Purdue University

Purdue e-Pubs

Department of Computer Science Technical Reports

Department of Computer Science

1997

Computer-Aided mechanical Assembly Design Using Configuration Spaces

Leo Joskowicz

Elisha Sacks
Purdue University, eps@cs.purdue.edu

Report Number:
97-001

COMPUTER-AIDED MECHANICAL ASSEMBLY DESIGN USING CONFIGURATION SPACES

Leo Joskowicz
Elisha Sacks

CSD-TR 97-001
January 1997

Computer-Aided Mechanical Assembly Design Using Configuration Spaces

Leo Joskowicz
Institute of Computer Science
The Hebrew University
Jerusalem 91904, Israel
E-mail: josko@cs.huji.ac.il

Elisha Sacks
Computer Science Department
Purdue University
West Lafayette, IN 47907, USA
E-mail: eps@cs.purdue.edu

January 6, 1997

Abstract

We describe a unified approach to computer-aided mechanical assembly design in which all design tasks are performed within a single computational paradigm supported by integrated design software. We have developed a prototype design environment for planar assemblies, called HIPAIR, that automates dynamical simulation and provides novel support for tolerancing and parametric design. We organize design tasks around the fundamental task of contact analysis, which we automate by configuration space computation. Configuration space is a complete, concise, and explicit representation of rigid body interactions and contains the requisite information for design tasks involving contacts. We describe algorithms for dynamical simulation, kinematic tolerancing, and parametric design of planar assemblies based on configuration space computation. HIPAIR allows designers to perform computations that lie outside the scope of previous software and that defy manual analysis. It allows them to visualize assembly function under a range of operating conditions, to find and correct design flaws, and to evaluate the functional effects of part tolerances. It has been tested on hundreds of pairs and on a dozen assemblies. HIPAIR performs at interactive speed on assemblies of ten parts with tens of thousands of contacts.

Submitted to *Computer-Aided Design*, January 1997.

1 Introduction

We describe a unified approach to computer-aided mechanical assembly design in which all design tasks are performed within a single computational paradigm supported by integrated design software. Mechanical assembly design is the task of devising an assembly of parts that performs a function reliably and economically. It is a ubiquitous activity that spans mechanical, electrical, and biomedical engineering. Designers need to devise, analyze, and compare competing design prototypes to produce optimal designs. Computer-aided design reduces design time and improves quality by allowing designers to substitute electronic prototypes for physical prototypes in diverse tasks. Our computational paradigm organizes the design tasks around the fundamental task of contact analysis. Our design software uses a general contact analysis module for planar assemblies to automate dynamical simulation and to provide novel support for tolerancing and for parametric design.

Part contacts are the physical primitives that make mechanical assemblies out of collections of parts. Assemblies perform functions by transforming motions via part contacts. The shapes of the interacting parts impose constraints on their motions. For example, a door rotates about its hinges and meshed gears rotate in unison. Reasoning about contacts lies at the heart of many design tasks because contact constraints largely determine the function of assemblies. Designers compute contact constraints to validate function and to measure performance. They correct design flaws by modifying part contacts. They choose part tolerances based on the variation in assembly function that they produce.

Contact analysis, also called kinematic analysis, determines the relation between the function of an assembly and the shapes and motions of its parts. It is difficult and time-consuming even for experienced designers due to the quantity and complexity of the contact constraints. Designers need to ensure that the intended contacts occur, to derive their constraints, and to guarantee that unintended contacts cannot occur. The difficulty is greatest in assemblies with multiple contacts, meaning that different parts or part features interact at various stages of the work cycle. Manual analysis is error-prone and time-consuming at best and is often infeasible. Multiple contacts pervade modern mechanisms and account for 65% of the 2500 mechanisms in Artobolevsky's encyclopedia [14]. The most common examples are gears, cams, clutches, and ratchets. Designers analyze multiple contacts when testing for part interference, jamming, cam under-cutting, and gear backlash. In kinematic tolerance analysis, designers study part variations that introduce multiple contacts into assemblies whose nominal function has permanent contacts, such as joint play in linkages.

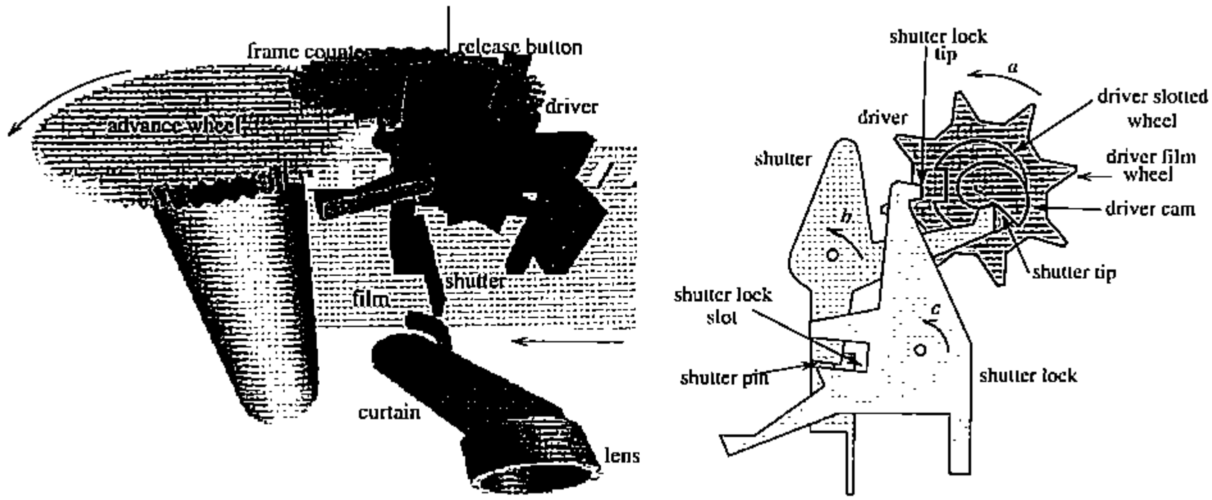
Multiple contact analysis is the limiting factor in mechanism theory and in computer-aided assembly design software. The theory shows how to analyze individual contacts, but not how to derive relations among contacts. This suffices for assemblies with permanent contacts or with simple contact sequences, such as linkages, cams, and involute gears, but is inadequate for assemblies with complex part interactions. Current computer-aided design software reflects this limitation in that it lacks general purpose contact analysis capabilities.

Each package tackles the contact problems in its application area, often placing excessive analysis burdens on designers. Drafting programs provide interactive environments for the design of part shapes, but do not support reasoning about shape interaction. Mechanical systems simulators derive the contact constraints for linkages and specialized pairs, but require users to provide constraints for other types of contacts. Kinematic tolerancing software requires the user to specify the contact constraints as functions of the tolerance parameters.

Previous research in mechanical engineering, graphics, and robotics does not provide general algorithms for contact analysis. Mechanical simulation research [13, 28] focuses on efficient methods of solving the contact constraints of permanent contact assemblies, such as linkages and manipulators. Research in gear design [20] and in cam design [12] addresses narrow classes of contacts. Assembly planning research [7] focuses on the combinatorics of sequencing assembly steps for simple part shapes and motions. Graphics research in physically based modeling [2, 6, 21] provides fast collision detection algorithms for polyhedral objects, but does not address the other aspects of contact analysis in mechanical design. Robotics research [18] studies contact analysis in the context of robot motion planning. The planners use a configuration space representation for the possible contacts between the robot and the obstacles, and search this space for collision-free paths. Most research addresses a single polyhedral robot moving amidst fixed polyhedral obstacles. It does not provide practical algorithms for curved shapes or for multiple moving parts, which are the norm in mechanical assemblies.

We have developed a new unified approach to contact analysis and to computer-aided assembly design based on configuration space computation that is inspired by robotics research in motion planning. We have shown that configuration space is a complete, concise, and explicit representation of rigid body interaction in mechanical assemblies. We have demonstrated that performing contact analysis on an assembly is equivalent to computing its configuration space. The configuration space contains the requisite information for design tasks involving contacts. It models permanent and multiple contacts uniformly. We have developed a prototype design environment for planar assemblies, called HIPAIR, that automates dynamical simulation and provides novel support for tolerancing and parametric design based on a fast, robust configuration space computation program. HIPAIR allows designers to perform computations that lie outside the scope of previous software and that defy manual analysis. It allows them to visualize assembly function under a range of operating conditions, to find and correct design flaws, and to evaluate the functional effects of part tolerances.

In this paper, we present a comprehensive description of our work on integrated assembly design using configuration spaces. We present prior results from diverse publications along with new examples. The paper is organized as follows. The next section describes the configuration space representation. The following section describes the architecture of our HIPAIR design environment and sets the stage for the following four sections that describe the modules for configuration space computation, simulation, tolerancing, and parametric



(a) (b)

Figure 1: Disposable camera: (a) shutter mechanism; (b) top view of driver, shutter, and shutter lock assembly.

design. We conclude with a discussion of future work on extensions to spatial assemblies and to other design tasks.

2 Configuration space

The configuration space of an assembly of parts is a parameter space whose points (tuples of parameter values) specify the spatial configurations (positions and orientations) of the parts. The parameters represent translations and rotations of parts with respect to a fixed global coordinate system. For example, a gear pair has a two-dimensional configuration space in which each gear configuration is specified by a rotation parameter. The configuration space dimension equals the total number of degrees of freedom of the parts in the assembly.

Configuration space partitions into free space where the parts do not touch and into blocked space where some parts overlap. The common boundary, called contact space, contains the configurations where some parts touch without overlap and the rest do not touch. Only free space and contact space are physically realizable. Free space represents the realizable motions of the parts and contact space represents the couplings between their motions induced by contacts.

We illustrate these concepts on the shutter mechanism of a disposable camera, which consists of ten moving parts in a fixed frame (Figure 1a). When the user turns the advance wheel, it moves the film forward by one frame and rotates the driver, which engages the shutter in the shutter lock. Pressing the release button rotates the shutter lock, which

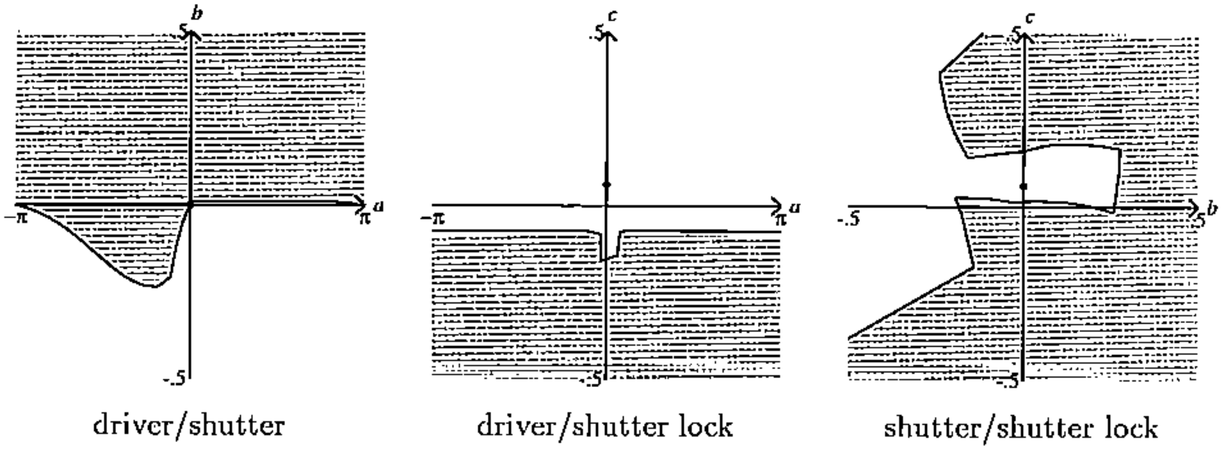


Figure 2: Pairwise configuration spaces of the shutter mechanism. The dot represents the initial configuration shown in the previous figure.

releases the shutter. The shutter spring (not shown) rotates the shutter, which trips the curtain, which rotates away from the lens and exposes the film.

We focus on the loading sequence of the driver, shutter, and shutter lock (Figure 1b). The driver consists of three planar pieces mounted on a shaft: a cam, a slotted wheel, and a film wheel. The shutter consists of two planar pieces, a shutter and a pin, and is spring-loaded counterclockwise. The shutter lock is planar and is spring-loaded clockwise. The driver cam interacts with the shutter tip. The driver slotted wheel interacts with the shutter lock tip. The shutter pin interacts with the shutter lock slot. The shutter is loaded by rotating the driver clockwise, which pushes the shutter back via the contact between the driver cam and the shutter tip. When the shutter pin leaves the shutter lock slot, the shutter lock spring rotates it counterclockwise, simultaneously locking the shutter and the driver. The shutter is locked with the pin pressed against the shutter lock surface below the slot. The driver is locked with the shutter lock tip inside the driver slotted wheel slot.

The pairwise configuration spaces (Figure 2) are two-dimensional because each part has one degree of freedom. The coordinates of the configuration spaces are the orientations a, b, and c of the parts. The shaded regions are the blocked space where the parts overlap. The white regions are the free space. The curves that bound the free and blocked regions are the contact spaces. Each contact space consists of many contact curves that represent possible feature contacts. Contact changes occur at curve endpoints.

The driver/shutter contact space shows the interactions between the driver cam and the shutter tip. The horizontal segment represents contact between the shutter tip and the small circular arc of the driver cam. The leftmost curved segment represents contact with the large circular arc of the driver cam, which pushes the shutter tip back then allows the shutter spring to restore its original position. The driver/shutter lock configuration space shows that the shutter lock tip follows the contour of the driver slotted wheel until it drops into the driver wheel slot. The shutter/shutter lock configuration space shows that the shutter pin moves out of the shutter lock slot and engages on the surface below.

The configuration space of an assembly is compositional: it is determined by the configuration spaces of its pairs of parts [14]. We embed the pairwise configuration spaces in the assembly configuration space by inverse projection. Each pairwise configuration (a, b) maps to the set of configurations $(a, b, \mathbf{x})$ where $\mathbf{x}$ is a parameter vector that varies over all values of the other coordinates. The assembly free space equals the intersection of the embedded pairwise free spaces because an assembly configuration is free when every pair of parts is free. The blocked space equals the union of the embedded pairwise blocked spaces because an assembly configuration is blocked when at least two parts overlap. This property formalizes Reuleaux's observation that mechanisms are chains of pairs of parts [23]. It suggests a divide-and-conquer strategy of computing and composing the configuration spaces of all pairs of interacting parts, which we use in contact analysis and in other design tasks.

Configuration space encodes in a uniform geometric framework the information for reasoning about part contact in all mechanical assemblies. It represents the motion constraints induced by part contacts and the configurations where contacts change. It specifies the space of kinematic functions under all external forces. The functions under specific forces are paths in configuration space that consist of contact and free segments separated by contact change configurations. For example, clockwise rotation of the driver produces a path in the driver/shutter configuration space that follows the contact curves from right to left.

Robotics research shows that configuration space computation can be formulated in terms of computational algebraic geometry [18]. The condition that the parts touch without overlap yields multivariate polynomial inequalities in the configuration space coordinates. The set of configurations that satisfy the contact constraints is the contact space. Computing it takes time polynomial in the geometric complexity of the parts and exponential in the number of degrees of freedom with large constant factors.

Although contact analysis is intractable in theory, it is manageable in practice because mechanical assemblies have characteristics that distinguish them from arbitrary collections of parts. A typical part interacts with one or two parts, not with all parts. Part motions are highly constrained, so typical assemblies have one or two true degrees of freedom (technically, the free space has dimension one or two as a semi-algebraic set). Part geometry and part contacts are either simple, as in linkage joints, or complex but with simple motions, as in gears that rotate around fixed axes. The challenge is to exploit these properties to develop efficient configuration space computation algorithms for realistic assemblies.

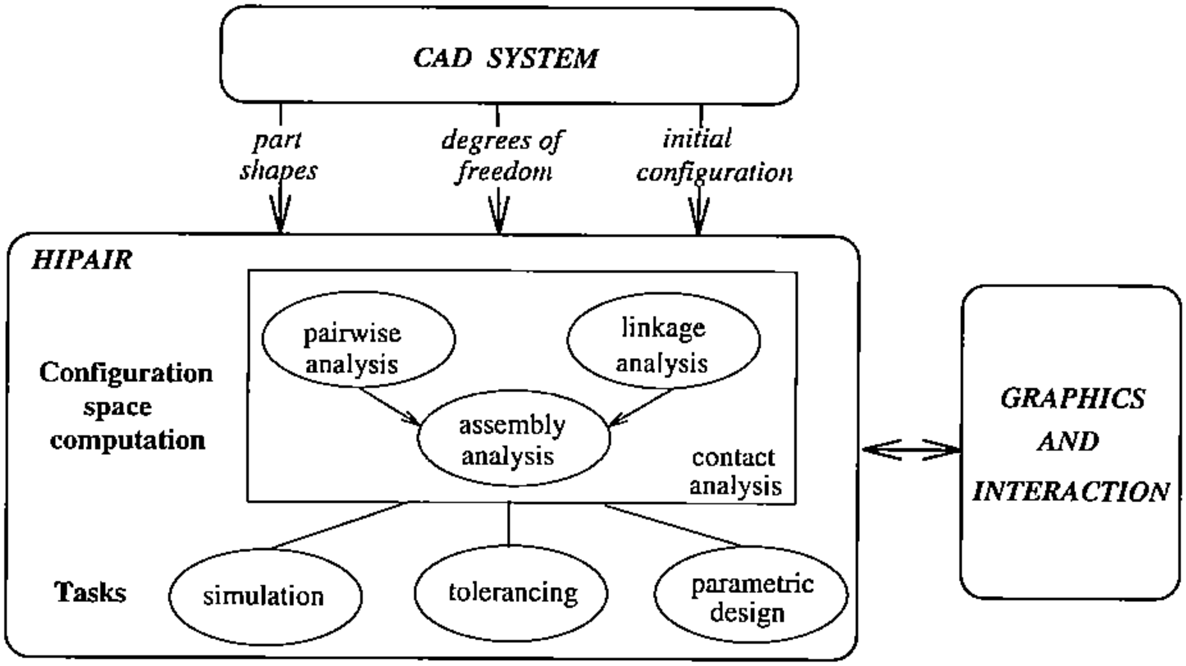


Figure 3: HIPAIR mechanical design environment.

3 HIPAIR architecture

Figure 3 shows the architecture of the HIPAIR environment for assembly design. The architecture embodies our computational paradigm in which all design tasks are organized around the fundamental task of contact analysis. HIPAIR handles assemblies of planar parts. The core module automates contact analysis via configuration space computation. It computes assembly configuration spaces by computing and composing the configuration spaces of their pairs. The task modules use the configuration spaces to support reasoning about contacts. They automate dynamical simulation and provide novel support for tolerancing and parametric design. We use the Microstation CAD software to draft and edit the part shapes, initial configurations, and motion axes. We use a custom graphics and interaction module to display assemblies, configuration spaces, and animations.

We have tested HIPAIR on hundreds of pairs and on a dozen assemblies with up to ten moving parts. We can analyze assemblies with thousands of potential contacts in a few seconds on a workstation. The running times in the paper are for a Silicon Graphics Indigo 2 workstation with 64MB of main memory and a 250 Mhz processor. HIPAIR is written in Allegro Common Lisp, except that the graphics module is written in C. All the figures in the paper are direct output from this module.

4 Configuration space computation

We have developed a fast, robust configuration space computation program for planar assemblies [25, 24]. Planar assemblies cover 90% of all assemblies based on our survey of 2500 mechanisms [14] and on an informal survey of modern mechanisms, such as VCR's and photocopiers. Our strategy is to design algorithms that maximize coverage while maintaining efficiency. We have developed algorithms for pairs, linkages, and general assemblies. In the conclusions, we discuss practical algorithms for spatial assemblies.

Previous research provides practical configuration space computation algorithms for a moving polygon amidst polygonal obstacles. These algorithms do not readily extend to curved bodies because they rely on the special structure of polygonal contact spaces, which are made up of simple, ruled surface patches generated by vertex/edge contacts. Avnaim et al. [1] compute the contact patches by tracing their generating line segments through the range of orientations over which the patches are defined. They compute the singular orientations where changes occur in the number of segments or in the expressions that define their endpoints. They link adjacent patches to obtain the boundary topology. Brost [3] presents a similar algorithm that produces correct configuration spaces on 1599 out of 1600 challenging test cases. He computes contact patches by tracing the boundary curve segments where vertex/vertex contacts occur. He intersects the patches and analyzes the arrangement of intersection edges to compute the contact space. Caine [4] computes contact patches at interactive speeds as part of an interactive design algorithm. He does not compute the patch intersections, so he cannot tell which contacts are adjacent or subsumed. Donald [8] studies configuration space computation for planning the motion of a polyhedral robot with six degrees of freedom. He develops parametric expressions for contact patches and their intersections, but does not compute configuration space partitions.

Pairs

We distinguish between lower and higher pairs. A lower pair consists of two parts joined by a permanent surface contact. There are three types of joints: revolute, prismatic, and sliding. Higher pairs are all other pairs. We divide higher pairs into higher pairs with two degrees of freedom, which account for 80% of the higher pairs in our survey, and into general pairs.

We use table lookup to compute lower pair configuration spaces. The table contains the contact equations for the three types of joints parameterized by the joint attachment points. The first two joints have two equations, which yield one-dimensional contact spaces, while the third joint has one equation, which yields a two-dimensional contact space. Lower pairs have no free spaces or contact changes because the contacts are permanent.

We use planar computational geometry to compute the configuration space of planar pairs with two degrees of freedom [25]. The pairs have two-dimensional configuration spaces: a

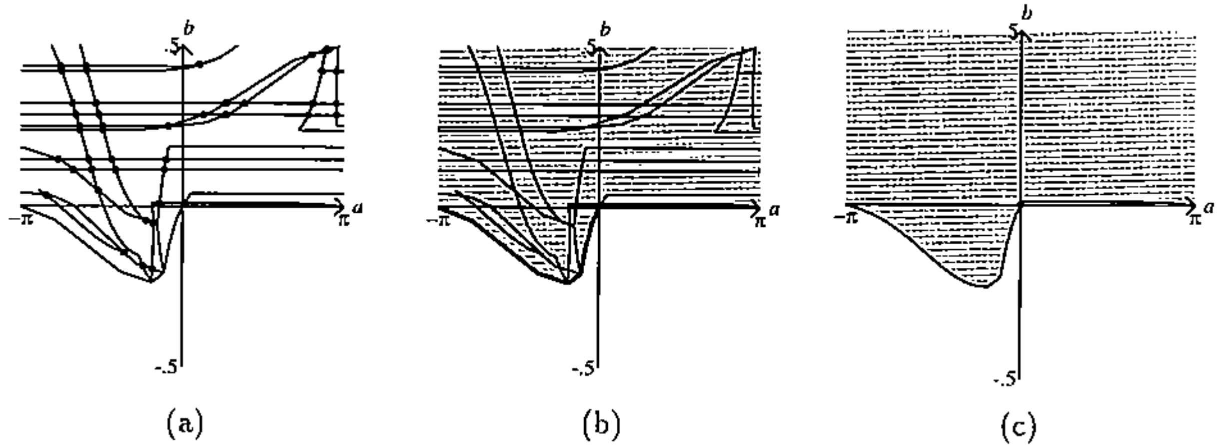


Figure 4: Detail of configuration space computation for the driver/shutter: (a) contact curves and intersection points; (b) connected component and free space; (c) final configuration space.

plane when both parts translate, a cylinder when one rotates and one translates, and a torus when both rotate. The free space and the blocked space are planar regions. The contact space is a collection of planar curve segments. Each segment is part of a contact curve that consists of the configurations where a pair of part features (vertices or curves on the part boundaries) touch. The contact curves partition configuration space into connected components. The component boundaries are sequences of contact curve segments that meet at curve intersection points where multiple feature contacts occur. The component that contains the initial configuration is the reachable portion of the free space.

The program enumerates the feature pairs, generates the contact curves, computes the planar partition with a line sweep, and retrieves the realizable component. The curves come from a table with entries for all combinations of part features and degrees of freedom. The entries are explicit, symbolic expressions for the contact functions. The program generates numerical approximations of the contact curves to within a tolerance by Brent's method [22]. Figure 4 shows the contact curves, intersection points, and components in the driver/shutter configuration space.

We compute the configuration spaces of general planar pairs by dimension reduction [24]. We reduce the six-dimensional configuration space (two translations and one rotation per part) to three dimensions by replacing the global coordinates of the parts with the coordinates of one part relative to the other, which amounts to holding one part fixed. We compute the three-dimensional configuration space by computing 2D configuration space slices along the rotation axis. We partition the rotation axis into intervals of equivalent slices separated by critical slices where the contact structure changes. We use the two-dimensional configuration space program to compute each slice. We use the axis partition to compute the topology

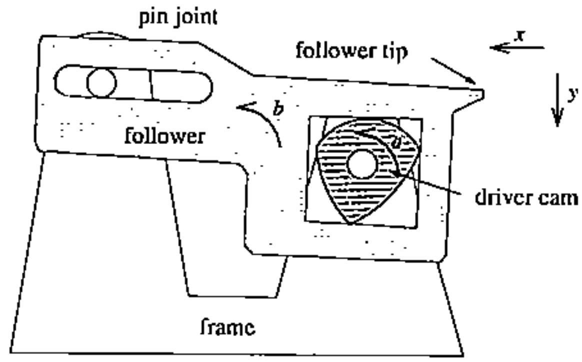


Figure 5: Movie camera film advance. The driver cam orientation is a and the follower configuration is (x, y, b).

of the three-dimensional configuration space, to approximate the contact space geometry, and to answer queries in support of design tasks. The output is topologically correct and accurate to within a specified tolerance.

We illustrate the approach on the film advance of a movie camera (Figure 5). The driver cam rotates about a shaft on the frame, while the enclosing follower is attached to the frame by a pin joint. As the cam rotates clockwise, the follower tip engages the film (not shown), pushes it down one frame, and retracts. Figure 6a shows the $\theta = 0$ slice of the driver/follower configuration space with x, y, and $\theta = a - b$ the relative horizontal, vertical, and orientation coordinates. The contact space is a rectangle that delimits the allowable cam translations at this orientation. The other slices have the same shape, since the cam has constant breadth, but are shifted horizontally and vertically. Figure 6b shows the three-dimensional configuration space. The free space forms a narrow spiral channel bounded by the contact space, which is shown in grey. The blocked space, everything outside the channel, is omitted for clarity. The computation produces 73 slices for a slice separation of 0.1 radians in two seconds.

Linkages

We distinguish linkages from general assemblies. A linkage is a collection of lower pairs. Linkages account for 33% of mechanisms in our survey and for most robots. They have received considerable attention in the mechanical engineering literature and are the primary focus of previous contact analysis research. We discuss linkages briefly because our treatment differs from the standard engineering treatment in form rather than in content.

The configuration space of a linkage consists of a single, algebraic contact surface with no free space because it is the composition of lower pair spaces that have this property. This implies that multiple contacts cannot occur. The contact space is the solution set of the joint

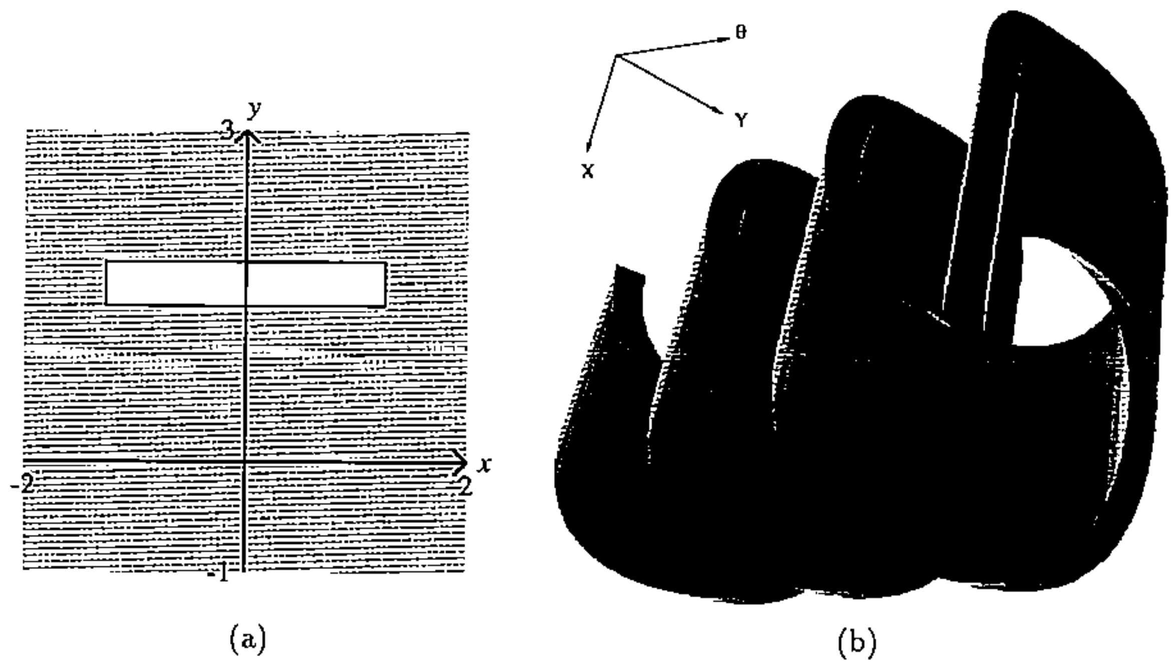


Figure 6: (a) Film advance configuration space slice $\theta = 0$; (b) Full configuration space.

equations. For linkages with one degree of freedom, which are the norm, the contact space is an algebraic curve. HIPAIR derives the curve numerically by homotopy continuation, obtaining an explicit contact space [14].

Assemblies

We compute assembly configuration spaces by composing the configuration spaces of their pairs [14]. The correctness of this procedure follows from the compositional form of configuration space. We linearize the contact zone boundaries to a tolerance and intersect them with the simplex algorithm. The result is an approximate partition of the assembly configuration space into linearized free and blocked regions. It is difficult to visualize and work with the assembly configuration space because of its high dimension. We have found it simpler and more efficient to develop simulation, tolerancing, and parametric design algorithms that work directly with the pairwise configuration spaces. We retain assembly analysis because it provides global information that should prove useful in other design tasks.

5 Dynamical simulation

We have developed a dynamical simulator for planar assemblies based on configuration space computation [26]. Dynamical assembly simulation allows designers to visualize part motions, to validate assembly function, to optimize performance, and to compute loads for stress analysis. Simulation is an iterative process in which equations of motion for the parts of a system are integrated over time. Contact analysis plays the central role of determining the touching parts and the ensuing contact forces at each time step.

Mechanical systems simulators [13, 28] provide contact models for lower pairs and require the user to provide models for other contacts. This is appropriate for linkages and robot arms, but requires an excessive modeling effort for assemblies with multiple contacts and complex contact geometry, such as clock escapements, gear chains, and part feeders.

General rigid-body simulators [2, 6, 21] compute the dynamics of general polyhedral assemblies without user contact analysis by testing for part collisions at each time step. The worst-case time complexity of the contact analysis is quadratic in the geometric complexity of the parts because every pair of parts can touch at every feature. The simulators speed up contact analysis with collision detection heuristics, such as spatial partitioning, which avoids comparisons between distant parts, and coherent computation, which predicts current contacts based on the past [19]. These simulators have several potential drawbacks for mechanical assembly simulation. The collision detection heuristics are designed for loosely coupled systems where few part are close together at most times and where part velocities are small relative to inter-part distances, such as a moving object in a static world, pendulums, rolling balls, and rock slides. It is unclear how well the heuristics work in the mechanical domain where most parts interact, contact changes are common, and parts are driven fast. The algorithms approximate curved parts with polyhedra, which creates spurious discontinuities in the contact functions that distort the dynamics of high-speed systems. The approximation also increases the running time when the parts interact often, as do many parts in mechanical systems, including lower pairs with play, gears, and cams. Collision detection with curved parts is possible [30], but appears impractical for dynamical simulation [29].

Our simulator replaces collision detection with configuration space computation. The user inputs the part shapes, masses, moments of inertia, friction and restitution coefficients, external forces, and initial configurations. The simulator precomputes the configuration spaces of the interacting pairs in the assembly. At each simulation step, it computes the contact forces in the current state, combines them with the external forces, and predicts the next state by integrating the Newtonian equations of the parts. The configuration spaces provide the contact data (which parts touch and where) for contact force computation. The simulator tests for part collisions between steps, which create discontinuities in the contact forces and in the part velocities, by querying the configuration spaces for transitions from free to contact space. It terminates the step at the collision time, updates the state, and resumes simulation.

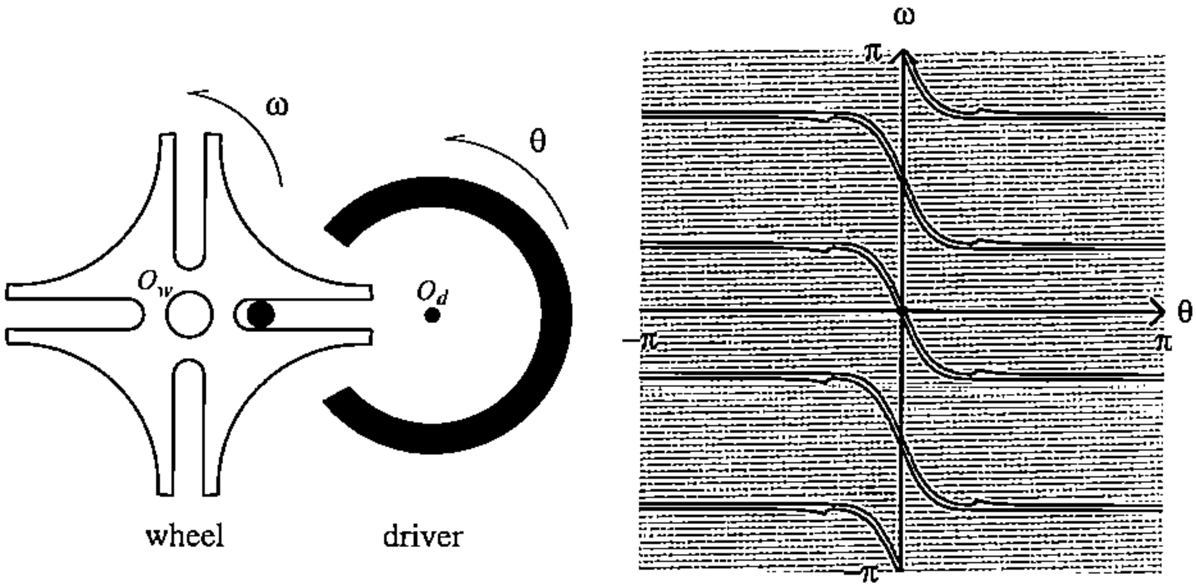


Figure 7: Geneva pair and its configuration space. The pair is displayed in configuration $\theta = 0$, $\omega = 0$, marked by the dot at the configuration space origin.

The worst-case running time of the configuration space computation is quadratic in the geometric complexity of the parts, as is a single collision detection. The queries take linear time in the number of parts and are independent of their geometric complexity. The program handles curved parts exactly. We have simulated the movie film advance (Figure 5), a Geneva pair (described next), a clock escapement governor, a planar knee reconstructed from CT data, and many other assemblies. The computations are ten to twenty times faster than real-time at 0.1% accuracy.

We illustrate configuration space simulation on a simple, but realistic scenario involving the design of a Geneva pair (Figure 7). The goal is to maximize the throughput of an assembly line where the Geneva pair alternately advances and locks a conveyor belt. We wish to simulate a range of driving torques to see how fast the pair can be driven safely. The collision detection approach is problematic because the parts are curved, the part clearances are very small, contact changes are frequent, and the velocities are high.

The configuration space shows that the Geneva pair has the correct function. The free space forms a single channel that wraps around the horizontal and vertical boundaries. The diagonal segments represent contacts between the driver pin and the wheel slots, which rotate the wheel. The horizontal segments represent contacts between locking arc segments, which hold the wheel stationary. As the driver rotates, the configuration follows the channel with the wheel rotating in the diagonal segments and blocking in the horizontal segments. The fact that the free space forms a two-dimensional channel, rather than a one-dimensional curve, indicates part play, which we investigate during the dynamical simulation.

driving torque	*rpm*	*maximum angular velocity*	*maximum contact force*	*maximum impact velocity*
1	60	13	8	63
5	120	32	20	139
10	180	46	40	196
20	300	71	158	227
50	600	114	253	448

Table 1: Geneva simulation results with driving torque in Newton-centimeters, angular velocity in radians per second, contact force in Newtons, and relative velocity at impact in meters per second.

We simulate the Geneva pair with HIPAIR in half a second per simulated second. We assume a coefficient of restitution of 0.3 and frictional coefficients of 0, which are typical values for lubricated steel parts. We assign each part a moment of inertia of 1 newton-centimeter^2, which corresponds roughly to a mass of one kilogram uniformly distributed over its profile. We find that the pair reaches steady-state behavior within one cycle. Table 1 summarizes the steady-state function. The cycle frequency increases with the driving torque, but at the cost of increased contact forces and impact velocities, which can increase part wear and can cause failure due to deformation or fracture. The simulation results provide the input to finite-element codes that test for these failure modes.

6 Kinematic tolerance analysis

The goal of tolerance analysis is to compute the variation in the function of mechanical assemblies resulting from manufacturing variation in the shapes and configurations of their parts. Kinematic tolerance analysis computes the variations of the kinematic function determined by the series of contact constraints over the assembly work cycle. For example, a meshed pair of rotating gears undergoes a series of tooth contacts that impose a relation between the gear angular velocities. Ideal gears transmit rotation linearly, whereas real gears exhibit backlash and chatter because of axis misalignment and gear profile imperfections. Designers use kinematic tolerance analysis to guarantee correct assembly function and to reduce manufacturing cost. Worst-case analysis derives guaranteed upper and lower bounds on the variation, while statistical analysis derives probabilistic bounds. The analysis complements tolerancing for assembly, which verifies that the parts can be assembled despite shape variations.

Tolerance specifications define the allowable variation in the shape and configurations of the parts of an assembly. The most common are parametric and geometric specifications[32].

Parametric specifications restrict the parameters of the assembly model to intervals of values. For example, a tolerance of $r = 1 \pm 0.1$ restricts the radius r of a disk to the interval $[0.9, 1.1]$. Geometric specifications restrict part features to zones around the nominal features, typically to fixed-width bands, called uniform profile tolerance zones, whose boundaries are the geometric inset and offset of the nominal features. For example, a uniform geometric profile tolerance of 0.1 on a disk of radius 1 constrains its surface to lie inside an annulus with outer radius 1.1 and inner radius 0.9. We discuss parametric tolerances because they are best suited to kinematic tolerance analysis. We analyze geometric tolerances by translating them into parametric tolerances or by a direct method [16].

Parametric kinematic tolerance analysis consists of contact analysis and sensitivity analysis steps. The contact analysis derives the functional relationship between the tolerance parameters and the assembly kinematic function. The sensitivity analysis determines the variation of the kinematic function over the allowable parameter values. Contact analysis has not been automated previously. It is difficult to perform manually because the contact constraints are numerous, are complicated, and vary during the work cycle. Multiple contacts occur in nominal designs with higher pairs, such as gears, cams, clutches, and ratchets. Part variations produce multiple contacts even in assemblies whose nominal designs involve only permanent contacts. Sensitivity analysis algorithms are well developed. The principal methods are linearization, statistics, and Monte Carlo simulation [5].

We have generalized the configuration space representation to model kinematic variation of toleranced parts and have developed a contact analysis algorithm for parametric planar assemblies with one degree of freedom per part [16, 27]. We couple the contact analysis with sensitivity analysis to obtain a program that derives worst-case and statistical bounds on kinematic variation along with qualitative changes in kinematic function, such as jamming, under-cutting, and interference. The program is fast enough to be practical for complete functional models of complex assemblies and for parametric representations of geometric tolerances, such as offsets, which typically require many parameters. The extension to general planar assemblies is straightforward.

Worst-case analysis of pairs

We model kinematic variation by generalizing the configuration space representation to toleranced parts. The contact curves are parameterized by the tolerance parameters. As the parameters vary around their nominal values, the contact curves vary in a band around the nominal contact space, which we call the contact zone. For example, Figure 8 shows a 26 parameter model of the Geneva pair and Figure 9 shows sample contact zones, each computed to 0.01% accuracy in 20 seconds. The contact zone defines the kinematic variation in each contact configuration: every pair that satisfies the part tolerances generates a contact space that lies in the contact zone.

Each contact curve generates a region in the contact zone that represents the kinematic

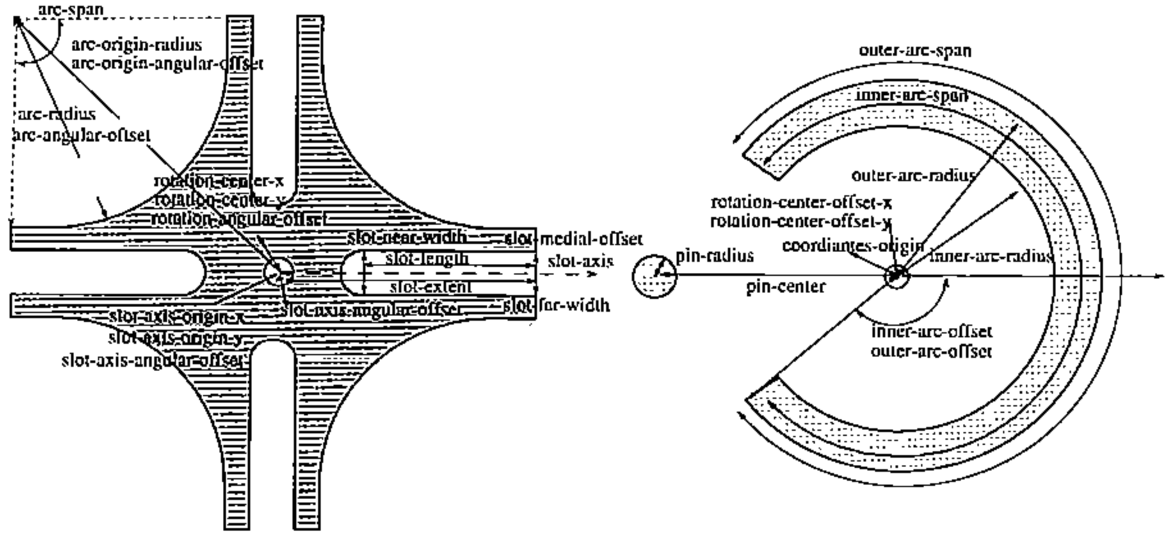


Figure 8: Parametric model of the Geneva pair.

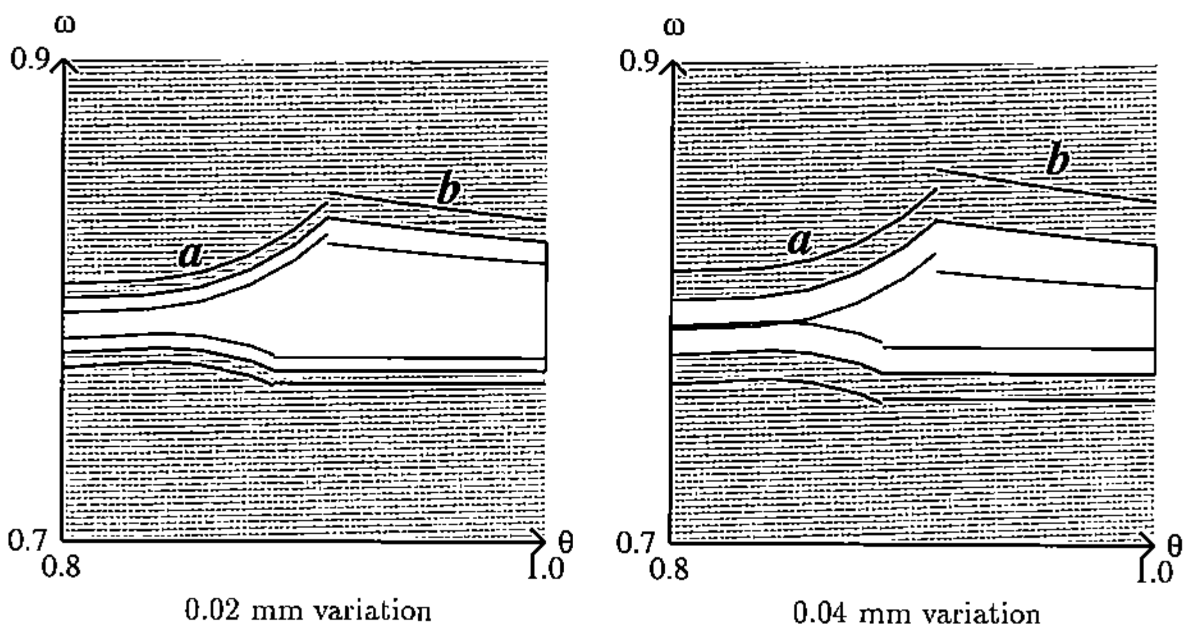


Figure 9: Detail of the contact zone of the Geneva pair in the region where the driver locking segment disengages from the wheel locking segment and the driver pin engages the slot of the wheel. The center curves are the nominal contact space. The upper and lower curves bound the contact zone.

variation in the corresponding part contact. The region boundaries encode the worst-case kinematic variation over the allowable parameter variations. They are smooth functions of the tolerance parameters and of the assembly configuration in each region. They are typically discontinuous at region boundaries because the contact curves depend on different parameters, as on the boundary between regions a and b in Figure 9. The variation at transition points is the maximum over the neighboring region endpoints. The contact zone also captures qualitative changes in kinematics, such as jamming, under-cutting, and interference. For example, the Geneva pair can jam when the contact zones of the upper and lower channels overlap, meaning that the channel closes for some allowable parts. The figure shows that this occurs when the variation equals 0.04 mm per parameter.

We compute the contact zone from the parametric model of the pair. The inputs are the part models, the nominal values and allowable ranges of the parameters, and an error bound. The outputs are closed-form expressions for the contact zone boundary. We first compute the nominal contact space with HIPAIR, obtaining a collection of contact curves of the form $y = f(x)$. We then derive parametric contact curves $y = f(x, \mathbf{p})$ by instantiating the contact table entries of the nominal curves with the symbolic tolerancing parameters $\mathbf{p}$ instead of with the nominal values. As $\mathbf{p}$ ranges over the allowable values, the parametric curves range over the contact zone. We compute closed-form expressions for the upper and lower boundaries of the contact zone by linearizing f around the nominal $\mathbf{p}$ values, making the standard tolerancing approximation that the kinematic variation is linear in the parameter variations.

Table 2 shows the results of the sensitivity analysis for the two regions shown in Figure 9. In region a where the driver pin touches the corner of the wheel slot, the two most important parameters are the wheel slot-axis-angular-offset and the driver pin-radius. The former accounts for 40%–45% of the variation, while both account for 49%–52% of the variation. In region b where the driver locking arc touches the wheel locking arc, the two most important parameters are the wheel arc-origin-angular-offset and arc-radius. The former accounts for 25%–50% of the variation, while both account for 38%–59% of the variation. Statistical analysis shows that the average kinematic variation is much smaller than the worst-case bounds. We derive similar results for an 82-parameter model of the shutter mechanism shown in Figure 1.

Assemblies and statistical tolerancing

The contact zone model of worst-case kinematic variation generalizes to assemblies. The assembly contact space is a semi-algebraic set in configuration space: a collection of points, curves, surfaces, and higher dimensional components. As the assembly tolerance parameters vary around their nominal values, the components vary in a contact zone around the nominal contact space. We compute the kinematic variation in a nominal operating mode, that is for specific external forces and initial conditions. This analysis is far easier than contact

part	parameter	nominal value	% of sensitivity region a	region b
driver	pin-radius	4.5	8	0
	pin-center	56.5	7	0
	outer-arc-radius	46.0	0	12
	outer-arc-span	49.416	0	3
	outer-arc-offset	-2.4708	0	3
	inner-arc-radius	36.0	0	0
	inner-arc-span	4.9416	0	0
	inner-arc-offset	-2.4708	0	0
	rotation-center-offset-x	80.0	7	11
	rotation-center-offset-y	0.0	3	4
wheel	slot-axis-origin-x	0.0	7	0
	slot-axis-origin-y	0.0	3	0
	slot-axis-angular-offset	0.0	43	0
	slot-extent	60.0	3	0
	slot-length	40.0	0	0
	slot-medial-offset	0.0	7	0
	slot-near-width	10.0	0	0
	slot-far-width	10.0	3	0
	arc-origin-radius	80.0	0	12
	arc-origin-angular-offset	0.0	0	28
	arc-radius	46.683	0	12
	arc-angular-offset	0.0	0	0
	arc-span	1.5708	0	0
	rotation-center-x	0.0	7	11
	rotation-center-y	0.0	3	4
	rotation-angular-offset	0.0	0	0

Table 2: Geneva pair nominal parameter values and relative sensitivities. Lengths are in millimeters, angles in radians.

zone computation, yet suffices for assemblies with a moderate number of operating modes, which is the norm. We compute the nominal motion path by simulation or by measurement, split it into fixed contact segments, and perform sensitivity analysis on each segment. The computation is simple and fast because it involves a single curve instead of the entire nominal contact space.

We use the worst-case analysis to perform a statistical analysis. The inputs are the pairwise contact zones, the nominal motion path, and the joint distribution of the tolerance parameters. The outputs are the distributions of the kinematic variation in the contact zones and along the motion path. We compute the kinematic distributions by propagating the input distributions through the linearized contact functions.

We have developed a comprehensive method of kinematic tolerance analysis based on our kinematic variation algorithms. The analysis is practical for complex models with many parts and parameters because the computation time is proportional to the product of the number of interacting pairs and the number of parameters per pair, both of which tend to be linear in the size of the model.

7 Parametric design

We have developed a parametric design program for part contacts based on configuration space manipulation [15]. A parametric design is one in which the shapes and the positions of the parts are specified in terms of symbolic parameters. The designer searches the space of allowable parameter values for points that realize or optimize behaviors. When the objective can be specified as a smooth function of the parameters, we can use nonlinear optimization to achieve it. This is impossible when the design involves contact changes, when the objective is to achieve a behavior, or when the objective is qualitative. The traditional approach to these problems is direct search of the parameter space. This is often impractical, especially when there are more than a few parameters. The designer must examine many points to assure that no good design has been overlooked. Each point requires a time consuming analysis. The search is even harder when the behavior is sensitive to small parameter perturbations.

We reduce search by interactively inverting the mapping from parameter values to configuration spaces. This allows the designer to modify an assembly configuration space while the design program updates the assembly to realize the changing contacts. The designer inputs a parametric model with initial parameter values. The program displays the initial pair and its configuration space. The designer specifies behavioral modifications by reshaping the configuration space with the mouse. The program continuously updates the parameter values to track the modified configuration space by differential constraint satisfaction [10, 31]. Caine describes a similar design program for part feeders and other planar, polygonal pairs [4], while Donald and Pai [9] design compliant planar fasteners by computing and manipulating contact surfaces in three-dimensional configuration spaces.

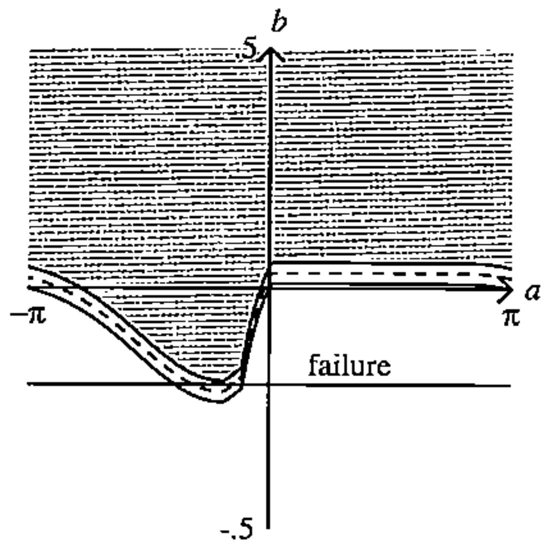


Figure 10: Interactive contact curve modification of a faulty driver/shutter pair.

We illustrate parametric design on the camera (Figure 1). The camera is appropriate for parametric design because the intended function requires a complex sequence of part interactions that is sensitive to small variations in part shapes and positions. The driver cam needs to push the shutter tip far enough to release the shutter lock. Figure 2 shows that failure occurs when the minimal shutter orientation b is greater than -0.2 radians. The designer can fix the problem by pulling down on the curved portion of the contact space in the driver/shutter configuration space or by pulling right on the bottom corner of the mouth in the shutter/shutter lock configuration space. Figure 10 illustrates the first option: it shows the faulty configuration space, an intermediate contact space (the dashed line), and the correct contact space (the solid line).

8 Conclusions

We present a unified approach to computer-aided mechanical assembly design in which all design tasks are performed within a single computational paradigm supported by integrated design software. We organize design tasks around the fundamental task of contact analysis, which we automate with configuration space computation. Configuration space is a complete, concise, and explicit representation of rigid body interactions that contains the requisite information for design tasks involving contacts. We describe the HIPAIR design environment for planar assemblies, which automates dynamical simulation and provide novel support for tolerancing and parametric design. HIPAIR has been tested on hundreds of pairs and on a dozen assemblies with up to ten moving parts. It performs at interactive speed on assemblies of ten parts with tens of thousands of contacts on a workstation.

Configuration space provides a comprehensive understanding and computational characterization of assembly contacts that systematizes diverse assembly design tasks. HIPAIR frees designers from contact analysis, which is often tedious, error-prone, or infeasible. In dynamical simulation, we replace collision detection with configuration space computation and querying, which is faster and more robust for mechanical assemblies. In kinematic tolerancing, we replace manual modeling and random parameter sampling with contact variation zones in which all parameter variations are considered. In parametric design, we replace blind search of the design space with interactive exploration. HIPAIR allows designers to perform several key design steps in a single environment. They can perform computations that lie outside the scope of previous software and that defy manual analysis. They can visualize assembly function under a range of operating conditions, can find and correct design flaws, and can evaluate the functional effects of part tolerances.

The next step in our contact analysis research is to extend HIPAIR to spatial assemblies. We need to define solid models of the parts, to derive contact constraints for spatial features, and to develop configuration space computation algorithms. The third step is impractical for general assemblies because of the high dimension of the configuration spaces. Even a single pair, which has six degrees of freedom, is probably impractical. Instead, we plan to develop specialized techniques by restricting the part geometry, motions, and interactions based on application constraints. The challenge is to obtain the data for specific tasks, especially the global data that only configuration space can provide, without computing entire configuration spaces.

We will address this challenge by dimension reduction and by selective computation. We can analyze parts with one degree of freedom, such as spatial gears and cams, by an extension of our planar algorithm. After deriving the spatial contact constraints, we can reuse the rest of the program. A similar approach applies to assemblies whose nominal motions are planar and whose off-plane motions do not cause contact changes. In general assemblies, we can construct individual configuration space regions that track the changing assembly configuration [17].

The next step in our mechanical assembly design research is extend the task coverage and to improve the algorithms. Extensions to parametric tolerance analysis include geometric form tolerances, non-kinematic parameters, and objective functions. Extensions to parametric design include general planar pairs and assembly design. Both require better parameter space exploration strategies that are less dependent on the configuration space representation. New tasks include tolerance synthesis, configuration design, flexible parts, stress analysis, functional classification, and an online assembly database. Many of these tasks require better tools for configuration space visualization and interpretation.

Acknowledgments

Ziv Yaniv developed the interface between Microstation and HIPAIR. Kwun-Nan Lin developed the reconstruction program and Dan Schikore helped produce the pictures of the three-dimensional configuration spaces. Sacks is supported in part by NSF grant CCR-9505745 and by the Purdue Center for Computational Image Analysis and Scientific Visualization.

References

[1] Avnaim, F. and Boissonnat, J. D. Polygon placement under translation and rotation. *Informatique Théorique et Applications* **31** (1989) 5–28.

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

[3] Brost, R. C. *Analysis and planning of planar manipulation tasks.* PhD thesis, Carnegie-Mellon University, 1991. Available as Technical Report CMU-CS-91-149.

[4] Caine, M. E. The design of shape interactions using motion constraints. in: *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 366–371, 1994.

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

[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] de Mello, L. S. H. and Lee, S. (Eds.). *Computer-Aided Mechanical Assembly Planning.* (Kluwer Academic Publishers, 1991).

[8] Donald, B. R. A search algorithm for motion planning with six degrees of freedom. *Artificial Intelligence* **31** (1987) 295–353.

[9] Donald, B. R. and Pai, D. The motion of planar, compliantly connected rigid bodies in contact, with applications to automatic fastening. *International Journal of Robotics Research* **12** (1993) 307–338.

[10] Gleicher, M. and Witkin, A. Through-the-lens camera control. *Computer Graphics* **26** (1992) 331–340.

[11] Goldberg, K., Halperin, D., Latombe, J., et al. (Eds.). *The Algorithmic Foundations of Robotics.* (A. K. Peters, Boston, MA, 1995).

[12] Gonzales-Palacios, M. and Angeles, J. *Cam Synthesis.* (Kluwer Academic Publishers, Dordrecht, Boston, London, 1993).

[13] Haug, E. J. *Computer-Aided Kinematics and Dynamics of Mechanical Systems*, volume I: Basic Methods. (Simon and Schuster, 1989).

[14] Joskowicz, L. and Sacks, E. Computational kinematics. *Artificial Intelligence* **51** (1991) 381–416. reprinted in [11].

[15] Joskowicz, L. and Sacks, E. Configuration space computation for mechanism design. in: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation.* IEEE Computer Society Press, 1994.

[16] Joskowicz, L., Sacks, E., and Srinivasan, V. Kinematic tolerance analysis. *Computer-Aided Design* **29** (1997) 147–157.

[17] Joskowicz, L. and Taylor, R. H. Interference-free insertion of a solid body into a cavity: An algorithm and a medical application. *International Journal of Robotics Research* **15** (1996) 211–229.

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

[19] Lin, M. C., Manocha, D., Cohen, J., et al. Collision detection: Algorithms and applications. in: *Proceedings of the 2nd Workshop on Algorithmic Foundations of Robotics,* 1996.

[20] Litvin, F. L. *Gear Geometry and Applied Theory.* (Prentice Hall, New Jersey, 1994).

[21] Mirtich, B. and Canny, J. Impulse-based dynamic simulation. In Goldberg et al. [11].

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

[23] Reuleaux, F. *The Kinematics of Machinery: Outline of a Theory of Machines.* (Dover Publications, 1963).

[24] Sacks, E. and Bajaj, C. Sliced configuration spaces for curved planar bodies. *International Journal of Robotics Research* (1997). to appear.

[25] Sacks, E. and Joskowicz, L. Computational kinematic analysis of higher pairs with multiple contacts. *Journal of Mechanical Design* **117** (1995) 269–277.

[26] Sacks, E. and Joskowicz, L. Mechanical systems simulation using configuration spaces. Technical Report 96-046, Purdue University, 1996.

[27] Sacks, E. and Joskowicz, L. Parametric kinematic tolerance analysis of planar mechanisms. *Computer-Aided Design* **27** (1997). to appear.

[28] Schiehlen, W. *Multibody systems handbook.* (Springer-Verlag, 1990).

[29] Snyder, J. M. An interactive tool for placing curved surfaces without interpenetration. in: *Proceedings of SIGGRAPH'95*, 1995.

[30] Snyder, J. M., Woodbury, A. R., Fleischer, K., et al. Interval method for multi-point collisions between time-dependent curved surfaces. in: *Proceedings of SIGGRAPH'93*, 1993.

[31] Tidd, W. F., Rinderle, J. R., and Witkin, A. Design refinement via interactive manipulation of design parameters and behaviors. in: *Proceedings of the 4th Design Theory and Methodology Conference*, 1992.

[32] Voelcker, H. A current perspective on tolerancing and metrology. *Manufacturing Review* 6 (1993).