Proc. 1996 IEEE Intl. Conf. on Robotics and Automation, pp. 1837–44

A Framework for Geometric Reasoning About Tools in Assembly*

Randall H. Wilson

Intelligent Systems and Robotics Center
Sandia National Laboratories
Albuquerque, NM 87185-0951

Abstract

When assembling a product, humans, robots, and other automation employ a variety of tools to manipulate, attach, and test parts and subassemblies. This paper proposes a framework to represent and reason about geometric accessibility constraints for a wide variety of assembly tools. Central to the framework is a volume encoding a minimum space that must be free in an assembly state to apply a given tool, and constraints on the placement of that volume relative to the parts on which the tool acts. Determining whether a tool can be applied in a given assembly state is an instance of the FINDPLACE problem [11]. In addition, we present more efficient methods to integrate the framework into assembly planning. For tools that are applied either before or after their target parts are mated, one method preprocesses a single tool application for all possible states of assembly. For tools applied after their target parts are mated, a complementary method guarantees polynomial-time assembly planning. We describe experiments with an initial implementation of the framework and a library of eight tools.

Introduction

Important constraints on assembly plans arise from the need to apply various tools to manipulate, attach, and test parts and subassemblies. Tools in this sense include a wide variety of implements, from screwdrivers and hammers, to welders and bolt drivers, to coordinate measuring machines and even human eyesight. In many cases process engineers find it difficult to determine tool accessibility for assembly operations without costly prototypes. This is especially true in cramped spaces containing geometrically complex parts that may interfere with the operation, such as inside the engine compartment of an automobile.

This paper presents a framework for representing and automatically reasoning about the geometric accessibility of tools in assembly. The approach first classifies tools by whether they are used before, during, or after mating of the parts upon which the tools act. A *use volume* encodes a minimum space that must be free in a subassembly to apply the tool, and *placement constraints* determine where that volume must be placed relative to a canonical reference frame. A particular application of the tool defines which parts the tool acts upon and places its reference frame at the position of required tool use.

Given this representation, a tool can be applied in a given subassembly if and only if a placement of its use volume exists that satisfies the placement constraints and collides with no parts in the subassembly. This is an instance of the FINDPLACE problem [11]. However, a typical assembly planner will make many queries about tool accessibility for a single tool application. For tools that are applied either before or after their target parts are mated, we give polynomial-time algorithms to preprocess a single tool application for all possible states of assembly. Moreover, for tools that are applied after their target parts are mated, we present an extension to previous assembly planning techniques that guarantees polynomial-time assembly planning with tools.

While limited to geometric accessibility issues, the approach provides coverage for a wide variety of assembly tools. We present a number of examples and describe progress toward an implementation of the approach in an experimental assembly planning system.

1 The Problem

An *assembly* is a product consisting of two or more *parts*; a *subassembly* is any nonempty subset of the parts. An *assembly plan* for a product is a sequence of motions and manipulations of the parts that transforms the individual parts into the finished product. Given a complete description of an assembly, *assembly planning* is the problem of determining a feasible assembly plan for it. We consider only *monotone two-handed* plans, which consist of a sequence of *operations*,

*This work was supported by Sandia National Laboratories under DOE contract DE-AC04-94AL85000.

each placing two rigid subassemblies in their final relative positions. While not considered explicitly, our methods also apply to disassembly and service plans.

Let a given use of a tool to affect a particular set of parts be called an *application* of that tool. We assume that each tool application fits within a single operation, i.e. it happens just before, during, or just after a single mating of two subassemblies. We also assume an unordered list of all tool applications required during assembly is given as input to the assembly planner. Then a *tool-level assembly plan* for a product is a sequence of part motions and tool applications that will construct the product from its parts.

This paper presents a framework to answer questions of the form "Is there space for this tool to be used?" There are many related issues that we do not address, such as the space required for a human or robot arm to wield the tool, how to choose the best tool among several that are feasible, finding an optimal tool-level plan, how to design new special-purpose tools, and design changes that might allow a tool to be used. These issues are discussed further in [16].

2 Previous Work

Several assembly planners have addressed problems related to tool constraints. For instance, Homem de Mello and Sanderson's relational model of assemblies includes *attachments*, which correspond to fastening operations [7]. Each attachment specifies the other parts whose presence prevents the parts from being fastened, but determining which parts prevent attachment is not addressed. Henrioud and Bourjault [6] defer to an engineer to determine feasibility of attaching parts, in some cases causing many hundreds of questions to be asked of the engineer.

Reasoning about the effects and use of machine tools is a well-studied problem (see e.g. [13, 14]). But in contrast with assembly tools, machine tools are essentially subtractive in their effect, and the constraints on machining rarely appear in the same form in assembly.

An ambitious system to reason about tools was proposed by Brady *et al.* [3]. The system was to recognize tools from images, determine their uses analogically or from first principles, and even design new tools automatically. However, the project was canceled before significant progress could be made.

Special-purpose planners have been created for certain tools. For instance, in [15] the position and approach path are planned for a coordinate measuring machine. Determining visibility regions for a camera (an assembly tool when used to facilitate or inspect an assembly operation) has been widely studied (e.g. [10]). Miller and Hoffman [12] describe an assembly planner that requires access space above screws, bolts, and nuts before they can be removed. However, the tests used to determine access are only approximate, and it is unclear how they could be used for other tools.

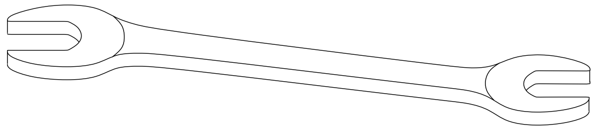


Figure 1: An open-end wrench

Experiments have been performed on the time human workers take to execute screwing, nut tightening, and pop riveting operations under conditions ranging from normal to obstructed access and restricted visibility [2]. Díaz-Calderón *et al* [5] present progress toward automatically determining the difficulty of using a screwdriver in a particular assembly operation.

This paper addresses geometric accessibility issues for a wide variety of assembly tools in a single framework. Our approach is based on now standard configuration-space techniques introduced by Lozano-Pérez [11].

3 Representing Tools

Our representation for tool constraints is divided into *canonical tools* that are independent of any assembly, and *applications* of tools in a particular assembly. A canonical tool is defined by (1) when it is used relative to the mating of parts, (2) a *use volume* that must be free to apply the tool, and (3) *placement constraints* on where the use volume must be located. A list of canonical tools available to a particular system is called a *tool library*. This section describes canonical tools; the next section describes tool applications.

We will illustrate each piece of the framework with a simple open-end wrench, shown in Figure 1. Section 5 gives other examples.

3.1 Relative Time of Application

Tools have very different characteristics depending on when they are applied relative to when parts are mated in the operation:

Pre-tools are applied strictly before the parts are brought together. An example is a glue gun used to apply glue to one part before mating it with another.

In-tools are applied while the parts are moving relative to each other. Examples include a wrench or screwdriver, or a jig used to guide one subassembly.

Post-tools are applied after the parts have been mated. Testing, inspections, and many fastening tools such as welders and riveters are common examples.

Which of these sets a tool belongs to is called the tool's *relative time*.

The criteria to determine whether a tool can be applied in a given operation vary depending on the tool's relative time. Pre-tools need only be feasible to apply to one of the two subassemblies S_1 or S_2 mated in the operation, while post-tools need only be feasible in the resulting subassembly $S = S_1 \cup S_2$. In-tools are the most complex case, since they must be feasible to apply to S_1 and S_2 under a particular relative motion.

Because of the complexity of reasoning about in-tools, it is often desirable to approximate them as post-tools or pre-tools where appropriate. In fact, the wrench example (Figure 1) is one such case. Although the wrench is employed while the bolt is moving, we instead represent tightening the bolt as if it does not move during the process, i.e. as a post-tool. Subsection 3.4 discusses this issue in greater depth.

In some cases a single tool might be usable at different relative times. For instance, a glue gun might be used to apply glue to a part before mating it with another, or after the parts are mated. We consider these two ways of using the same physical tool to be two distinct canonical tools.

3.2 Use Volume

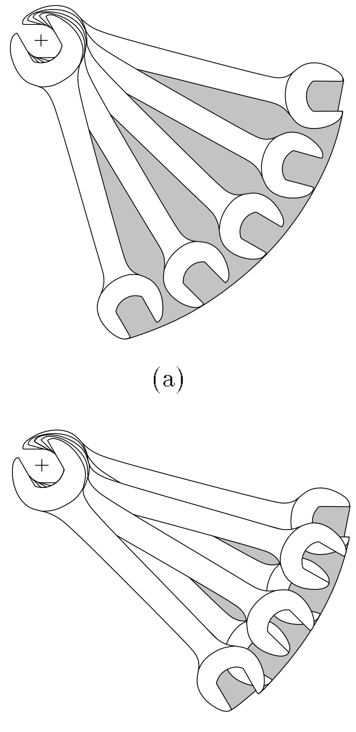
We represent the spatial constraints on applying tools as problems of placing certain volumes in the assembly. Let a tool's *use volume* be a minimum volume that must be free in the assembly for the tool to be applied. For pre- and post-tools, an intersection-free placement of the use volume must exist in a subassembly. For in-tools, the use volume must be able to move rigidly with the inserted subassembly, colliding with no parts of either subassembly.

For some tools, the use volume is simply the spatial extent of the tool itself. For example, to spot weld two parts with a laser welder, only the space occupied by the welder and laser beam must be free of obstructions. However, not all use volumes are so simple.

For many tools the use volume is the space swept out by the tool as it is applied. One use volume for the wrench in Figure 1 is the volume swept out as the wrench turns a bolt $\frac{1}{6}$ turn, is raised off the bolt, returns to the original orientation, and is replaced, ready for another turn. A two-dimensional projection of this use volume is shown in Figure 2a. This is the minimal space that must be free in the assembly for the wrench to tighten a bolt in the standard way.

If more than one possible use volume exists for a tool, we treat each as a distinct canonical tool. For instance, another use volume for the wrench is swept out as it turns the bolt $\frac{1}{12}$ turn, flips over, turns another $\frac{1}{12}$ turn, and returns to its initial position (Figure 2b).

The use volume is usually only a subset of the space required to apply a tool; for instance, the use volume for the wrench does not include the space to bring the tool into the assembly or that required by a robot or human arm. Hence the use volume represents only a necessary condition for tool application. A motion planner could be incorporated to plan approach and removal paths for tool and wielder (e.g. [9]). Alternatively, a larger use volume could include these constraints, becoming a sufficient condition for tool use, but it would overconstrain assembly plans.



(a)

(b)

Figure 2: Some use volumes for the wrench in Figure 1: (a) a $\frac{1}{6}$-turn swept volume, (b) two $\frac{1}{12}$-turns

3.3 Placement Constraints

A tool's *placement constraints* specify the valid positions of the use volume such that the tool can be used effectively. For instance, a screwdriver's tip must mate with the screw head, and its axis must be close to vertical over the screw. The placement constraints are given relative to a canonical reference frame; each tool application locates that frame relative to the parts that the tool acts on.

In the simplest case, the tool use volume must be placed at a completely specified position relative to certain parts of the assembly. For instance, a laser spot weld must be welded by a beam normal to the welded surfaces. The beam is rotationally symmetric, so the position of the laser is fully specified.

However, many tools have some freedom in the placement of their use volumes. The wrench's use volume must be placed such that the center of the wrench jaws aligns with the axis of the bolt, but it can be placed at any angle around that axis. If a tool's placement constraints leave m degrees of freedom in placing the use volume, then we say that the tool is an *m-DOF* tool. For instance, the laser spot welder above is 0-DOF, and the wrench is 1-DOF. Note that tool degrees of freedom refer to the freedom of placing the tool use volume, not to the motion of the tool itself. If the laser welder did not need to be perpendicular to the welded surface, it would be 2-DOF, even though it still would not move during use.

3.4 Choice of Relative Times

There are several situations when tools are represented with a different relative time than the obvious one. For instance, a bolt or nut moves very little while being tightened, and the height of the wrench is almost the same at the start of tightening as at the end. In addition, our representation of in-tools requires the use volume to move with one subassembly throughout the operation, whereas the wrench can disconnect and reconnect with a bolt. For these reasons, we represent the bolt-tightening process as applying the wrench to the bolt in final position, i.e. as a post-tool.

In-tools are the most difficult and expensive tools to reason about, while post-tools are most efficient (see Section 6). Hence even when the in-tool moves rigidly with one subassembly, if the use volume is constant for all mating trajectories (for instance when only one mating trajectory is possible), then a post-tool representation is often better. This is the case for a vertical screwdriver: although it moves rigidly with the screw, the space it requires in the assembly is always the same because the screw always follows the same trajectory.

4 Tool Applications

This section describes a *tool application*, which gives the position and timing of a required tool use in an assembly. A single tool might be required many times in a given assembly; one tool application must be specified for each. Together with its canonical tool, an application gives all the information necessary to determine when the tool can be applied.

4.1 Target Parts

An operation requiring use of a tool is called a *target operation* for the tool. A tool application specifies target operations by means of a *target part set*, which is a subset of the parts of the assembly. A target operation is any operation that brings all of the target parts together in a single subassembly (possibly including other parts) for the first time. To be more precise, an operation mating subassemblies S_1 and S_2 to make a larger subassembly $S = S_1 \cup S_2$ is a target operation for a tool with target part set T if and only if

$$T \subseteq S \bigwedge T \not\subseteq S_1 \bigwedge T \not\subseteq S_2. \qquad (1)$$

This condition is equivalent to Homem de Mello's determination of when an attachment is activated [7] but generalizes to non-attachment situations.

The most common target part set consists of two parts, where the tool is used to fasten the parts together. For example, a target part set for the open-end wrench might be $\{\texttt{bolt}, \texttt{part1}\}$, where the wrench is used to tighten `bolt` into `part1`. Welding, screwing, and gluing operations also apply to two parts in most cases. However, target part sets often include more than two parts, such as a subassembly testing or inspection operation. In most cases parts that are not being tested may be present in the subassembly, as long as they do not interfere with the test.

For pre-tools and in-tools, some additional information is required. For pre-tools, this identifies which of the two subassemblies involved in an operation the tool is applied to. For in-tools, it identifies which subassembly the tool moves with. We implement this as a *primary part* P, which must be a member of T. Whichever subassembly includes P is the identified subassembly for the pre-tool or in-tool.

4.2 Application Frame

The application specifies where the tool must be used by placing the tool's canonical reference frame in the correct position and orientation relative to the assembly. We call this relative position the *application frame*. The placement constraints of the tool can be transformed by the application frame to get the real constraints on the use volume in this application. For example, an application frame for the wrench must place the origin of the canonical frame at the base of the head of the bolt, with the positive z-axis pointing out of the bolt head and parallel with the bolt's axis.

5 Examples

The previous sections have contained some example tools and descriptions of how their accessibility constraints might be represented using the above framework. Here we present additional examples to further illustrate the framework.

Open-End Wrench A wrench can be applied at an angle out of the xy plane to avoid a short obstruction (a 2-DOF use). However, the use volume is not only rotated out of the plane, but changes shape slightly, which cannot be handled exactly in our framework. For small angle use volumes, a rigid volume is an adequate approximation.

Screwdriver The most natural way to use a screwdriver is from directly above the screw. Because the use volume is rotationally symmetric about the z-axis, this is a 0-DOF post-tool. All screwdriver tips (slotted, Phillips, etc.) are handled the same way; a nut- or bolt-driver is the same with a slightly different use volume. When obstructions are present, a screwdriver might be placed a maximum angle from vertical encoded by the placement constraints; this canonical tool is 2-DOF.

Hammer Since a hammer does not move rigidly with the nail, we model it as a post-tool, i.e. we place the nail in final position then strike it after. The use volume is the volume swept as the hammer rises over the nail and strikes. This use volume has one degree of freedom: the side from which the nail is struck. Other hand striking implements are similar.

Resistance Welder A resistance welder is a post-tool that contacts metal pieces between two electric pads. The use volume is just the volume of the welder end tool, and it has one degree of rotational freedom around the spot to be welded. As with all tools, successfully placing this volume does not guarantee that the welder can reach the welding position. A riveter is very similar.

Glue Gun A glue gun used to place a drop of glue on a surface is a 2- or 3-DOF pre-tool, depending on whether the gun is rotationally symmetric. A glue gun or other spreading tool used to place adhesive on a larger area must be approximated by a number of point applications distributed over the surface.

Visual Inspection Inspecting the result of an assembly operation is a post-tool, and the use volume is a 2-DOF line of sight to the inspected point. Robotic cameras often require a view from directly above the inspected point, in which case they are 0-DOF.

CMM A typical coordinate measuring machine has a rotationally symmetric tip that can reach a point from any angle, making it a 2-DOF post-tool. Note that the resulting FINDPLACE computation (see below) resembles the method in [15].

Drill In some cases (such as in aircraft construction) parts are aligned, then holes are drilled for fasteners, making the drill an assembly tool. The use volume is a vertical sweep of the drill and bit, and one degree of freedom around the hole exists for access. A drill can be either a post-tool (applied after the two drilled parts are mated) or a pre-tool (if the fastener is considered a part and is a member of the target part set).

In [16] we present the results of two informal surveys of tool listings to determine the applicability of the above framework to common assembly tools. In summary, the framework was judged adequate to represent accessibility of from 55% to 70% of the tools in general, and from 75% to 85% of mechanical assembly tools. In addition, the vast majority of tools were found to be post-tools, with low degrees of freedom (2 or less) predominating.

The most common tool attributes in the examples above (and also in the surveys) that prevent adequate representation in the current framework are a variable use volume or variable placement constraints. In Section 8 we discuss extensions to the framework that would partially address these limitations.

6 Planning with Tools

This section describes how planning is accomplished using the above tool and application representation. We first show how to test an assembly operation for compliance with a tool constraint, and then present techniques to more efficiently merge tool constraints into the assembly planning process.

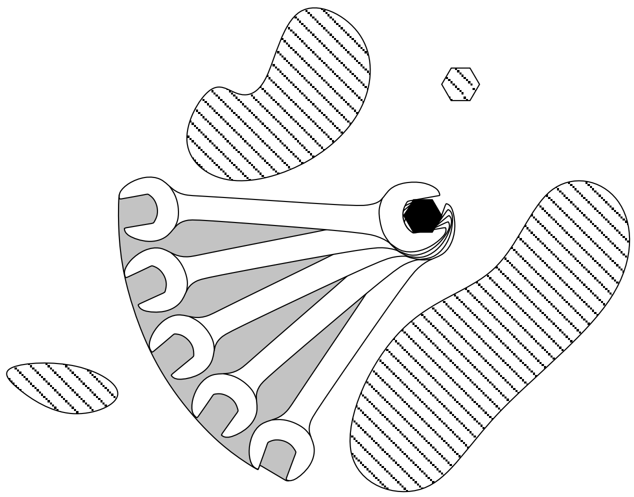


Figure 3: A placement of the $\frac{1}{6}$-turn wrench use volume that allows tightening a bolt (in black)

6.1 Tool Feasibility

Consider an assembly operation mating subassemblies S_1 and S_2 along trajectory t to make a larger subassembly $S = S_1 \cup S_2$, and a tool application with target part set T. If Equation 1 holds, then this is a target operation for the application, and we must determine whether the tool can be applied. We consider pre- and post-tools here; the computation for in-tools is more complicated [16].

A tool can be applied in a given subassembly only if there exists a placement of the tool's use volume that obeys the application's placement constraints and does not intersect with any parts of the subassembly. A post-tool is applied after the subassemblies are mated, so the use volume must be placed in S. A pre-tool's use volume must be placed in S_1 or S_2, whichever includes the application's primary part P. For example, a subassembly and feasible placement for the $\frac{1}{6}$-turn wrench use volume are shown in Figure 3.

Finding a collision-free placement of a use volume U in a subassembly S is a straightforward instance of the FINDPLACE problem [11]. The set of all configurations of U form a six-dimensional space called the *configuration space*, or *C-space*, of U. For each part $P_i \in S$, the *configuration obstacle* $O_U(P_i)$ of P_i is the set of all configurations in which U intersects with P_i. The placement constraints define a subset C of the C-space that satisfies them. A collision-free placement of the use volume exists if and only if the region

$$\text{FREE} = C \setminus \bigcup_{P_i \in S} O_U(P_i)$$

is non-empty, in other words if there exists a configuration of U satisfying the placement constraints but outside all the configuration obstacles.

In a fixed-dimensional C-space, FINDPLACE can be solved in time polynomial in the total number of surfaces describing the parts, use volume, and placement constraints [4], assuming the surfaces are all algebraic of bounded degree. For 0-DOF tools, only one configuration satisfies the placement constraints, and feasibility reduces to an intersection test between the use volume and the parts.

6.2 Preprocessing Tool Applications

In assembly planning, a single tool application might be tested for feasibility in many operations. Here we show how to preprocess each pre- or post-tool application in its assembly, such that every feasibility test thereafter can be answered very quickly.

In a given configuration, a tool's use volume intersects with a subset of parts called an *interference set*. The application is feasible in any subassembly that contains no parts from the interference set. To preprocess a tool application for an assembly, we compute a set of all the interference sets. Then the tool application in a subassembly S is feasible if and only if S has no parts from at least one of the interference sets.

Consider first the 0-DOF case. A 0-DOF tool application has only one configuration that satisfies its placement constraints, so it also has only one interference set, the set of all parts in the assembly that intersect with the use volume in its required placement. The tool can be applied in any subassembly that has no parts in the interference set.

Now consider a use volume U with some placement freedom. A part P_i is in the interference set for a configuration ϕ if and only if $\phi \in O_U(P_i)$. The boundaries of the configuration obstacles of all the parts of the assembly subdivide the C-space of the use volume into cells, and the interference set is the same for all configurations in a cell. Again assuming the surfaces of the parts and use volume are algebraic of bounded degree, the number of cells in this subdivision is polynomial in the number of surfaces, and a representative point can be found in each in polynomial time [1].

Figure 4 illustrates the preprocessing of a 1-DOF wrench application to tighten a bolt (in black), using a $\frac{1}{6}$-turn use volume. A reference ray from the use volume's origin has been attached to indicate placement angles. The one-dimensional set of configurations that satisfy the placement constraints is mapped onto a circle. The cells of the subdivision in this case are intervals of the circle, and the interference set for each interval is shown. The ends of the intervals are angles of the reference ray where the use volume either starts or stops intersecting with a part. The use volume is shown at one such angle. The wrench can tighten the bolt in any subassembly that is missing at least one of the interference sets.

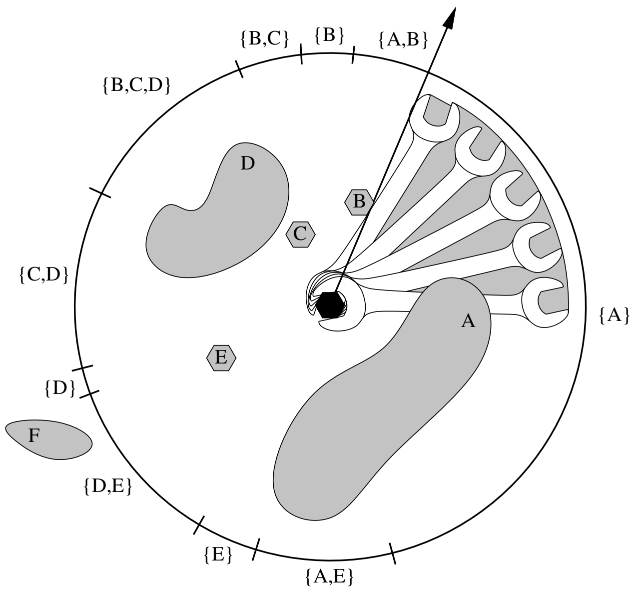


Figure 4: Preprocessing a wrench application: the subdivision of the circle of use volume placements, with corresponding interference sets

The set of interference sets can often be simplified. If one interference set I_1 is a subset of another set I_2, then I_2 can be removed, because any subassembly that does not intersect with I_2 also does not intersect with I_1. For instance, the interference sets from Figure 4 can be simplified to $\{\{A\}, \{B\}, \{D\}, \{E\}\}$.

6.3 Partitioning with Post-Tools

Because there may be a very large number of operations to test, the tool-feasibility tests described above are not an efficient way to *generate* feasible operations. Here we describe how to extend the NDBG approach in [17] to achieve polynomial-time assembly partitioning with post-tools. In [16] we prove that repeated partitioning is guaranteed to generate a valid assembly plan in polynomial time.

Let $A = \{P_1, \ldots, P_n\}$ be an assembly of polyhedra. A *blocking graph* of A for a trajectory t is a directed graph with a node for each part of A and an arc from P_i to P_j if and only if P_i will collide with P_j when moved along t. If a subassembly S_1 can be removed along t exactly when no arcs connect parts in S_1 to parts in $S_2 = A \setminus S_1$. Such a subassembly exists if and only if the blocking graph for t is not strongly connected. The *non-directional blocking graph*, or *NDBG*, of A is a subdivision of the space of trajectories t into cells such that all trajectories within a cell have the same blocking graph. This space is of finite dimension for certain types of motions [17]. By checking the strong connectedness of the blocking graphs for all cells, a removable subassembly can be found (or failure returned if none exists) in polynomial time.

Now assume we have a list of post-tool applications required to assemble A, and we want to find a feasible operation that mates two subassemblies to make A, or determine that none exists. A post-tool application with target part set T imposes the following constraint on an operation O that mates subassemblies S_1 and S_2 to make A:

> "If T is not a subset of either S_1 or S_2, then the use volume must be placed in A."

Conversely, if the use volume cannot be placed in A, then any subassembly removed from A must include all or none of the parts in T.

To keep T together in a blocking graph, we add bidirectional arcs between every pair of parts in T. If many post-tool applications are infeasible in A, we do the same for the target part set of each. Bidirectional arcs between a pair of parts place those parts in the same strong component of the blocking graph. We call this the *augmented* blocking graph of t; its arcs are a superset of the arcs of the blocking graph of t, and an NDBG with augmented blocking graphs is a *post-tool NDBG*. The standard NDBG algorithm applied to a post-tool NDBG is correct and complete for assembly partitioning with post-tool constraints. In other words, it will produce a removable subassembly that also satisfies the post-tool constraints, or correctly report that one does not exist. See [16] for the proof.

7 Implementation and Experiments

We have begun implementing geometric tool constraints in the Archimedes 2 assembly planning system [8], and have constructed an initial library of eight 0-DOF tools: three screwdrivers (slotted, Phillips, and long-shafted slotted), a nutdriver, a hex L-wrench, a laser spot welder, a subassembly test tool, and needle-nose pliers. All are post-tools except the pliers, which are an in-tool. Tool applications are read from an auxiliary text file for the product.

Only 0-DOF tool feasibility has been implemented to date, using the ACIS® solid modeling kernel to test intersections, along with the methods of Sections 6.2 and 6.3 for efficiency.

Tool constraints have been tested on three assemblies, including the 42-part discriminator shown in Figure 5. Its assembly requires 55 laser spot welds, 8 applications of a Phillips screwdriver, 4 applications of a hex L-wrench, 4 subassembly tests, and one use of the pliers. Archimedes finds a tool-level assembly plan for the discriminator in 50 seconds on an SGI 100Mhz R4000 Indigo II Extreme workstation. Figure 5 shows screen dumps from the animated output of the resulting tool-level assembly plan.

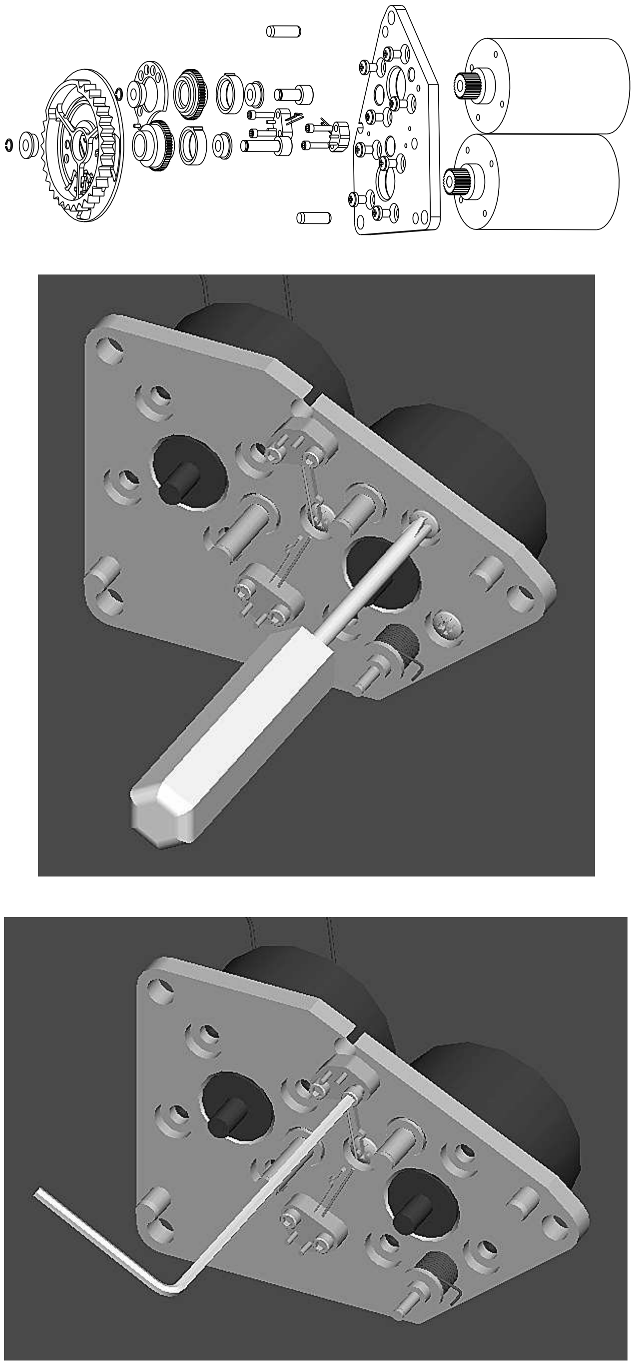


Figure 5: The discriminator: an exploded view and applications of a Phillips screwdriver and a hex L-wrench

8 Discussion

The framework for tool constraints presented in this paper can represent accessibility constraints for a wide variety of assembly tools. We have presented methods that preprocess a single tool application for all possible assembly states, and for a large subset of tools, perform assembly planning in worst-case polynomial time.

The framework as described above supports multiple simultaneous tool applications, such as when one wrench is required to turn a nut and another to hold the bolt. This is accomplished by multiple tool applications with the same target part set. However, the current framework makes the simplifying assumption that the tools will not collide with each other, which is not always correct.

A straightforward extension allows selection of a tool from a set of possible tools to execute an assembly operation; an example is a bolt that must be tightened by a ratchet, an open-end wrench, or a box wrench. Each tool application simply specifies a list of possible tools to execute the action.

One of the most common reasons tools cannot be represented is variations in placement constraints or use volumes. A pipe wrench is a good example of the former. Our representation requires placement constraints to be constant for each tool, whereas the pipe's length (and hence where the use volume may be placed) varies with the application. One solution would allow parameterized placement constraints. For instance, the pipe wrench might have a single parameter z_1, and allow the use volume to be placed anywhere from $z = 0$ to $z = z_1$. The application would set parameter z_1 to the length of the pipe. However, this extension would not handle more complex placement constraints, such as the stability conditions for a pair of pliers used to manipulate a part.

Similarly, some use volumes can be parameterized. For instance, a pair of calipers occupy volume that depends on the measured dimension. The calipers consist of two rigid bodies whose relative position can be parameterized by a distance d between the caliper points. The application gives the value of the parameter, from which a rigid use volume is calculated.

Finally, most tools have few degrees of freedom in a six-dimensional configuration space. High-dimensional FINDPLACE algorithms are slow and hard to implement accurately. Instead, we must find practical and efficient methods that exploit the low real dimension of the solution space.

Acknowledgments

I thank Randy Brost, Jean-Claude Latombe, and David Strip for thought-provoking discussions.

References

[1] S. Basu, R. Pollack, and M.-F. Roy. A new algorithm to find a point in every cell defined by a family of polynomials. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*. Springer-Verlag. To appear.

[2] G. Boothroyd, P. Dewhurst, and W. Knight. *Product Design for Manufacture and Assembly.* Marcel Dekker, 1994.

[3] M. Brady, P. E. Agre, D. J. Braunegg, and J. H. Connell. The mechanic's mate. In T. O'Shea, editor, *Advances in Artificial Intelligence.* Elsevier, 1985.

[4] J. F. Canny. *The Complexity of Robot Motion Planning.* MIT Press, 1988.

[5] A. Díaz-Calderón, D. Navin-Chandra, and P. K. Khosla. Measuring the difficulty of assembly tasks from tool access information. In *Proc. IEEE Intl. Symp. on Assembly and Task Planning*, pages 87–93, 1995.

[6] J. M. Henrioud and A. Bourjault. LEGA: a computer-aided generator of assembly plans. In L. S. Homem de Mello and S. Lee, editors, *Computer-Aided Mechanical Assembly Planning*, pages 191–215. Kluwer, 1991.

[7] L. S. Homem de Mello and A. C. Sanderson. A correct and complete algorithm for the generation of mechanical assembly sequences. *IEEE Trans. on Robotics and Automation*, 7(2):228–240, 1991.

[8] S. G. Kaufman, R. H. Wilson, R. E. Jones, T. L. Calton, and A. L. Ames. The Archimedes 2 mechanical assembly planning system. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, 1996.

[9] Y. Koga, K. Kondo, J. Kuffner, and J.-C. Latombe. Planning motions with intentions. In *Proc. SIGGRAPH*, pages 395–408, 1994.

[10] D. J. Kriegman and J. Ponce. Computing exact aspect graphs of curved objects: Solids of revolution. *Intl. J. of Computer Vision*, 5(2):119–135, 1990.

[11] T. Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, C-32(2):108–120, 1983.

[12] J. M. Miller and R. L. Hoffman. Automatic assembly planning with fasteners. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 69–74, 1989.

[13] R. Narang and G. Fisher. Development of a framework to automate process planning functions and to determine machining parameters. *Intl. J. of Production Research*, 31(8):1921–42, 1993.

[14] A. Requicha and J. Vandenbrande. Automatic process planning and part programming. Technical Report 217, Inst. for Robotics and Intelligent Systems, 1987.

[15] A. Spyridi and A. Requicha. Accessibility analysis for the automatic inspection of mechanical parts by coordinate measuring machines. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 1284–1289, 1990.

[16] R. H. Wilson. Geometric reasoning about assembly tools. Technical report, Sandia National Labs, 1996. In preparation.

[17] R. H. Wilson and J.-C. Latombe. Geometric reasoning about mechanical assembly. *Artificial Intelligence*, 71(2):371–396, 1994.