Equivalence Classes for Shape Synthesis of Moving Mechanical Parts*

Horea T. Ilieş
Ford Motor Company†
hilies@ford.com

Vadim Shapiro
University of Wisconsin-Madison‡
vshapiro@engr.wisc.edu

Abstract

Moving parts in contact have been traditionally synthesized through specialized techniques that focus on completely specified nominal shapes. Given that the functionality does not completely constrain the geometry of any given part, the design process leads to arbitrarily specified portions of geometry, without providing support for systematic generation of alternative shapes satisfying identical or altered functionalities. Hence the design cycle of a product is forced to go into numerous and often redundant iterative stages that directly impact its effectiveness.

We argue that the shape synthesis of mechanical parts is more efficient and less error prone if it is based on techniques that identify the functional surfaces of the part without imposing arbitrary restrictions on its geometry. We demonstrate that such techniques can be formally defined for parts moving in contact through equivalence classes of mechanical parts that satisfy a given functionality. We show here that by replacing the completely specified geometry of the traditional approaches with partial geometry *and* functional specification, we can formally define classes of mechanical parts that are equivalent, in the sense that all members of the class satisfy the same functional specifications. Moreover, these classes of functionally equivalent parts are computable, may be represented unambiguously by maximal elements in each class, and contain all other functional designs that perform the same function.

Keywords: Equivalence classes, functional equivalence, shape synthesis, design space, higher pairs, contact, sweep, unsweep, moving parts.

1 Motivation

Conventional design of mechanical parts focuses on generating a single completely specified nominal shape that is toleranced to allow for variations in form. The corresponding design processes usually involve arbitrary decisions affecting the geometry, and do not support systematic generation of alternative shapes satisfying identical or altered functionalities. This places a serious handicap on the design cycle of a product, since most new designs are obtained by modifying existing products to comply with new functional specifications. Despite their well known benefits, these essentially "generate and test" scenarios are costly and time consuming during the product development process, and their effectiveness is limited by the available "design experience", as well as by the time and resources that are being allocated for product development.

A comprehensive review of mechanical design literature is outside the scope of this paper. Broadly, the existing approaches to conceptual design focus on formulating a set of design rules that need to be satisfied [1, 2], on interactive (and iterative) specialized design approaches, such as knowledge-based reasoning, and on applying various search techniques - typically "analysis in the loop" type approaches based on generate and test paradigms – to find a single feasible solution for the design problem [3]. While the more recent solution search techniques, such as "path finding", "constraint satisfaction", "simulated annealing", "genetic

*To appear in ASME Transactions, Journal of Computing and Information Science in Engineering.

†Please use e-mail as means of contact. Alternate e-mail address: ilies@sal-cnc.me.wisc.edu

‡Spatial Automation Laboratory, Department of Mechanical Engineering, 1513 University Avenue, University of Wisconsin-Madison, 53706 USA.

algorithms" and various shape optimization techniques [4, 5], are becoming more popular, they typically lead to single designs with fully specified geometries. Configuration spaces (C-spaces) have been shown to be a convenient representation of the design space for parts moving in contact because the C-spaces explicitly capture the motion constraints imposed by the relative motion, as discussed, for example, in [6, 7, 8, 9]. C-spaces are typically used to validate constrained motion of *known* shapes, and techniques based on C-spaces are, fundamentally, analysis tools. However, constraining and designing shapes using C-spaces has proved problematic due to a number of representational and computational issues [6, 9, 10].

Consider the automotive latch shown in Figure 1, which is part of a typical hood latch assembly of an automobile. Broadly, the functionality of a hood latch assembly is to engage and retain a (generally cylindrical) striker[1] *and* to prevent it from accidental release. During its downward motion, a vertically

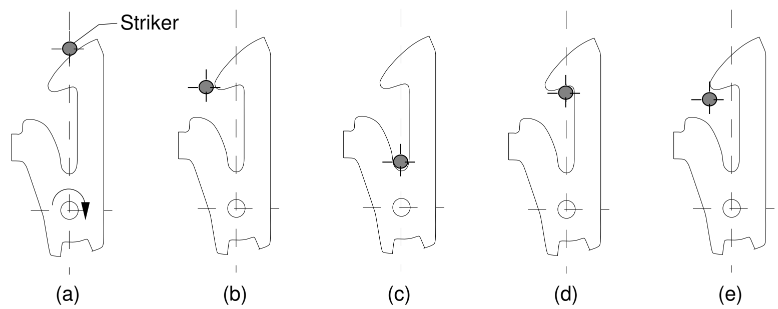


Figure 1: A secondary hood latch must engage and retain a vertically moving striker.

moving striker forces the secondary latch to rotate and latch it. While the striker continues to move vertically down under the influence of externally applied loads, the secondary latch is brought back to its original position by an attached reaction spring; the primary latch, not shown, closes down and holds the striker fixed. The release of the striker from the hood latch assembly is accomplished in two phases: first, manually applying external forces indirectly to the primary latch from the cabin, and then directly to the secondary latch – so that the secondary latch rotates and the hood can be lifted. A spring (not shown) attached to the secondary latch provides the reaction force needed to insure the return to its original position, but the latch must continue to function even if the spring breaks. The secondary latch must also prevent the upward motion of the striker (and therefore of the hood) when the primary latch fails to engage the striker. In addition, the secondary latch has to remain inside a specified containing set during its motion to avoid interference with neighboring parts.

We focus here on the *contact function* of the latch, since engaging the striker and avoiding its accidental release is accomplished by the contact between the striker and the latch. A complete design of such a latch must also take into consideration other constraints, such as strength, containment, and manufacturability, as discussed in [11, 12]. At the very least, the contact function presumes that (1) each part is contained in its work space without interfering with other parts; (2) the moving parts are positioned relative to each other so that their boundaries always touch each other; and (3) the contact surfaces are able to support the externally applied loads on the contacting parts. Specifically, in the case of the latch shown in Figure 1, the latch comes in contact with the striker, it must not interfere with other components, and external forces are applied indirectly to the striker through the hood of the automobile.

This latch is an example of a part that is often redesigned to accommodate new car models and/or changing requirements. Since the geometry of the latch, in itself, does not contain any information about the functionality of the latch, or about the reasons that led the designer to choose this particular geometry, the redesign process often requires guesswork or a complete re-design of the part. This leads to costly generate and test design procedures that may not capture the original design intent, either because the process may not converge, or because the geometry may be overconstrained, or because we may simply not know whether a solution exists (i.e., the corresponding equivalence class is empty).

[1] In this example the striker is attached to the hood of the automobile.

We can use this common automotive latch to make few essential observations that are central to the rest of the paper. The functionality of the latch determines portions of functional geometry, and *all* functional surfaces are determined by some intended function of the latch. On the other hand, non-functional surfaces can be regarded as opportunities to improve the performance of the part. This implies the existence of a *class* of (infinitely many) functionally equivalent designs that satisfy the same mechanical function. Thus, there are infinitely many shapes that satisfy the specified containment constraints for a given relative motion. At the same time, there are infinitely many shapes (some of them unbounded) that can move in contact with the striker according to the known motion[2], but not all these shapes will simultaneously satisfy the containment constraints. Among the latch shapes that do, some may not latch the striker for a given set of externally applied loads. This would eliminate them from the feasible designs and effectively reduce the design space. It is intuitively clear that each of the three constraints induces a class of functionally equivalent designs, and that any selected part must lie in the intersection of all three classes that still contains infinitely many parts.

Outline

In this paper we focus on the problem of designing a moving object B that maintains contact with a given moving object A during a known relative motion M. To this end, we identify in section 2 three equivalence classes[3] for artifacts with parts moving in contact under applied loads that are also subjected to containment constraints. We show that each of these classes may be represented unambiguously by maximal elements of the class (see also [13]), and is augmented by a partial ordering of its elements. The *containment class* addresses the spatial containment constraints on a moving parts. *Positional equivalence class* contains all parts B that may come in contact with a moving object A during a known motion M. The positional equivalence [14] enforces conditions on the geometry and motions of the elements of the class without considering the externally applied loads. The latter are taken into account by the *class of statically equivalent* parts, which is a subset of the positional equivalence class, as discussed in section 2.

We show in section 3 that by replacing the completely specified geometry of the traditional approaches with partial geometry *and* functional specification, we can generate classes of mechanical parts that are equivalent in the sense that all members of the class satisfy the same functional specifications. In particular, we show that our approach captures the design decisions by generating functional surfaces that do not embed arbitrary geometric restrictions, and identifies fully defined representative members of the equivalence classes that contain an essentially unlimited number of functional designs. We finally discuss in sections 3 and 4 some of the limitations as well as notable extensions of this work.

2 Functional Equivalence Classes

An equivalence class is a set of elements that are related to each other via an equivalence relation "$\sim$" that must be, by definition, reflexive ($a \sim a$), symmetric ($a \sim b$ implies $b \sim a$) and transitive ($a \sim b$ and $b \sim c$ implies $a \sim c$).

The main theme of this paper is that equivalence classes become particularly important in shape synthesis of mechanical parts when one can define equivalent relations that capture the functionality of a given part. In this case, any member of the equivalence class will be functionally equivalent with any other member of the same class. At the very least, such an equivalence class must have computable representative members, and a well defined test for membership in the class.

If there is a partial ordering of the elements within each class so that each class has a maximal element, then all other elements of the class can be compared to that maximal element of the class. For the containment equivalence class the partial ordering follows from the standard subset relationship between sets, while for the positional and static equivalence classes, the partial orderings follow from the *restriction* and the *static restriction* of triplets discussed in sections 2.2 and 2.3 respectively.

[2] The simplest way to generate two such shapes is to add non-functional holes to a functional latch. Intuitively, one can also modify non-functional surfaces without changing the function of the part.

[3] These particular classes of equivalence are driven by the common requirements for moving parts in contact that we mentioned above. As we discuss in section 4, there are other types of functional requirements that a mechanical part must typically satisfy.

2.1 Containment Equivalence

Every moving part A needs to remain inside a given containing space E to prevent interference with other parts during its operation. In this paper a motion M is a one parameter family of transformations $M(t)$ from the normalized time domain $t \in [0,1]$ to the space of geometric configurations $\mathcal{C}$ [11]. Then the containment constraint can be formally expressed as

$$A^{M(t)} \subset E, \ \forall t \in [0,1] \tag{1}$$

where $A^{M(t)}$ denotes set A at instance t during its motion M.

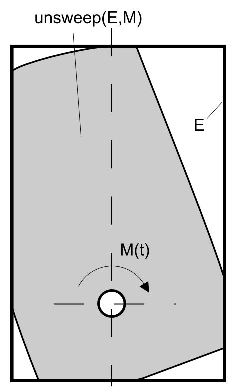


Figure 2: The largest shape (set of points) remaining inside a containing set E during a known motion $M(t)$ is given by $\texttt{unsweep}(E, M)$. The motion is a clockwise rotation with an angle of 21^o.

All sets that satisfy the condition (1) for a given containing set E and motion M are equivalent because they all serve the same containment function. Formally,

Definition 2.1 *Two objects A_1 and A_2 are part of the same* equivalence class of containment *– or containment equivalence for short, if they both satisfy equation (1) with the same containing set E and motion $M(t)$.*

The above definition is not very useful for design purposes because it does not suggest a direct method for deciding whether two shapes A_1 and A_2 belong to the same containment equivalence class, or how a shape may be modified without violating the containment constraint. The problem is solved by identifying the unique *maximal* shape in the containment equivalence class. This maximal shape is formally defined by the `unsweep` operation, which returns the largest moving set of points remaining inside a given containing set, and can be thought of to be a material removal operation [12]. As shown in Figure 2, the largest "latch" remaining inside the stationary containing set E during a known motion M is $\texttt{unsweep}(E, M)$, and contains *all* points of E that would remain inside E during M. This maximal set $\texttt{unsweep}(E, M)$ defines an equivalence class in the sense that any of its subsets will satisfy the same containment constraints. In practice, it can be computed based on the two equivalent definitions of `unsweep`, one in terms of a trajectory test, and the other in terms of an infinite intersection of the moving set:

$$\begin{gathered} \texttt{unsweep}(E, M) = \{x \mid T_x \subset E\} \\ \texttt{unsweep}(E, M) = \textstyle\bigcap_{q \in \hat{M}} E^q \end{gathered} \tag{2}$$

where T_x is the trajectory of a moving point $x \in E$, and q is a configuration that determines the position and orientation of a moving object relative to a fixed coordinate system. Furthermore, $\hat{M}$ is the *inverted* motion that corresponds to motion M so that for a range of values of $t \in [0;1]$, $\hat{M}(t)$ is the inverse of $M(t)$

for every instance of t. These definitions can be used to prove the duality between `unsweep` and the general `sweep` operation (see also [12] for details), which is given by

$$[\texttt{unsweep}(A^c, \hat{M})]^c = \texttt{sweep}(A, M). \tag{3}$$

2.2 Positional Equivalence

Informally, moving parts that do not interfere may or may not serve the desired contact function, depending on whether their respective boundaries are touching each other at all times during the motion. On the other hand, all pairs of shapes that can execute the same motion while touching each other may be deemed *positionally equivalent.*

We formally introduced *conjugate triplets* in [10, 14] as a means of representing and manipulating classes of positionally equivalent parts. A conjugate triplet $\Psi = \langle A, B, M \rangle$ consists of two shapes A and B moving relative to each other according to a motion M while touching each other. As illustrated in Figure 3(a-c), a triplet may not exist for given shapes and motion because A and B may lose contact during M. However, if one triplet does exist, then there are infinitely many such conjugate shapes for the same motion. This assertion naturally leads to the notion of positionally equivalent triplets that are formally defined in [14]. Informally, triplet $\Psi_1 = \langle A_1, B_1, M_1 \rangle$ is a *positional restriction* of another triplet $\Psi_2 = \langle A_2, B_2, M_2 \rangle$ if the contact boundaries of A_1 and B_1 are subsets of those of A_2 and B_2 respectively, and if the two motions M_1 and M_2 share the same set of configurations. Then, two triplets are positionally equivalent if they are positional restrictions of a larger known conjugate triplet Ψ.

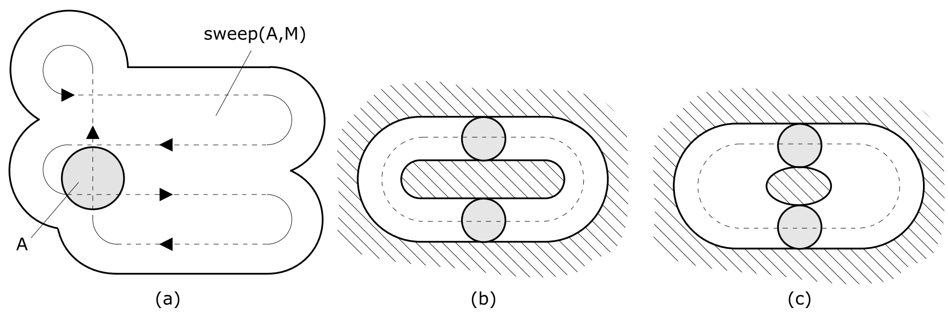


Figure 3: A conjugate triplet, which requires continuous geometric contact, does not always exist for given object and motion (a). If one conjugate triplet exists (b), there are infinitely many conjugate triplets one of which is shown in (c).

The positional equivalence is essentially a "less than or equal" relationship, which induces a partial ordering of all triplets. It follows that the positional equivalence class can simply be defined by providing the largest element of the class. The fact that the positional equivalence with respect to some *fixed* triplet Ψ is indeed an equivalence relation follows from arguments analogous to those used to prove that all numbers less than some fixed number X are equivalent to each other. One can similarly show that positional equivalence with respect to Ψ is both symmetric and transitive. Thus, *any* given triplet immediately induces an equivalence class of triplets that are its positional restrictions.

Rather than comparing triplets directly, we seek a unique maximal triplet capable of representing the corresponding equivalence class. Starting with a given triplet $\Psi = \langle A, B, M \rangle$, let us fix motion M and part A, but let B 'grow'[4] while its boundaries remain in contact with A at all times. This process may produce additional contacts between A and B, but cannot eliminate the contacts that are already present. Thus Ψ is guaranteed to be in the positional equivalence class of the grown triplet. The growing process stops when the largest possible object B^A_{max} cannot be grown without violating the contact constraints. But now we

[4]Here 'growth' is a conceptual explanatory mechanism and not used as a definition or in a computational sense. Intuitively, an object that 'grows' undergoes incremental material addition to its shape.

can also grow object A to obtain the new set A^A_{max} that remains in contact with B^A_{max} for the same motion M. Formally,

$$\begin{aligned} B^A_{max} &= k[(\texttt{sweep}(A, M))^c] \\ A^A_{max} &= k[(\texttt{unsweep}((B^A_{max})^c, M))] \end{aligned} \quad (4)$$

where X^c denotes the standard set complement of a set X, and kX is the closure of X. The properties of `sweep` and `unsweep` operations imply that both B^A_{max} and A^A_{max} are well defined for a given Ψ, and in this sense Ψ^A is *unique*. Note that $A \subset A^A_{max}$, based on the properties of the `unsweep` operation discussed in [12]. In other words, A^A_{max} is the largest (often unbounded) A that would generate the largest B during the given M.

Figures 4(a-c) show the largest latch shape B^A_{max} that moves in contact with the striker A according to the relative motion M. It is obvious that the set shown in Figures 4(a-c) is not related from a geometrical point of view to the set shown in Figure 2. The largest striker A^A_{max} (not shown) can be similarly obtained from equations (4).

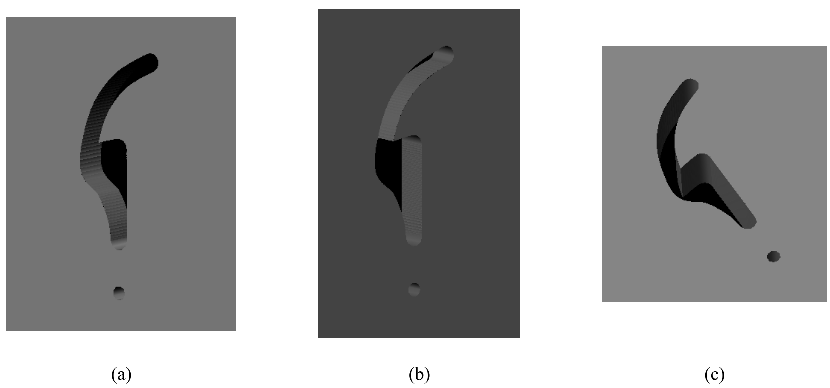


Figure 4: The largest connected latch seen here from three different directions, which includes all other positionally equivalent shapes moving in contact with the given striker according to a known motion. It corresponds to set B^A_{max} in equation (4).

Because we have a choice of which object to grow first, the above process is clearly order-dependent and asymmetric with respect to A and B. Therefore there exists another maximal conjugate triplet $\Psi^B = \langle B^B_{max}, A^B_{max}, \hat{M} \rangle$ defined as in (4) with A and B interchanged, with the maximal set A^B_{max} that will come in contact with B while B moves relative to A^B_{max} according to $\hat{M}$. Thus, in principle, we could fix the shape of the latch and design the largest possible striker that works with this latch, even if this does not appear practical for this particular application.

To summarize, any given triplet Ψ naturally belongs to two unique and distinct equivalence classes that are represented unambiguously by *two* maximal triplets. These triplets are made up from the largest (and hence unique) shapes that maintain the specified motion and contact. If the initial object A is given, then the function of the triplet is to maintain a moving contact with the given A, and we focus on the properties of the maximal triplet Ψ^A. Changes to A are permitted *only* to the extent that they do not affect the shape of the largest conjugate shape B, because this would effectively lead to a new design problem. By symmetry, the maximal triplet Ψ^B should be used when designing the shape of A to move in contact with a partially known object B.

2.3 Static Equivalence

The positional equivalence class discussed above captures the contact constraints imposed on a given part by identifying the boundary of a shape B which contains *all* points that will come in contact with a given object

A during a prescribed motion M. But for any given set of applied external loads, only some of these contact points will serve the contact function by carrying loads and transmitting the contact forces. This implies that those points that are not carrying loads can be eliminated without violating the contact function.

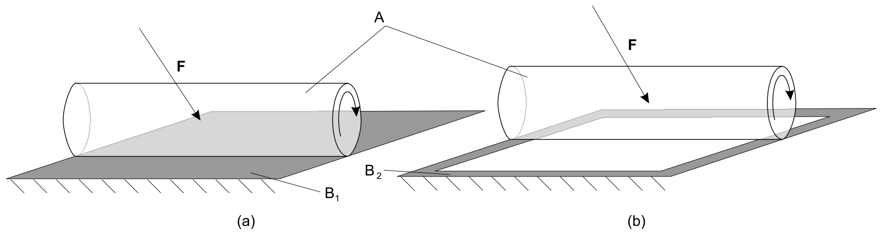


Figure 5: A cylinder rolling on a plane under the influence of an externally applied force. Set B_1 in (a) shows all contact points between the cylinder and the plane. Set B_2 shows a subset of B_1 that *may* be sufficient to carry the applied force.

For example, Figure 5 shows a cylinder rolling on a planar surface under an applied external force $\mathbf{F}$ and set B_1 contains all contact points of the planar surface that carry the load $\mathbf{F}$. In general, not all these boundary points B_1 are required by a given $\mathbf{F}$, and Figure 5(b) shows a subset B_2 of B_1 that may be sufficient to carry the applied load. It follows that the two sets B_1 and B_2 perform the same function and are functionally equivalent in this sense.

The *static equivalence* class identifies all shapes that share boundary points demanded by the contact constraints and the applied loads. Our definition assumes quasi-static relative motion of the parts – so that inertia effects are negligible, and relies on a classification of all the boundary points of the maximal triplets into *load-bearing* (the contact points that carry the applied loads - formally defined in appendix A), and *load-free* boundary points. The maximal triplets are a logical starting point because, by definition, the boundaries of the maximal parts contain *all* points where load-bearing contact may take place. The set of all load-bearing points of the contact boundary form the *load-bearing boundaries* of B. Therefore the *load-free boundaries* of B are the remaining contact points of B, and the union of the load-bearing and load-free points form the contact boundary of B. As we formally show in appendix A, this boundary partitioning[5] can be obtained through straightforward geometric computations.

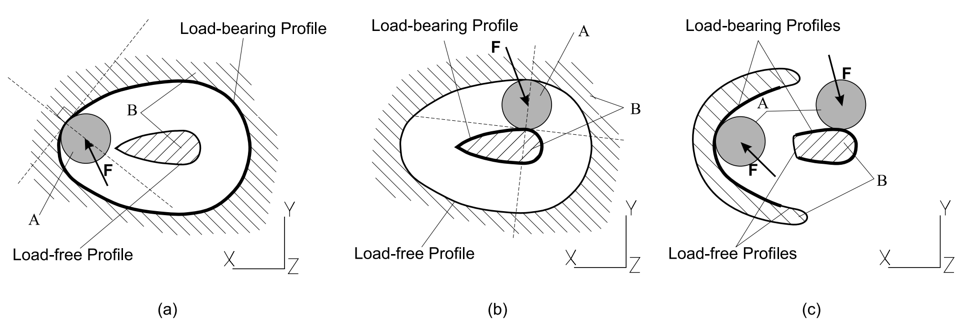


Figure 6: A cylindrical follower A maintains contact with B while moving under the influence of externally applied loads. For loading condition shown in (a) only the points on the outer profile of B are carrying the loads, but different loading conditions may redistribute these load-bearing points as shown in (b) and (c).

[5]Load-bearing and load-free contact points and boundaries of A can be defined in a similar manner. But since we assume A to be given, we are interested only in determining the load-bearing and load-free contact points of B. By symmetry, the load-bearing boundaries of A will contain points that provide support through contact to B.

Figure 6 shows the maximal triplet corresponding to a cylindrical follower A and known relative motion M, while an external force $\mathbf{F}$ is applied on A so that $\mathbf{F}$ has a component collinear and opposite to the surface normal at the contact point throughout the relative motion. More generally, at a given contact point between A and B, there can be a contact force that is either positive (one object "supports" the other – in the sense that the two objects interact based on the action-reaction principle) or zero. We do *not* assume frictionless contact, but observe that when the normal contact force at a given contact point is zero, the friction force at that point is also zero independently of the friction model being used.

For a quasi-static motion, determining whether a given contact point P is load-bearing requires three pieces of information [10]: the directions of the external loads - which are generally known, the coordinates of the contact point P and information about the surface normal at P. Since we have complete geometric information about the objects, the contact points and their normals are easily computed as shown in appendix A. For the example shown in Figure 6(a), all contact points of the outer profile of B are load-bearing, while all the contact points of the inner profile are load-free. Changing the direction of the externally applied loads may change the spatial distribution of the load-bearing points. For example, all load-bearing points are part of the boundary of the inner profile in Figure 6(b). More interesting may be a situation where the load-bearing points are shared by both the inner and outer profiles of B for certain loading conditions as illustrated in Figure 6(c). Note that the contact function of a mechanism requires load-bearing points for every relative configuration between A and B.

Defining the Static Equivalence

This classification of boundary points into load-bearing and load-free points naturally leads to a subclass of the positional equivalence class discussed in Section 2.2. Informally, only those members of the positional equivalence class that contain load-bearing boundary points at every configuration of the relative motion will be members of the static equivalence class. In other words, the class of statically equivalent triplets is a subclass of the positional equivalence class.

Assume that $\Psi_1 = \langle A_1, B_1, M_1 \rangle$ is a triplet, and thus A_1 and B_1 maintain contact at every configuration of M_1. Also assume that A_1 and B_1 have load-bearing points at every configuration $q \in M_1$. Observe that duality between contact point and contact forces implies that if load-bearing points are required on B_1 at every configuration q, one is guaranteed to have load-bearing points on A_1 at every configuration of the relative motion.

Definition 2.2 *For a given set of loads externally applied to two triplets $\Psi_1 = \langle A_1, B_1, M_1 \rangle$ and $\Psi_2 = \langle A_2, B_2, M_2 \rangle$, we say that triplet Ψ_2 is a* static restriction *of a triplet Ψ_1, and we denote this by $\Psi_2 \sqsubseteq \Psi_1$, if the following conditions are satisfied:*

(1) *Ψ_1 and Ψ_2 are positionally equivalent;*

(2) *the load-bearing boundaries of B_2 and A_2 are subsets of the corresponding load-bearing boundaries of B_1 and A_1;*

(3) *Ψ_2 has load-bearing boundaries at every configuration of motion M_2.*

Based on the definition of the positional equivalence, the two motions M_1 and M_2 of Ψ_1 and Ψ_2 will share the same set of configurations [10, 14], and the two triplets will satisfy an inclusion relationship between the corresponding contact boundaries. Note that condition (3) in Definition 2.2 implies (based on condition (2)) that triplet Ψ_1 has load-bearing contact at every configuration $p \in M_1$.

Thus the static equivalence of conjugate triplets induced by some largest known triplet Ψ is defined as follows:

Definition 2.3 *Two triplets Ψ_1 and Ψ_2 are* statically equivalent *if they are static restrictions of some conjugate triplet Ψ.*

The fact that the static equivalence with respect to some *fixed* triplet Ψ is an equivalence relation results from arguments similar with those used for positional equivalence. Once again, *any* given triplet subjected to externally applied loads immediately induces an equivalence class of triplets that are its static restrictions. Given two positional equivalent triplets we can test if one of them is a static restriction of the other by checking whether the conditions on load-bearing boundaries in Definition 2.2 hold. But even if these conditions do not hold, the two triplets may belong to the same positional equivalence class induced by some "larger" triplet.

As already stated, for a given motion M and a moving object A, there may not exist a conjugate triplet (see also [10]). In other words, the input design specifications may be incompatible. Similarly, for a given motion M, known object A, and given loading conditions, there may not exist a triplet that would be functional from a contact point of view. Imagine a low-speed cam-follower mechanism whose shapes maintain contact during operation. Now reverse the spring force[6] applied on the follower: this would tend to take the follower and the cam apart. In this case, the cam has no load-bearing boundaries under these new loading conditions.

The class of statically equivalent parts is a subclass of the positional equivalence class, and this follows directly from definition 2.2: for a given loading condition, only those members of the positional equivalence class that satisfy definition 2.2 will be elements of the static equivalence class. However, the mere existence of the load-bearing contact points will not guarantee the proper functionality of a given triplet which is subjected to some external loads along the known directions. A more complete validation must include strength and motion analysis which is impossible without knowing the actual load magnitudes and profiles. Nevertheless, the importance of the static equivalence to designing moving parts is twofold. First, the static equivalence class contains all shapes that satisfy the necessary conditions of the contact function, without over-constraining the actual shape of the parts. Second, the maximal triplets with classified contact surfaces contain critical design information that relates the form to the intended function, which would otherwise fade away. The load-free boundary points are as important as the load-bearing ones to the shape synthesis, since they can be modified without violating the contact constraints.

3 Synthesis Through Equivalence Classes

3.1 Functionally Equivalent Latches

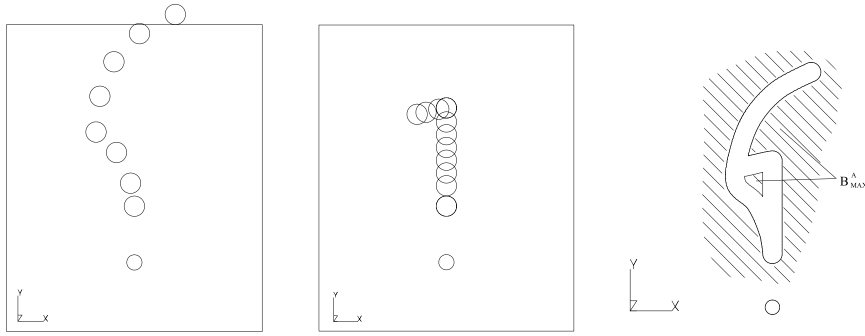
We demonstrate the usefulness of our approach by focusing on the design of the automotive secondary latch illustrated in Figure 1, which is subjected to both containment and contact constraints. The (maximal) solution satisfying the containment constraints is obtained by computing, according to either one of the equations (2), the representative member of the *containment equivalence* class (Figure 8(a)), which is guaranteed to contain all shapes satisfying the same containment constraints. One can show that any functioning latch must be a member of the containment equivalence class.

To solve for the contact constraints in the design problem of the latch we use the conjugate triplet formulation: we are given the shape of the cylindrical striker A and partial information about motion M, and we need to determine the shape of the object B that moves in contact with A. In the initial stages of the design of such a latch, the designer does not have an analytical representation for the relative motion M, but rather has a feeling for how the relative motion between the striker and the latch may look like. We assume that such a partial understanding of the relative motion is properly expressed by an incomplete description of the motion through a set of relative configurations of the striker and the latch to be designed, as shown in Figure 7.

Such a set of relative configurations were interpolated here using the algorithm described in [15], which resulted in a continuous description of the motion of the striker relative to the latch. Different design cases may require different interpolation algorithms producing motions with dissimilar properties: in some cases the smoothness and continuity of the resulting motion can be essential, while in others, the simplicity of the resulting shape may be more important [10].

The corresponding maximal triplet Ψ_A represents the class of all *positionally* equivalent latches B^A_{max} that move in contact with striker A. Once the motion M is known, it is straightforward to compute the maximal shape B^A_{max} according to equations (4). If we require the latch to be a single part, it should be a

[6]The gravity effect on the follower can be neglected for low-speed cam-follower mechanisms.



(a) (b) (c)

Figure 7: The discrete relative configurations of the striker A relative to the latch to be designed: the motion of the striker was separated into its "downward " motion (a) and its "upward" motion (b) because they correspond to different functions of the latch. Figure (c) shows the top view of the B^A_{max} forming the maximal triplet $\langle A^A_{max}, B^A_{max}, M\rangle$ according to equations (4).

---

connected set, and we must be able to maintain the contact with the given striker A using only one of the connected components of B^A_{max} shown in Figure 7(c).

By identifying all the load-bearing points of B^A_{max} according to definition A.1 in appendix A, we generate the class of statically equivalent latches which will contain the functional surfaces of any latch satisfying the same contact function. A vertical external force is exerted upon the striker, which first moves the striker down and then up. The maximal connected shape represents the largest functional shape satisfying the given contact constraints, and is shown in Figures 8(b-c), where the functional surfaces are in lighter color. It contains all other *statically equivalent* shapes satisfying the given contact function. One can also test whether new geometries of the latch satisfy the contact function by performing simple geometric computations described in the appendix A to confirm the existence of load-bearing points for every configuration of the striker's motion relative to the latch.

Any final latch subjected to the same external loads must form a triplet with the striker and their relative motion. This triplet must be *statically equivalent* with the maximal triplet shown in Figures 8(b-c) and be part of the *containment equivalence* class represented by the largest shape in Figure 8(a). The largest such latch satisfying both contact and containment constraints is shown in Figure 8(d) and is obtained by intersecting the shapes shown in Figures 8(a) and 8(b-c). Once the result of the intersection is known, one must confirm that the latch has load-bearing points at every configuration of the relative motion between the striker and latch. On one hand, if the maximal element of the containment class does not intersect the load bearing boundaries, it means that the intersected shape satisfies the contact constraints. On the other hand, if the intersection is non-empty, one either needs to check whether the trimmed contact surfaces are sufficient for the contact constraints, or modify the functional requirements for the problem. Figures 8(e-f) show two other possible functional shapes that are statically equivalent with the maximal latch shown in Figure 8(d).

Refining the geometry of this largest latch must *also* take into account the magnitude of the loads applied on the mechanism, the strength of the latch, the manufacturing constraints and so on, but the functional latch must be part of the same static and containment equivalence classes. The non-functional surfaces of the latch can be changed without violating the contact function to accommodate additional constraints, or shape modifications that result from shape optimization algorithms.

3.2 Composite Classes of Equivalent Designs

Each equivalence class relies upon an equivalence relationship that embeds specific functional information, and all members of a given equivalence class satisfy the same functional specifications. Each of the three

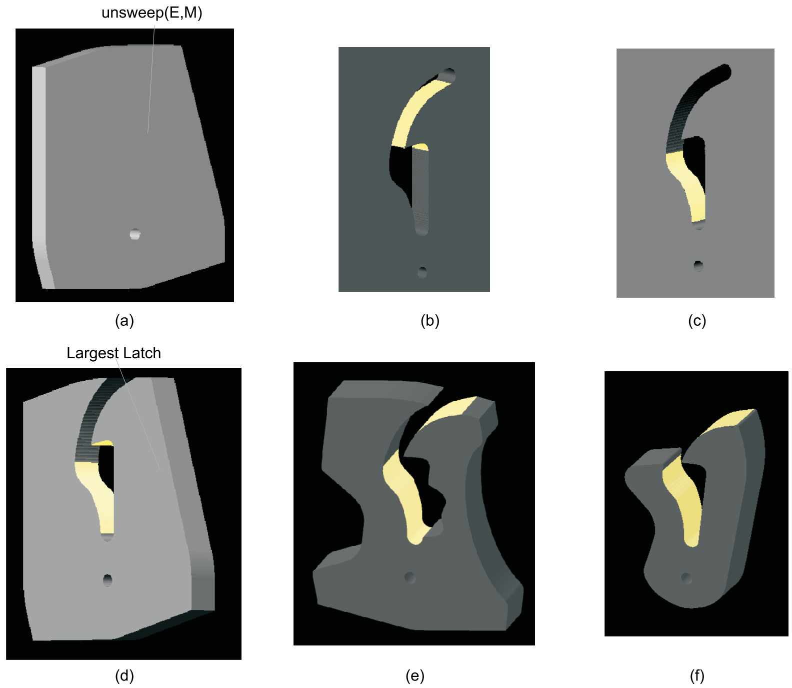


Figure 8: Figure (a) illustrates the largest shape satisfying the containment constraints. Figures (b-c) show the largest unbounded connected shape, which includes all other statically equivalent shapes satisfying the given contact function. The load-bearing boundary points have been computed according to the functional classification detailed in appendix A and are shown in lighter color. Figure (d) displays the largest latch satisfying both contact and containment constraints obtained through an intersection of the two corresponding maximal shapes. Figures (e-f) show two other functional latches that are statically equivalent with the maximal latch shown in (d).

---

classes discussed in this paper possesses a partial ordering of their elements as well as a maximal element within the class. These classes were derived for typical constraints that need to be satisfied by moving objects. Specifically, each moving part needs to satisfy containment constraints, *and* contact constraints, *and* contact constraints under external loading. This implies that any part satisfying all these three functional constraints must be an element of all corresponding equivalence classes, as shown intuitively in Figure 9. In fact, the largest such part satisfying all three functional requirements is obtained by intersecting the maximal elements from each class. As long as each such equivalence class maintains a partial ordering of its elements and includes a maximal element, the largest functional part will be the intersection of the maximal elements from each class.

By replacing the completely specified geometry of the traditional approaches with partial geometry *and* functional specification, one can generate classes of equivalent mechanical parts so that all members of the class satisfy the same functional specifications. Our approach to design through equivalence classes captures the design decisions by generating functional surfaces that do not embed arbitrary geometric restrictions, and identifies fully defined representative members of the equivalence classes that contain an essentially unlimited number of functional designs.

Consistent with common engineering practice, the definition of the static equivalence assumes that the magnitudes of the applied loads are not known while the design problem is being defined, but their directions are well understood. This implies, of course, that the equilibrium conditions cannot be examined at this stage under the current hypotheses. Imagine, for example, a train running on two rails. The train, the rails and the motion of the train, quasi-static by assumption, form a conjugate triplet, and both rails contain

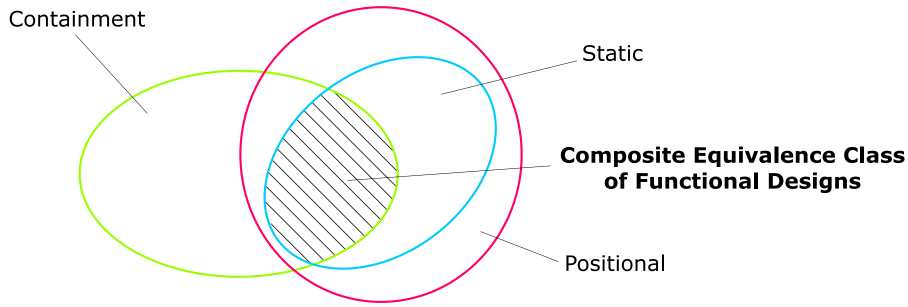


Figure 9: Any functional part must concurrently satisfy all imposed functional requirements, and thus must be an element of all three classes of equivalence.

---

contact surfaces that are load-bearing. If we would remove one of the rails from our problem definition while preserving the motion, we would still have – by definition – a conjugate triplet that will be statically equivalent with the first one since we would still have load-bearing contact point at every configuration during the motion. However, not only that this train may not satisfy the equilibrium conditions, i.e., it is incompatible with the given input, but the equilibrium conditions cannot be examined unless we know both the directions and magnitudes of the loads applied on the train and the rail.

The introduction of known load magnitudes this early in the design process would enable the enforcement of the equilibrium conditions within the class of statically equivalent designs, and would guarantee that any statically equivalent triplet would support the applied loads, at least in a quasi-static sense. However, such an assumption would unnecessarily restrict the design problem, and would diminish the effectiveness and applicability of any design technique, including of the proposed approach.

4 Towards More Rational Conceptual Design

Numerous novel design tools have surfaced during the last decades, tools that primarily take advantage of the added computational power to accommodate new and growing market demands. However, mechanical design has remained conceptually unchanged in the sense that traditional approaches to mechanical design, still widely used today, fully rely on the creativity and engineering intuition of the designers and lead to design procedures that are intensely iterative.

Functional equivalence classes, on the other hand, can play an essential role in design because they capture design decisions without imposing arbitrary restrictions on the geometry of the parts. The traditional full geometric specification is replaced by partial geometry and functional specification, and the corresponding design process is shifted from one leading to a single solution of the design problem, to a process leading to a *class* of functionally viable solutions. It should be clear that functional equivalence cannot be defined generally and uniquely, because this would also imply complete and *unique* characterization of the corresponding mechanical function.

This paper presents the first steps in formulating a theory of design through equivalence classes for mechanical parts. One important and fruitful direction for future research is the investigation of other functionalities that can lead to the establishment of other functional equivalence classes that may be derived and represented computationally. These equivalence classes should lead to the identification and classification of additional categories of functional shapes and geometries. For example, assuming that the motion is quasi-static allowed us to neglect the inertia effects. Taking the kinetic phenomena into account would imply that the functional surfaces would not only depend on the geometry and motion configurations, but also on the first and second order derivatives of the motion as inertia is expected to become an important player. Furthermore, manufacturing constraints imposed by specific manufacturing processes, constraints imposed by the mechanics of materials used in a specific design, or functional robustness of the designs against the stochastic variability in the design variables are all constraints that eliminate (otherwise functional) designs from the equivalence classes described in this paper.

Acknowledgments

This work was supported in part by the National Science Foundation grants DMI-0115133, CCR-0112758 and by the National Institute of Standards grant 60NANB2D0126.

References

[1] Nam P. Suh. *The Principles of Design*. Oxford Series on Advanced Manufacturing, 1990.

[2] Gerhard. Pahl and Wolfgang Beitz. *Engineering design: a systematic approach*. Springer, 1996.

[3] R Gupta and M. Jakiela. Kinematic simulation and shape synthesis via small scale interference detection. *Research in Engineering Design*, 6:103–123, 1994.

[4] T.F. Stahovich. Artificial intelligence for design. In E. K. Antonsson and J. Cagan, editors, *Formal Engineering Design Synthesis*. Cambridge University Press, 2001.

[5] C.-Y. Lee, L. Ma, and E.K. Antonsson. Evolutionary and adaptive synthesis methods. In E. K. Antonsson and J. Cagan, editors, *Formal Engineering Design Synthesis*. Cambridge University Press, 2001.

[6] M.E. Caine. *The Design of Shape From Motion Constraints*. PhD thesis, Massachusetts Institute of Technology, 1993.

[7] L. Joskowicz and S. Addanki. Innovative shape design: A configuration space approach. Technical Report 356, Courant Institute of Mathematical Sciences, New York University, March 1988.

[8] L. Joskowicz and E. Sacks. Computer-aided mechanical design using configuration spaces. *Computing in Science and Engineering*, 1(6):14–21, 1999.

[9] T. F. Stahovich, R. Davis, and H. E. Shrobe. Qualitative rigid-body mechanics. *Artificial Intelligence*, 119(1-2):19–60, 2000.

[10] Horea T. Ilieş. *On Shaping Moving Mechanical Parts*. PhD thesis, University of Wisconsin, Madison, 2000.

[11] H. Ilieş and V. Shapiro. An approach to systematic part design. In *Proceedings of the 5th IFIP WG5.2 Workshop on Geometric Modeling in CAD*, pages 383–392, May 1996.

[12] H. Ilieş and V. Shapiro. The dual of sweep. *Computer Aided Design*, 31(3):185–201, March 1999.

[13] H. Ilieş and V. Shapiro. On the synthesis of functionally equivalent mechanical designs. In *Computational Synthesis: From Basic Building Blocks to High Level Functionality*, AAAI Spring Symposium Series, March 2003.

[14] H. Ilieş and V. Shapiro. A class of forms from function: the case of parts moving in contact. *Research in Engineering Design*, (13):157–166, 2002. A preliminary version of this paper appeared in the *Proceedings of the ASME 2001 DETC,* 13^{th} *International Design Theory and Methodology Conference*.

[15] T. Horsch and B. Jüttler. Cartesian spline interpolation for industrial robots. *Computer Aided Design*, 30(3):207–224, 1998.

[16] Donald L. Greenwood. *Principles of Dynamics*. Prentice-Hall, 1988.

A Appendix: Functional Partitioning of Boundary Points

Without loss of generality, we can consider the applied loads to act[7] at the center of mass of a moving set A, since any number of forces and torques acting on a given rigid object can be reduced to a resultant force and a resultant torque acting along directions passing through the center of mass of the object. Our functional partitioning of the boundary points relies on the duality between contact points and contact forces. This duality is implicitly used in statics, and dynamics, and follows directly from the fundamental symmetries of the equal and opposite directed load pairs of Newton's third law.

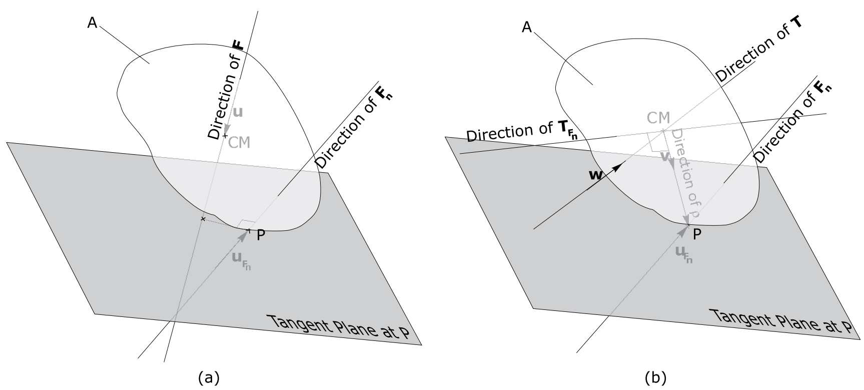


Figure 10: Applied force (a) and torque (b) with components opposite to the normal contact force $\mathbf{F}_n$ at a contact point P.

To determine whether the resultants of the loads applied on moving A generate a positive contact force at a given contact point of B, it is sufficient to investigate whether the component normal to the contact surface of B provides support to A. More precisely, at every point P of the load-bearing boundaries of B, the externally applied loads on A must have a component along the direction normal to the contact surface of B at P, and oriented towards the interior of B at P.

Consider first two forces $\mathbf{F}$ and $\mathbf{F}_n$ as in Figure 10(a). Force $\mathbf{F}_n$ is a normal contact force at a contact point P of a moving object, while $\mathbf{F}$ is the force applied at the center of mass of the object. The fact that $\mathbf{F}$ has a component collinear with and opposite to $\mathbf{F}_n$ can be expressed using the dot product as

$$\mathbf{F} \cdot \mathbf{F}_n < 0 \qquad \textit{or} \qquad \mathbf{u} \cdot \mathbf{u}_{\mathbf{F}_n} < 0 \tag{5}$$

where $\mathbf{u}$ and $\mathbf{u}_{\mathbf{F}_n}$ are unit vectors in the direction of $\mathbf{F}$ and $\mathbf{F}_n$.

Then, given a torque $\mathbf{T}$ that acts at the center of mass of the moving object, and a normal contact force $\mathbf{F}_n$, as in Figure 10(b), we can determine the directions of $\mathbf{T}$ for which $\mathbf{T}$ has a component opposite to the torque generated by $\mathbf{F}_n$ about the center of mass of the object. The torque $\mathbf{T}_{\mathbf{F}_n}$ induced by $\mathbf{F}_n$ can be written in terms of the standard vector cross product as

$$\mathbf{T}_{\mathbf{F}_n} = \mathbf{F}_n \times \boldsymbol{\rho}_P \tag{6}$$

where $\boldsymbol{\rho}_P$ is the position vector of the contact point P with respect to the center of mass of the body. Clearly, $\mathbf{T}_{\mathbf{F}_n}$ is perpendicular to both vectors $\mathbf{F}_n$ and $\boldsymbol{\rho}_P$. Thus torque $\mathbf{T}$ has a component opposite to the torque generated by $\mathbf{F}_n$ about the center of mass of the object iff

$$\mathbf{T} \cdot \mathbf{T}_{\mathbf{F}_n} < 0 \qquad \textit{or} \qquad \mathbf{T} \cdot (\mathbf{F}_n \times \boldsymbol{\rho}_P) < 0 \tag{7}$$

[7]The geometrical interpretation of a vector is a directed line segment whose essential features include magnitude, and direction, but *not* location. Although forces and torques are considered vector quantities, their location in space is not irrelevant in a physical sense. From a physical point of view, vector quantities can be classified as *free*, *sliding*, and *bound* vectors [16]. For the *sliding* vectors, the line of action is critical information, while for the *bound* vectors both line of action and point of application are critical. In our case, all loads are considered to be sliding vectors, i.e., vectors whose line of action is specified.

If $\mathbf{T} = \|\mathbf{T}\|\mathbf{w}$, $\mathbf{F}_n = \|\mathbf{F}_n\|\mathbf{u}_{\mathbf{F}_n}$ and $\boldsymbol{\rho}_P = \|\boldsymbol{\rho}_P\|\mathbf{v}$, where $\mathbf{u}_{\mathbf{F}_n}$, $\mathbf{v}$ and $\mathbf{w}$ are unit vectors, then inequalities (7) can be rewritten as

$$\mathbf{w} \cdot (\mathbf{u}_{\mathbf{F}_n} \times \mathbf{v}) < 0 \tag{8}$$

which shows that the magnitudes of the vectors do not influence the sign of inequality (7). Then,

Definition A.1 *Point $P \in \partial B$ is a* load-bearing contact point *of the boundary of B if either one or both of the following two conditions are satisfied:*

$$\mathbf{u}_{\mathbf{F}_n} \cdot \mathbf{d}_{\mathbf{F}} < 0$$

or

$$\mathbf{d}_{\mathbf{T}} \cdot (\mathbf{u}_{\mathbf{F}_n} \times \boldsymbol{\rho}_P) < 0.$$

Observe that, for every point of the boundary ∂B, definition A.1 gives a test to determine whether a given boundary point is load-bearing, which is easily implementable.