intersection points is greater than the input tolerance. The program cannot handle one-dimensional contact regions, such as zero width channels, which correspond to parts mating with no interplay. The user must pick dimensions that allow some interplay.

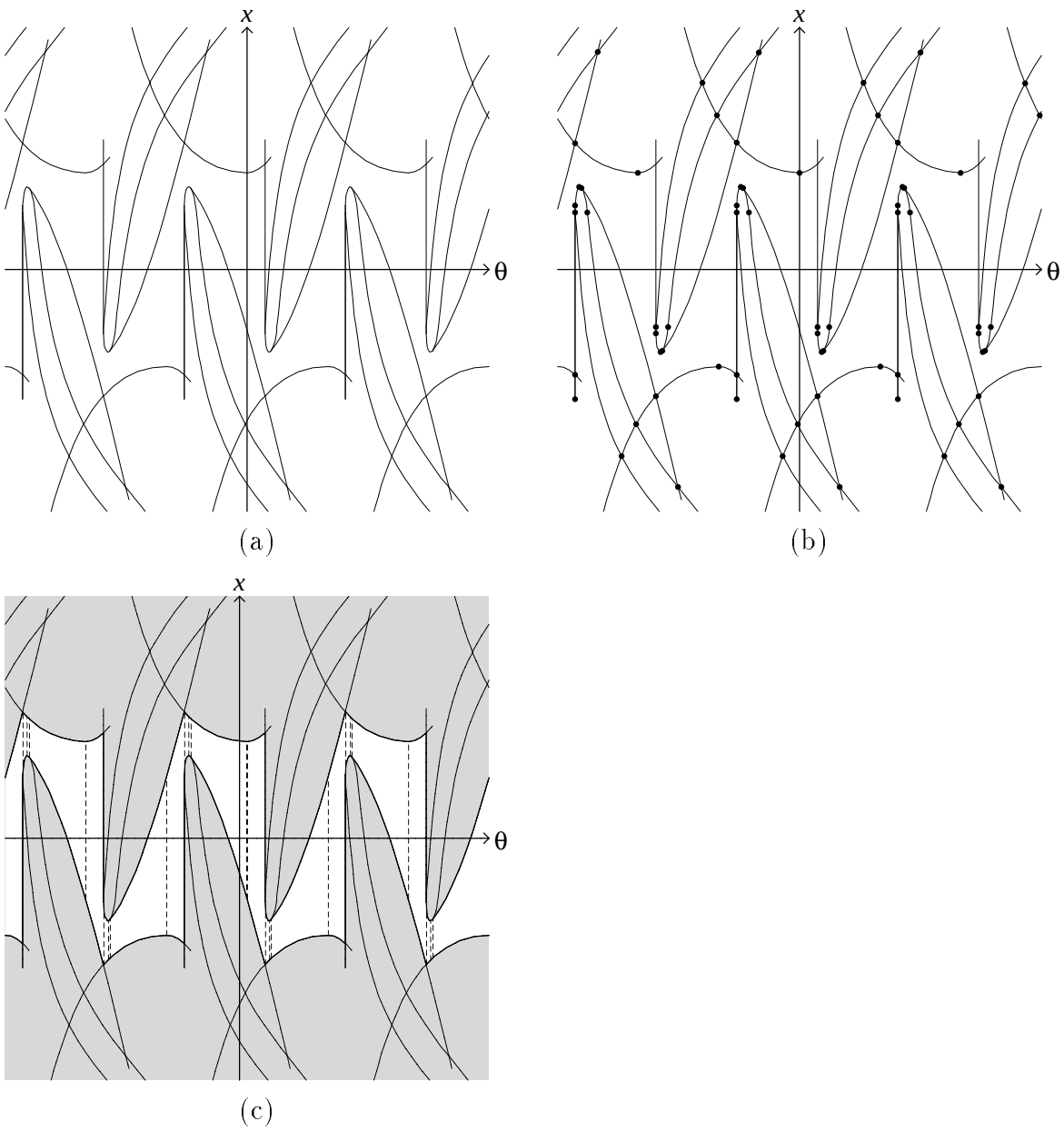


Figure 9: Computing the configuration space of the cam/follower: (a) contact curves, (b) intersection points, (c) reachable regions.

HIPAIR finds contact curve intersections by discretizing the closed-form curves, intersecting the discretizations, and refining the approximate roots using the secant method. Given a curve $f(x)$ defined on an interval $[a, b]$, HIPAIR computes a sequence of points $(a, a_1, \ldots, a_n, b)$ for which the linear interpolation agrees with f to within the tolerance. It finds the value $x = x_0$ that maximizes the error between $f(x)$ and its linear approximation by golden section minimization [18]. If the maximal error is less than the tolerance, the discretization is $[a, b]$. Otherwise, HIPAIR recursively discretize $[a, x_0]$ and $[x_0, b]$ and concatenates the results. HIPAIR splits the discretized contact curves into non-intersecting, monotone segments by a vertical line sweep. We have extended the standard algorithm [17] to degeneracies such as overlapping line segments and multiple segments meeting at a point. We minimize spurious intersections by rounding points to the input tolerance. HIPAIR performs a second line sweep to partition configuration space into regions and to compute the region adjacency graph.

3.3 Simulation and interference detection

Once the configuration space of a pair is computed, simulation and interference detection are straightforward and efficient. HIPAIR simulates a driving motion by computing the configuration space paths of the parts. It starts from the region that contains the initial configuration, computes the piecewise linear segment of the path lying in that region, finds the next region that the path enters, and repeats the process. It computes the segments by propagating the linearized input motion through the contact constraints within the regions. The full algorithm and several examples appear in a previous paper [21]. HIPAIR computes the velocity and acceleration plots from the motion path by the finite difference method.

HIPAIR checks for part interference along a user-specified motion path by testing if the path lies wholly in free space. Given the configuration space and the motion path of the parts, it discretizes the motion path and tests each point for membership in the free space regions. Points that do not belong to any free space region indicate part overlap configurations. The configuration space boundary closest to the offending point identifies the overlapping features and the depth of overlap.

4 Evaluation

We have tested HIPAIR on over 100 parametric variations of 25 pairs. The test cases represent the wide range of pairs in Artobolevsky's encyclopedia of mechanisms [1]. They include planar and spatial pairs of many functional types, including gears, cams, and locks. They range from tens to hundreds of features and from tens to thousands of contacts. HIPAIR computes the configuration space, the simulation, and the plots for every example in under a minute on a workstation. (It is written in Common Lisp, except for the golden section

routine, which is written in C.) To give the reader a better idea of HIPAIR's scope, we augment the examples of the previous sections with two more complex examples.

Figure 10a shows a dwell gear pair. The driver (right) is a concentric arc segment mounted on a gear sector. The driven part (left) is a star-shaped disk mounted on a gear wheel. As the driver rotates clockwise, the gear sector engages and turns the gear wheel. When the sector disengages, the concentric arc segment engages one of the ten concave surfaces of the star-shaped disk, thereby locking it in a dwell position. Each rotation of the driver advances the driven part by 1/10th of a rotation. Figure 10b shows the configuration space. The horizontal channels represent the dwell period. The slanted channels represent the gear engagements; the rough edges reflect chatter and slight variations in the motion transmission ratio. The channel width shows the amount of backlash.

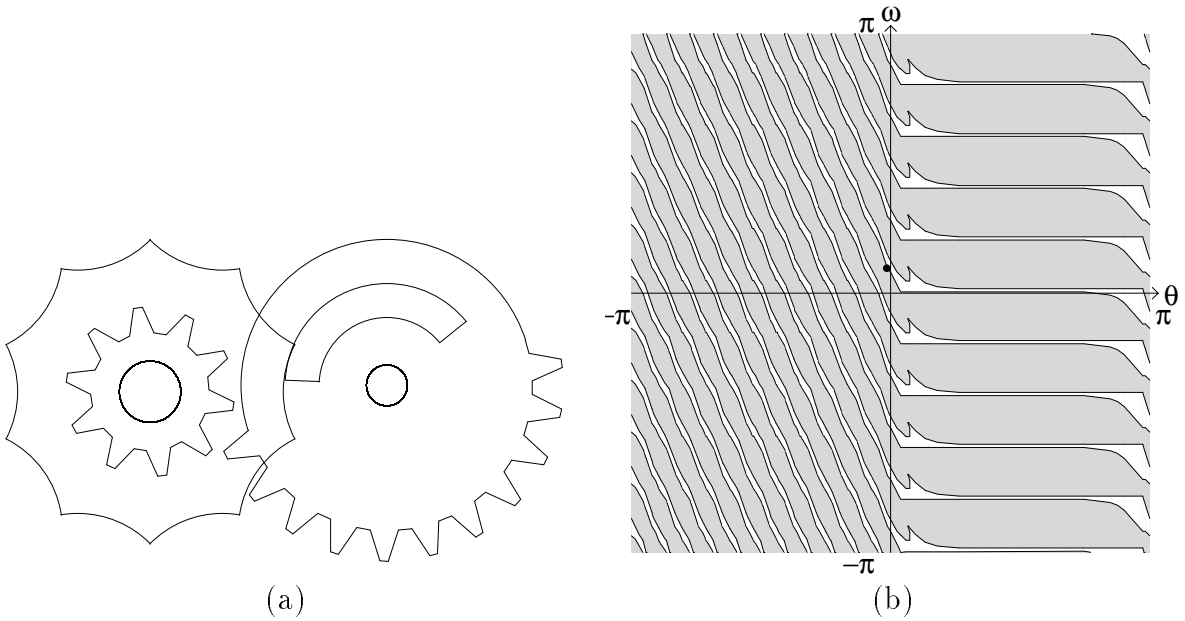


Figure 10: (a) Dwell gear pair; (b) Configuration space. The orientations of the driver (right) and the driven part (left) are θ and ω.

Figure 11a shows an escapement: a wheel and an anchor rotationally mounted on a fixed frame. Attached to the anchor is a pendulum (not shown) that oscillates with a constant period. The wheel rotates counterclockwise due to a torque acting on its hub. As the anchor oscillates, the pallets alternately engage teeth on the left and right sides of the wheel. The wheel advances one notch each oscillation. It rotates at a constant velocity determined by the period of the anchor. Figure 11b shows a portion of the configuration space. The upper and lower boundaries correspond to contacts of the right and left anchor arms with the teeth. The interleaved boundaries prevent the free rotation of the wheel.

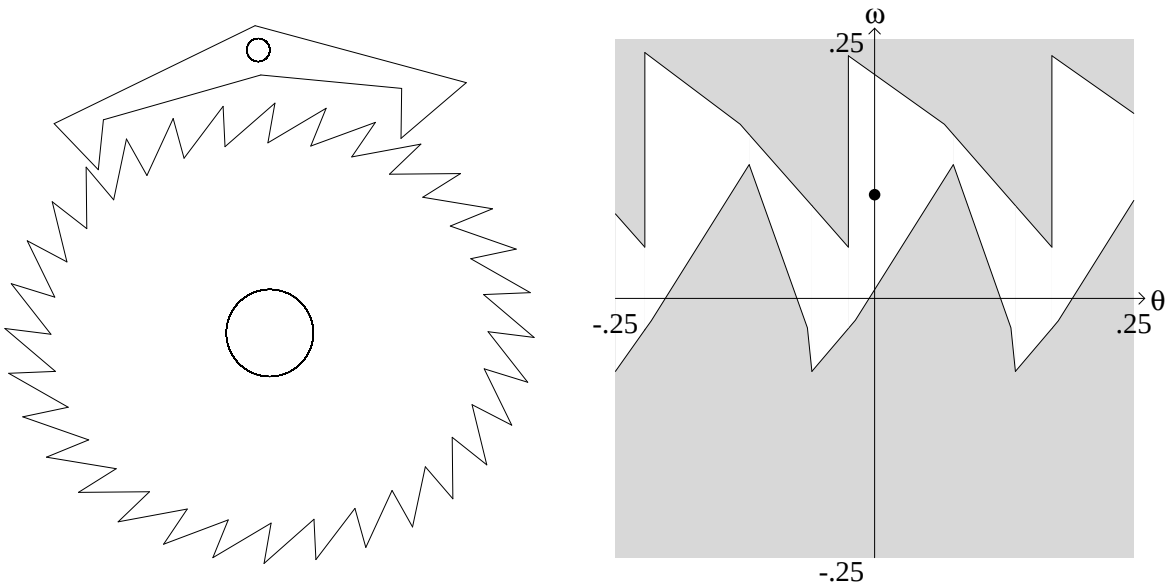


Figure 11: (a) Escapement; (b) Configuration space detail. The orientations of the anchor and the wheel are θ and ω.

We surveyed 2500 mechanisms in Artobolevsky's encyclopedia [1] to determine the coverage of our program. We found that 90% of the pairs in those mechanisms are higher pairs with two degrees of freedom and that 80% are 2.5D as well. Significant exceptions include 3D cams, which are not 2.5D, and spherical joints, which have three degrees of freedom. Details of the survey and examples of pairs not covered appear in a previous paper [13].

We can improve HIPAIR by making it faster and by extending its coverage. Rewriting HIPAIR in C would speed it up by a factor of 10. Adding closed-form contact curves for additional planar contacts, such as elliptical cams and gear involutes, or for spatial contacts, such as screw pairs and helical cams, would reduce numerical computation and would improve both speed and accuracy. Adding 3D surface contacts requires defining a solid model representation of the surfaces [12] and formulating and numerically solving spatial contact equations similar to the planar ones [4]. These extensions are independent of the region computation and motion simulation algorithms. Extending HIPAIR to pairs with three degrees of freedom is much harder, since we must compute and intersect contact surfaces, rather than contact curves. Brost [3] presents such an algorithm for two planar polygons with three degrees of freedom and without degeneracies.

5 Conclusion

In this paper, we describe a program, called HIPAIR, that automates the kinematic analysis of 2.5D higher pairs with two degrees of freedom. It handles intermittent, simultaneous, and degenerate contacts. It computes configuration spaces, derives contact constraints and contact transitions, and simulates driving motions. It analyzes pairs of average shape complexity in under five seconds.

HIPAIR supports interactive mechanism design and optimization. Configuration space computation provides a behavioral description that is more detailed and more complete than previously available kinematic analysis software. It frees engineers from manual derivation of contact constraints and contact transitions. It provides automated part interference detection, validation of desired behaviors and quick identification of design flaws. Our efficient implementation supports interactive simulation, dynamical analysis, and parametric design. It helps engineers identify unexpected failure modes in familiar pairs, produce interesting variations of familiar designs, and manipulate subtle geometric interactions.

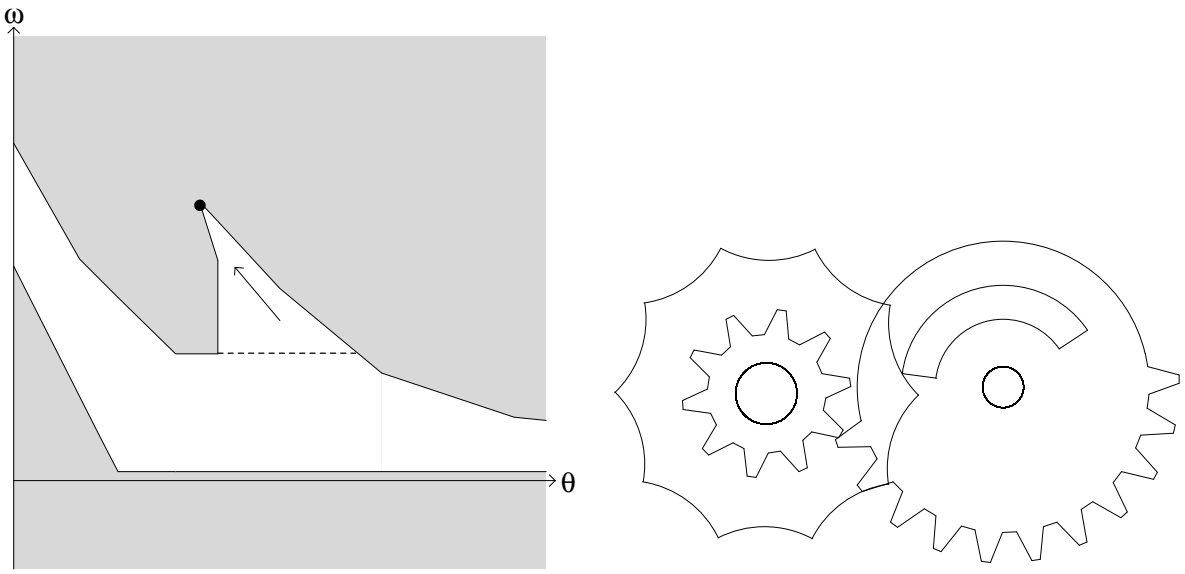


Figure 12: (a) Detail of the dwell gear configuration space. The jamming region lies above the dashed line. The arrow indicates a jamming motion. (b) Jammed configuration.

We have demonstrated these capabilities on several examples. The dwell gear pair provides a final example. The pair exhibits the desired behavior in the initial configuration shown in Figure 10. Rotating the driver in either direction causes the driven part to rotate in the opposite direction with dwell periods in between. But a close examination of the con-

figuration space reveals that starting from configurations above the dotted horizontal line and rotating the driver clockwise causes the pair to jam (Figure 12). Modifying the locking arc or the first tooth of the driver eliminates the problem.

References

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

[2] Baraff, D. *Dynamic Simulation of Non-Penetrating Rigid Bodies*. PhD thesis, Cornell University, 1992.

[3] Brost, R. C. Computing metric and topological properties of configuration-space obstacles. in: *Proceedings IEEE Conference on Robotics and Automation*, pages 170–176, 1989.

[4] Chakroborty, J. and Dhande, S. G. *Kinematics and Geometry of Planar and Spatial Cam Mechanisms*. (Wiley, New Dehli, India, 1979).

[5] Chen, S.-C., Kinzel, G. L., and Kuhlmann, D. J. A numerical method for the kinematic analysis of planar higher pairs in rolling contact. *Mechanism and Machine Theory* **20** (1985) 565–575.

[6] Cremer, J. F. and Stewart, A. J. The architecture of *newton,* a general-purpose dynamics simulator. in: *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, pages 1806–1811, 1989.

[7] Doedel, E. AUTO 86 user manual: software for continuation and bifurcation problems in ordinary differential equations. Technical report, Princeton University, Feb. 1986.

[8] Donald, B. R. A geometric approach to error detection and recovery for robot motion planning with uncertainty. *Artificial Intelligence* **37** (1988) 223–271.

[9] Faltings, B. Qualitative kinematics in mechanisms. *Artificial Intelligence* **44** (1990) 89–120.

[10] Hahn, J. K. Realistic animation of rigid bodies. *Computer Graphics* **22** (1988) 299–308.

[11] Haug, E. (Ed.). *Computer Aided Analysis and Optimization of Mechanical System Dynamics*. (Springer-Verlag, 1984).

[12] Hoffmann, C. M. *Geometric & Solid Modeling*. (Morgan Kaufmann, 1989).

[13] Joskowicz, L. and Sacks, E. Computational kinematics. *Artificial Intelligence* **51** (1991) 381–416.

[14] Latombe, J.-C. *Robot Motion Planning.* (Kluwer Academic Publishers, 1991).

[15] Lozano-Pérez, T. Spatial planning: A configuration space approach. in: *IEEE Transactions on Computers*, volume C-32. IEEE Press, 1983.

[16] Potter, C. Mechanism analysis moves with the times. *Computer Graphics World* **15** (1992).

[17] Preparata, F. P. and Shamos, M. I. *Computational Geometry.* (Springer-Verlag, New York, 1985).

[18] Press, W. H., Flannery, B. P., Teukolsky, S. A., et al. *Numerical Recipes in C.* (Cambridge University Press, Cambridge, England, 1990).

[19] Reinholtz, C. F., Dhande, S. G., and Sandor, G. N. Kinematic analysis of planar higher pair mechanisms. *Mechanism and Machine Theory* **13** (1978) 619–629.

[20] Sacks, E. Hierarchical reasoning about inequalities. in: *Proceedings of the National Conference on Artificial Intelligence*, 1987. Reprinted in [22].

[21] Sacks, E. and Joskowicz, L. Automated modeling and kinematic simulation of mechanisms. *Computer-Aided Design* **25** (1993) 106–118.

[22] Weld, D. S. and de Kleer, J. (Eds.). *Readings in Qualitative Reasoning about Physical Systems.* (Morgan Kaufman, San Mateo, Ca., 1990).