Submitted to the 1996 IEEE International Conference on Robotics and Automation.

SAND95-2411C
CONF-960448--6

Automatic Design of 3-d Fixtures and Assembly Pallets

Randy C. Brost* and Ralph R. Peters*
Sandia National Laboratories
Albuquerque, NM 87185-5800

This paper presents an implemented algorithm that automatically designs fixtures and assembly pallets to hold three-dimensional parts. The designed fixtures rigidly constrain and locate the part, obey task constraints, are robust to part shape variations, are easy to load, and are economical to produce. The algorithm is guaranteed to find the global optimum solution that satisfies these and other pragmatic conditions. We present the results of the algorithm applied to several practical manufacturing problems. For these complex problems the algorithm typically returns initial high-quality fixture designs in less than two minutes, and identifies the global optimum design in just over an hour.

1 Introduction

Fixture design is a practical problem. When manufacturing products, it is often necessary to hold a part in place during the course of several manufacturing tasks, such as machining, assembly, or inspection operations. The fixtures used to hold the part must prevent undesired part motions and avoid interfering with these tasks, often with the additional requirement that the part must be held in an accurate, repeatable position. These conditions must be maintained even in the face of small variations in part shape that inevitably occur in real manufacturing operations. For process efficiency, the fixture must also be easy to load and unload. In addition to these technical considerations, the fixture must perform well in the economic context of the surrounding business enterprise, implying that the fixture must be inexpensive to fabricate and provide flexibility appropriate to the manufacturing operation.

In this paper, we present an implemented algorithm that automatically designs optimal fixtures for a particular class of fixture problems. The resulting fixtures provide rigid constraint and deterministic location of the part, obey all associated task constraints, are robust in the face of part shape variations, are easy to load and unload, and are economical to produce.

All fixture designs returned by the algorithm are comprised of a few basic fixturing elements. These include round lateral locators, a side clamp, cylindrical support pads, and swing-arm top clamps. Locating and clamping elements in this class are widely available.

These elements are used by the algorithm to design fixtures that hold the part in kinematic form closure; that is, part motion is only possible through deformation of either the part or the fixture. Thus, the returned fixtures do not rely on friction to prevent part motion. Form closure is assured by using the supports and top clamps to prevent motion out of the xy-plane, and by employing the round lateral locators and side clamp to prevent motion within the xy-plane. Further, the algorithm only places top clamps directly above support points to avoid clamp-induced part deformations.

Given a fixturing problem specified by a part description and a set of task constraints, the algorithm enumerates all of the feasible fixture designs that provide form closure while obeying the constraints. These fixtures are then passed to a quality metric which rates each fixture design. Our quality metric considers the fixture's ability to resist expected applied forces without exerting large reaction forces on the part, the location repeatability of critical part features, and the ease of loading the fixture.

The algorithm displays fixture designs as they are generated, allowing the user to interactively study the designs that have been generated thus far. Once all designs are generated, the algorithm sorts the designs according to their quality scores and returns the resulting sorted list. This allows the user to either accept a high-quality design that appears early in the computation, or let the computation run to completion and obtain the global optimum solution.

This algorithm is both complete and practical. If a fixture design exists that solves a given problem using the available fixture elements, the algorithm is guaranteed to find it. Further, the algorithm efficiently finds the global optimum solution by employing pruning methods that greatly reduce the required search. Finally, we employ heuristics that cause high-quality designs to appear early, thus allowing fast identification of high-quality suboptimal solutions. The sections that follow show how this algorithm may be used to solve a variety of manufacturing problems.

2 Case Studies

2.1 Final Machining of Complex Parts

Near-net-shape fabrication methods are techniques for efficiently producing parts with complex shapes. Examples include using casting or welding to produce parts that would be very costly to machine from raw stock. Final machining operations are then performed to create precise part features such as gasket surfaces, threaded holes, etc. These machining operations require fixtures that can hold the complicated part while avoiding interference with cutting paths.

*Supported by US DOE contract DE-AC04-94AL85000.

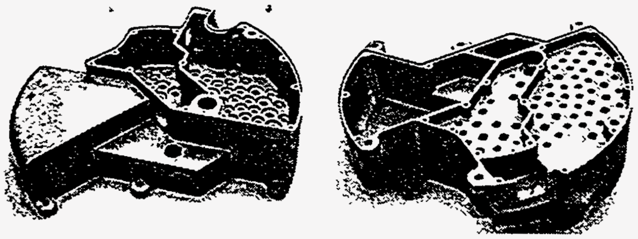


Figure 1: A cast housing that requires finish machining.

Figure 1 shows a part that is manufactured by casting a near-net-shape part, and then applying finish machining operations. These include drilling several holes and milling two gasket surfaces. In the following paragraphs, we show how the algorithm may be used to design the required fixtures for both the prototype and production phases of product development.

Prototype Fabrication

In the prototype phase of product development, a small number of copies of the part are required for initial testing. These prototype parts are typically fabricated using manual casting and machining methods. Because of the small production quantities, fixture design and fabrication costs can comprise a significant portion of the total prototype production cost.

Fixture fabrication costs may be reduced by constructing the fixture from re-usable modular elements. Modular fixture systems are available from a number of commercial firms worldwide. These systems generally fall into two categories: hole-based and slot-based. We focus on hole-based systems here because they have higher precision than slot-based systems, and because they can be assembled more quickly.

The modular fixture kits that we consider have four basic elements: A base plate, cylindrical side locators, cylindrical support pads, and swing-arm top clamps. The base plate has a grid of holes which allow placement of these elements; in some cases, side locators may only be attached to holes on an alternating grid that are equipped with precisely machined bushings. Fixture elements may be set at various heights using vertical spacers; shims allow support pad heights to be set precisely without tedious manual adjustment. The resulting precision in vertical and lateral locating surfaces can be as good as ±0.0006" (0.015mm), depending on the number of spacers used.

This fixture kit allows a machinist to fabricate fixtures quickly and precisely. But the fixture design problem still remains: Given a part such as the one shown in Figure 1 and a set of required machining operations, how should these elements be arranged to produce a good fixture?

The algorithm reported in this paper allows the machinist to identify the global optimum fixture design using these elements, or a high-quality fixture design that is not the global optimum in less time. The algorithm accepts a CAD description of the part, and a description of the machining operations to be performed expressed as geometric volumes swept by the cutter and a set of expected cutting forces. The algorithm also accepts a description of the possible variation in the part shape, as well as tolerance requirements on the location of critical part features.

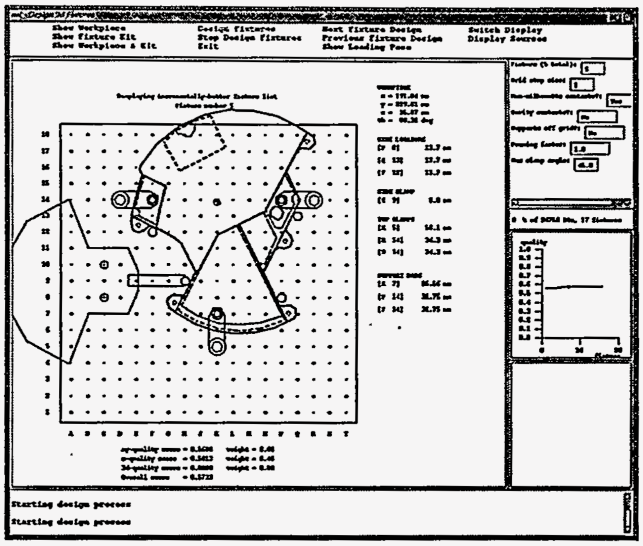


Figure 2: Fixture design for machining a prototype cast housing, shown with the design tool user interface.

Given this information, the algorithm begins generating fixture designs. Each generated design holds the part in 6-dof form closure, avoids cutting paths, and may be assembled using the available fixture elements. Further, each fixture design is robust to part shape variations in the sense that shape variations will not corrupt form closure or cause inadvertant contact between the part and a non-locating surface. Each generated design also passes all of the user-specified thresholds regarding contact reaction force, position repeatability and ease of loading. Fixtures are displayed as they are generated, along with a plot that shows their evolving quality scores. The machinist may let the program run to completion to find the global optimum fixture, or utilize a high-quality fixture that appears early in the computation.

Figure 2 shows a fixture design that appeared after 102 seconds of computation on an SGI workstation. This fixture's quality score is 0.57, which is within 10% of the global optimum score. The early appearance of high-quality fixture designs is typical, partially because of heuristics we employ to sequence the search. Figure 3 shows the physical fixture, which requires roughly five minutes to assemble and load. Thus the total time required to go from problem specfication to ready-to-use physical fixture was less than ten minutes.

Mass Production

Prototype production is characterized by small production volumes and labor-intensive manufacturing

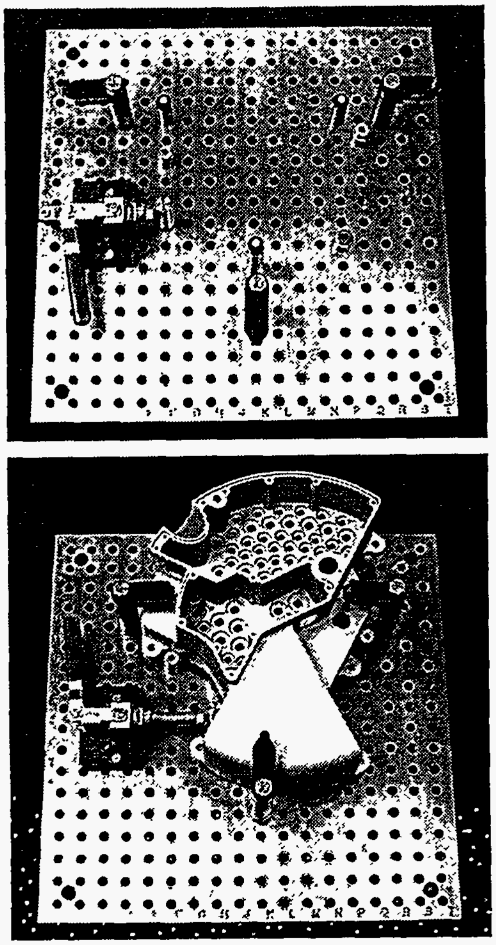


Figure 3: Prototype fabrication fixture.

methods. Thus, reducing fixture design and fabrication time are significant concerns. In mass production, automated manufacturing methods are often employed to improve productivity and process consistency.

The fixture design algorithm may be used to address this problem as well. In this scenario, the user would provide the same problem definition to the algorithm, but would configure the algorithm to produce solutions suitable for automation rather than manual use. In addition to replacing the manual clamps with automated clamps, there are several differences between the manual and automated scenarios:

First, it is reasonable to run the computation to completion to identify the fixture that is the global optimum, since a design is not immediately required in a few minutes.

Second, the fixture may be constructed using plain tooling plate instead of a modular base plate. The base plate is typically the most expensive part of a modular fixture kit, primarily because of its large number of precisely machined holes. Fabricating the fixture from plain tooling plate also allows fixture elements to be placed at arbitrary positions; this provides the opportunity to both increase fixture quality and decrease computation time, as we shall see in Section 4.

Third, a more thorough loading analysis is performed. Because the part will be loaded by an automatic manipulator with limited motion accuracy, the algorithm applies a more sophisticated loading analysis. The algorithm checks several conditions required by a typical loading strategy: For each fixture, the algorithm calculates a loading position that clears the lateral locators by a specified horizontal distance, and verifies that the part may be lifted out of the fixture while clearing all other fixture elements by at least this distance when the clamps are open. The algorithm also verifies that the center of mass falls inside the support triangle in both the loading position and in the final loaded position. Finally, the algorithm verifies that the top-clamp arms may swing to the closed position without interference. Fixture designs that fail any of these tests are rejected.

Fixture designs that pass all these tests may still fail to load easily because friction may stop the part prematurely when the side clamp closes, or because a non-linear part motion may cause the center of mass to exit the support triangle, allowing the part to tip. Our fixture design algorithm does not check for these conditions, although we hope to add this capability in future work. In the meantime, empirical testing may be used to verify robust loading.

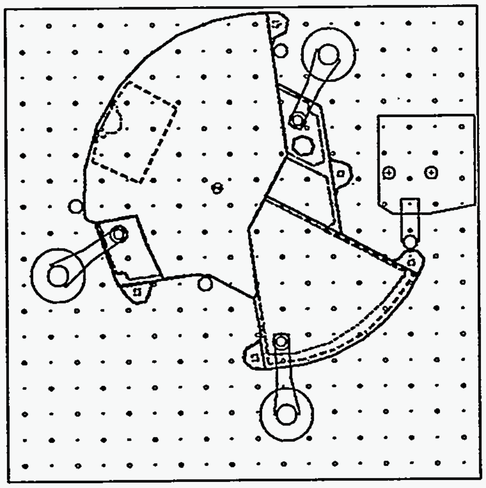
Figure 4 shows the global optimum fixture and its associated loading position. There are over 480,000 feasible solutions to this problem, but the algorithm was able to apply branch-and-bound pruning constraints to find the global optimum while exploring less than 1% of the total search space. The entire computation was performed in 66 minutes; the first fixture design appeared in 40 seconds. Figure 4 shows the physical fixture; robust loading and unloading of this fixture was verified with a robot manipulator over repeated experimental trials.

2.2 Light Mechanical Assembly

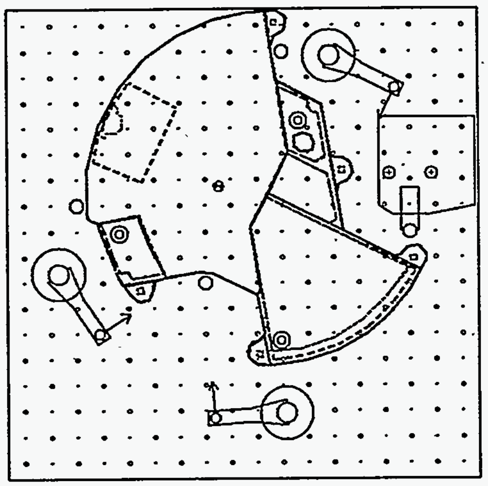
The preceding examples have addressed the design of fixtures for finish machining of parts with complex shapes. Another manufacturing process that commonly requires fixtures is mechanical assembly.

Product assembly problems vary widely; here we focus on assemblies that are characterized by a single base part to which a number of smaller parts and subassemblies are attached. These assembly tasks require a fixture to hold the base part without interfering with any of the assembly operations.

Products of this type are often designed so that parts may be added from a single direction, allowing the assembly to be oriented so that insertions may be performed vertically. It is desirable that the assembly fixtures also be loaded and unloaded by vertical motions.



Loaded



Loading Analysis

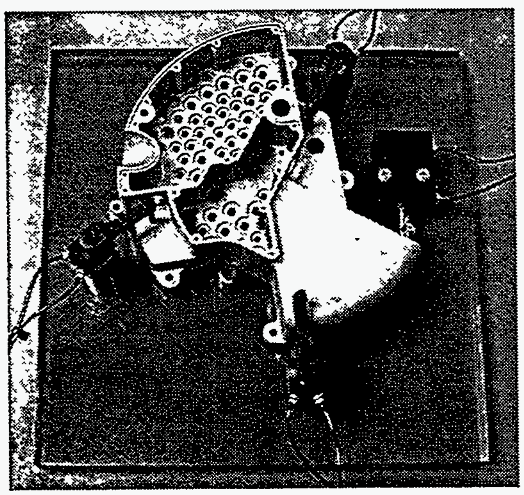


Figure 4: Optimal fixture design for mass-production of the housing. The arm closing directions were automatically selected to avoid collisions.

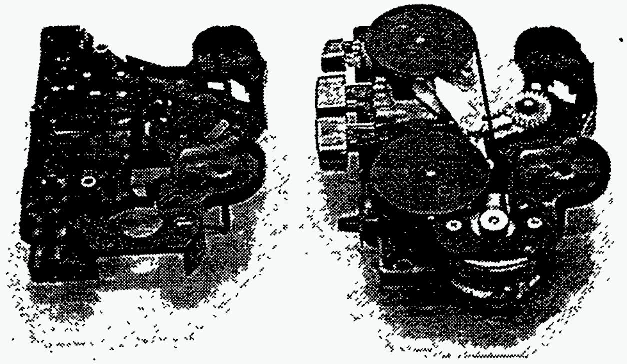


Figure 5: The chassis of a personal cassette player, before and after assembly operations.

Assembly Pallet Design

The cost of fabricating assembly fixtures is a primary concern, since assembly lines often require hundreds of copies of these fixtures to carry the assembly from station to station. Thus assembly fixtures — often called pallets — must be inexpensive in order to be cost-effective.

In order to reduce pallet fabrication cost and allow vertical part loading and unloading, we consider very simple pallet designs comprised of a collection of pins attached to a base plate. Pins have either flat or conical tips; the pins with flat tips support the base part, while the pins with conical tips provide lateral constraint and guide the part into place during vertical part loading. Each pallet has four lateral constraint pins, and three or more support pins.

The fixture design algorithm may be used to design these pallets. The algorithm accepts a CAD description of the base part, along with volumes swept by parts and tooling during insertion operations. The algorithm also accepts a description of the forces that will be exerted during assembly and pallet transfer, as well as tolerances on the location of critical features. Given this input, the algorithm enumerates all feasible pallet designs that hold the base part in planar form closure while avoiding the volumes swept during insertion operations and obeying the input position tolerances.

Since a tripod of support pins may not be adequate to prevent the object from tipping during downward insertions near the perimeter of the part, the algorithm constructs the convex hull of all possible support points and places supports at the convex-hull vertices, after eliminating vertices that only add a small amount to the support area.

The quality metric for assembly pallets is the same as for fixtures with top clamps, except that out-of-plane forces are treated differently. Instead of calculating the contact reaction forces required to resist out-of-plane applied forces, the algorithm calculates the tipping moment exerted by downward forces applied outside the support region or near its boundary, and com-

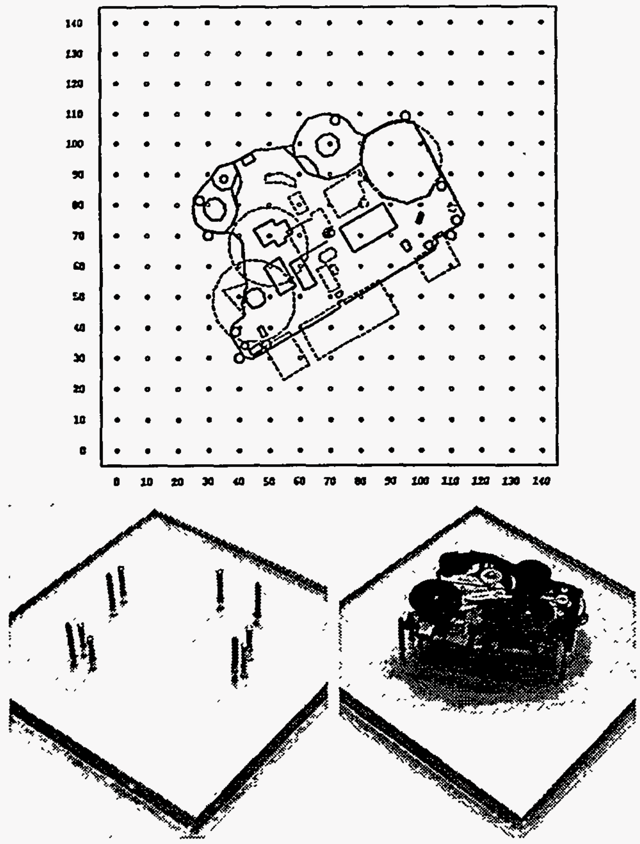


Figure 6: Optimal assembly pallet design for the cassette player. Additional support for the motor during assembly should be designed manually.

pares this moment with the opposing moment exerted by the weight of the subassembly. The difference between these moments is used to score the pallet's ability to resist out-of-plane assembly forces.

Figure 5 shows an example cassette player chassis, which is assembled using vertical insertions. There are several protrusions from the chassis bottom and parts that overhang the chassis perimeter; these complicate the pallet design problem by reducing the available contact surfaces.

Figure 6 shows the global optimum fixture for this assembly. The total computation time was 21 minutes; the first design appeared in 1.2 minutes. Robust vertical loading and unloading was verified using a robot manipulator over repeated trials.

Mixed-Product Assembly

The preceding example shows how the fixture design algorithm may be used to design a pallet for the assembly of a single product. Some manufacturing scenarios require the assembly of more than one product on a single assembly line. An example is a company that manufactures a family of products, each of which is slightly different.

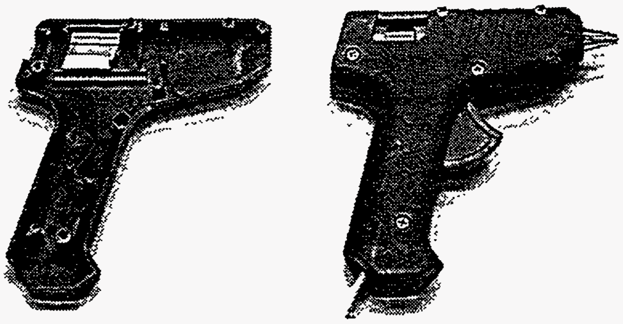


Figure 7: A second product to manufacture in tandem with the product in Figure 5.

A more severe scenario occurs when the products are dissimilar, but a single manufacturing line is desired because it is difficult to anticipate the market demand for each product. In this case rapid switching between products is desired to allow the manufacturer to adapt to changing market conditions. For example, suppose a company wishes to produce both products shown in Figures 5 and 7, in variable quantities. A cost-effective assembly system is required that can switch between products with minimal overhead.

The design algorithm may be used to design an assembly pallet capable of holding either assembly. This is accomplished by generating the possible pallet designs for each problem individually, and then looking for pairs of designs that may be merged into a single pallet with a minimal number of pins. Figures 8 and 9 show an example of such a mixed-product pallet. This pallet may be used to assemble either product shown in Figures 5 and 7 with no required tooling changes. This design reduces pallet fabrication costs by requiring only 13 pins and 112 cm^2 of pallet space, instead of the 16 pins and 175 cm^2 required by a pair of single-part pallets. Further, this pallet allows the manufacturer to switch between products in zero changeover time, at least as far as pallets are concerned.

The time required to design this optimal mixed-part pallet was 140 minutes, including the generation of all solutions to each individual problem. There were 225 pallet designs for the cassette chassis, and 2,991 designs for the glue gun. Of the 672,975 possible pallet pairs, there were 2 high-quality designs that shared three common pins, and over 2,000 high-quality designs that shared two common pins. Pallets of this type appear to be very difficult for humans to design manually.

2.3 Remarks

The above examples are all based on proven hardware that is commercially available and in use today in real manufacturing production facilities. This reflects our approach to studying the fixture design problem: Begin with viable hardware solutions observed in practice, and develop a design algorithm that matches the capabilities and limitations of this hardware.

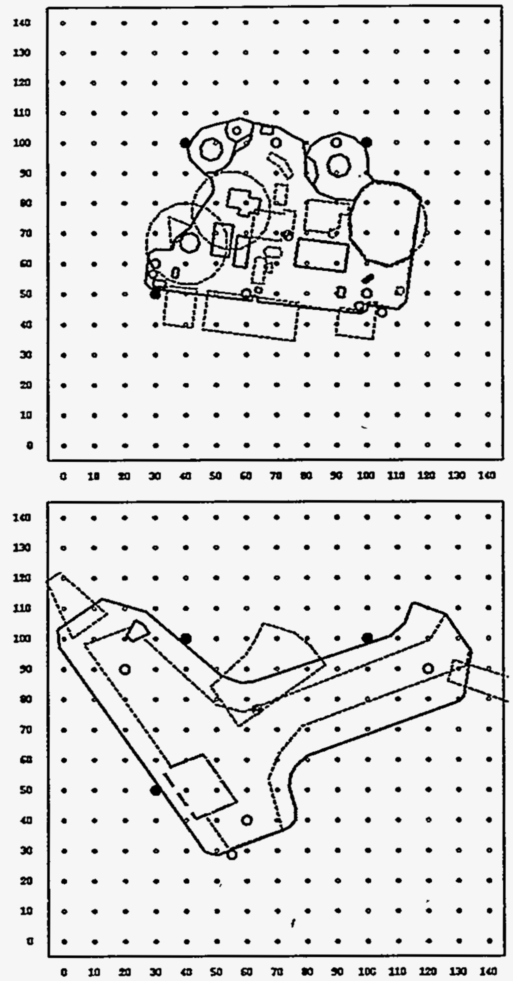


Figure 8: Optimal mixed-part pallet design for assembling the cassette player or glue gun. Pins common to both pallets are darkened for clarity.

Despite the differences between the various manufacturing scenarios presented above, all of the associated fixture design problems submit to a generic underlying problem formulation. This problem is characterized by a part to constrain, geometric regions that must not be violated, a suite of fixture elements with associated placement constraints, and quality criteria that depend on the particular problem instance. The following sections will give a precise definition of this generic problem, and sketch the design algorithm used to solve it.

3 Problem Statement

The primary assumption we make is that the part is a rigid body. A second assumption is that the style of fixtures considered by the algorithm is sufficient for producing an acceptable solution to the given fixturing problem. We will discuss the implications of these assumptions in Section 6.

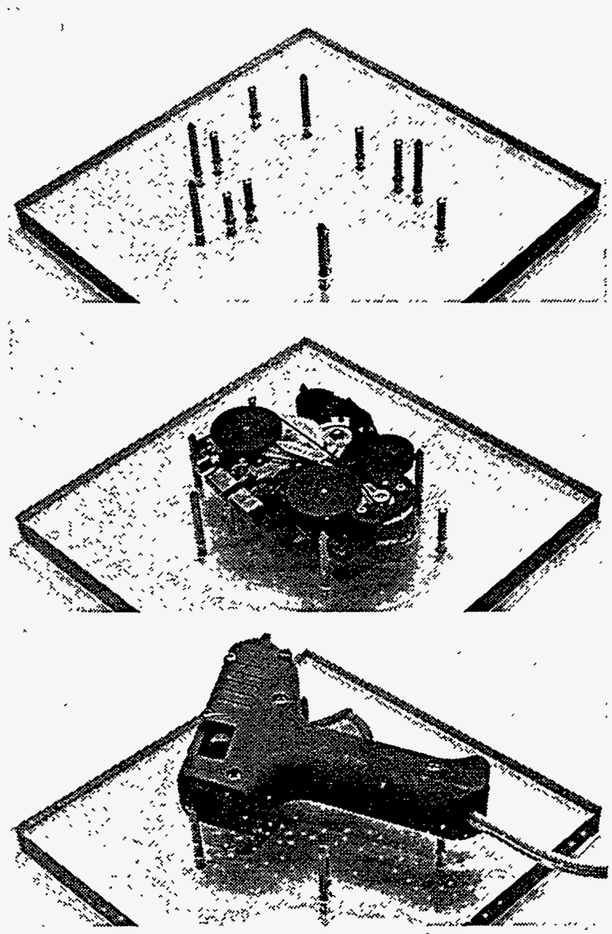


Figure 9: Optimal mixed-part assembly pallet.

Under these assumptions, the fixture design algorithm accepts the following input:

- Part $\mathcal{P}$, represented in a CAD representation that supports identification of vertical and horizontal surfaces, as well as interference queries with cylinders and prisms aligned with the z-axis. Features of $\mathcal{P}$ have associated shape and surface normal tolerances of ϵ_{shape} and ϵ_{normal}.
- Geometric constraint volumes $\mathcal{C}$ defining regions that must remain clear of fixture elements.
- A fixture kit, comprised of a base plate, side locator, support pad, side clamp, and possibly a top clamp. These components have associated shape descriptions, and placement constraints.
- Minimum horizontal and vertical clearances d_{xy} and d_z, and a horizontal loading clearance d_{load}.
- A quality metric comprised of three functions $\mathcal{Q}_{xy}$, $\mathcal{Q}_z$, and $\mathcal{Q}_{3d}$, which accept a planar fixture, out-of-plane fixture, and total fixture, with associated weights w_{xy}, w_z, and w_{3d}. These functions return scalars in $[0, 1]$, or $\emptyset$ indicating that the fixture should be rejected. If no function returns $\emptyset$ during

the evolution of a fixture design, then the resulting overall quality score is $q_{xy}w_{xy} + q_z w_z + q_{3d}w_{3d}$, where q_{xy}, q_z, and q_{3d} are the results from each quality function.

Given this input, the fixture design algorithm generates a stream of fixture designs with associated quality scores. Each generated fixture satisfies all of the following conditions:

1. The part is held in 6-dof form closure, or planar form closure and near-maximal z-support if top clamps are absent.
2. The form closure condition is robust in the face of local part shape variations. That is, for any choice of surface normal errors in $\pm\epsilon_{\text{normal}}$, form closure is preserved.
3. Part location is deterministic, in the sense of [1].
4. No part of the fixture interferes with $\mathcal{P}$ or $\mathcal{C}$.
5. All top clamps are directly above support pads.
6. The fixture is feasible to fabricate or assemble.
7. Except for intended contacts, all fixture elements clear $\mathcal{P} \cup \mathcal{C}$ by at least d_{xy} and d_z in the horizontal and vertical directions.
8. The fixture obeys all specified minimum-quality thresholds. In our metric, this implies:
   (a) No expected applied force will cause a contact reaction force greater than F_{max}.
   (b) No expected shape deviation can cause a critical part feature to deviate from its nominal xy-position by a distance greater than its associated tolerance d_{max}.
   (c) The fixture is easy to load, meaning that: *(i)* there is a placement of $\mathcal{P}$ that clears all fixture elements by at least d_{load}, *(ii)* the part can be loaded vertically, clearing the fixture by at least d_{load}, and *(iii)* the part center of gravity is supported during loading.

4 Algorithm Synopsis

The short format of a conference paper does not provide nearly enough room to explain all of the details involved in making the algorithm correct and efficient. Thus we will only sketch the algorithm here, and refer the interested reader to [3] for a full description.

The fixture design algorithm is implemented with two processes. The fixture generation process accepts a problem description data structure and outputs a stream of fixture designs to a fixture queue. The user interface process accepts problem specification information from the user, starts the fixture generation process, and allows the user to interactively inspect the contents of the fixture queue. From here on we will focus on the fixture generation process.

The first step of the fixture generation process is to extract legal contact surfaces from $\mathcal{P}$. These correspond to horizontal top and bottom surfaces, and vertical side surfaces. These surfaces are projected onto the xy-plane to form planar regions that correspond to the top and bottom surfaces, and contours that correspond to the vertical surfaces. Pointers back to the generating surfaces of $\mathcal{W}$ are stored for later height lookups. Our current implementation only accepts linear descriptions of $\mathcal{P}$, so non-linear features should be converted to piecewise linear approximations.

The algorithm uses these projected features to generate fixture designs, checking for geometric interference against $\mathcal{P} \cup \mathcal{C}$ as it proceeds. The basic generation process entails synthesis of a planar xy-constraint, synthesis of an out-of-plane z-constraint, and merging these two constraint designs to form a 3d-fixture. The sequence and content of these operations depends on whether the fixture kit includes top clamps, and whether support pads are constrained to the grid.

The algorithm synthesizes xy-constraint designs using the algorithm reported in [2]. This algorithm produces a complete enumeration of the possible planar fixture designs for an input polygonal shape.

Let us first consider the case with top-clamps and the supports restricted to the grid. This case is of practical interest because in prototype production situations, the resulting designs may be rapidly assembled to produce a precise fixture. In this situation the algorithm first generates and scores a valid xy-constraint. The algorithm then synthesizes a z-constraint by using the resulting (x, y, θ) part configuration to identify all valid support pad locations which have at least one valid placement for the top clamp body. The algorithm generates all tripods of these locations, attempting to identify a placement of the top-clamp bodies that avoids all sources of interference. If a valid placement is found which produces non-$\emptyset$ quality scores, then the algorithm outputs the design to the fixture queue.

Now suppose we allow supports to be placed off the grid. This is reasonable in mass production scenarios where fixtures are fabricated from plain tooling plate. Fixture kits that allow this freedom are fundamentally less complex to plan for than fixture kits that restrict support locations to the grid, since the xy-constraint and z-constraint enumeration procedures may be decoupled instead of nested.

In this case the algorithm identifies all valid support locations for $\mathcal{P}$ at (0,0,0), sampling both a grid of possible points and also the vertices of the horizontal surfaces shrunk by $r_{\text{support}} + d_{xy}$. The algorithm then forms all feasible tripods of these points, producing a list of z-constraints sorted by q_z. The algorithm then enumerates xy-constraints. For each xy-constraint, the algorithm proceeds down the sorted list of z-constraints, starting with the best first. For each z-constraint, the algorithm transforms the z-constraint by the xy-constraint's (x, y, θ) pose, and attempts to find a valid placement of the z-constraint elements that avoid all sources of interference.

Here is where pruning may be applied. For a given xy-constraint and z-constraint, the algorithm can examine the partial quality scores q_{xy} and q_z and the remaining weight w_{3d} to determine whether the current xy-constraint and z-constraint have the potential to better the best-quality fixture produced so far. If not, then the algorithm can avoid analyzing this z-constraint, as well as all subsequent z-constraints on the list. Because there are so many z-constraint tripods, this branch-and-bound pruning method can produce substantial computational savings, leading to speedup factors better than 100.

In the assembly pallet case, allowing supports to be placed off the grid makes less of a difference, since there is only one convex-hull z-constraint for each xy-constraint. Still, speedup factors of 2 are possible when supports are allowed off the grid, simply because the support-pad interference checks do not need to be repeated for each xy-constraint.

Regardless of the enumeration procedure employed, the fixture design algorithm applies quality metrics to score and sort fixture designs. These quality metrics may be arbitrary, but should be designed to allow meaningful combination of quality values from functions that consider incomparable aspects of the fixture. In our quality metric we apply a canonical method of normalizing quality values. Each quality metric is based on some scalar $v_{\max}$ that increases as quality decreases. For example, $v_{\max}$ might correspond to the maximum contact reaction force exerted by the fixture on the part, or the maximum deviation of a critical feature from its nominal location. We assume that the user will specify some maximum allowable value v_{limit} and a minimum value v_{ideal}, below which improvements are unimportant. A quality score $q \in [0, 1]$ can then be obtained by calculating $v_{\max}$ for the fixture, and then using $[v_{\text{ideal}}, v_{\text{limit}}]$ to produce a thresholded, normalized q.

5 Previous Work

The literature in fixture design and the related problem of grasp planning is vast, and space does not allow us to provide a thorough review of it here. Our work is especially inspired by Asada and By [1], Hoffman [6], Kim [7], Sakurai [9], Englert [5], and Chang [4], who emphasized the importance of including task constraints and process considerations in the fixture design process. Further, our algorithm builds directly on the prior results of Brost and Goldberg [2] and Wallack and Canny [10]. Please see [3] or [2] for a thorough review of the literature in fixture and grasp design.

6 Discussion

The style of fixtures generated by the current design algorithm implicitly restricts the set of fixturing problems that may be solved. For example, the algorithm only places contacts on horizontal or vertical surfaces. Thus while the algorithm may be applied to parts of arbitrary shape, it will perform poorly on problems where the part has few of these surfaces. Further, since the algorithm only returns fixture designs that provide kinematic form closure, the algorithm cannot find solutions in cases where kinematic constraint is impossible. Examples include spherical or cylindrical parts, which require friction to constrain [8].

Another important limitation is the rigid-body assumption employed by the algorithm. When applied forces are large compared to the strength of the part, additional support is required to prevent part deformation. Such supports could conceivably be synthesized by an extension of this algorithm.

A number of problems remain unresolved in the area of mixed-part and multi-part fixture design. The results in Section 2.2 were obtained by a generate-and-test procedure for identifying mixed-part pallet designs; stronger algorithms appear possible. Further, some assembly problems require the location and support of multiple parts before fastening; these problems lie outside the scope of our current algorithm. Developing strong algorithms for mixed-part and multi-part fixture design remains a fertile area for future work.

References

[1] H. Asada and A. B. By. Kinematics of workpart fixturing. In *Proceedings, IEEE Intl. Conf. on Robotics and Automation*, pp. 337–345, March 1985.

[2] R. C. Brost and K. Y. Goldberg. A complete algorithm for synthesizing modular fixtures for polygonal parts. Technical Report SAND Report 93-2028, Sandia National Laboratories, January 1994.

[3] R. C. Brost and R. R. Peters. Automatic design of 3-d fixtures and assembly pallets. Technical report, Sandia National Laboratories, 1995. In preparation.

[4] C. Chang. Computer-assisted fixture planning for machining processes. *Mfg. Review*, 5(1), March 1992.

[5] P. J. Englert and P. K. Wright. Applications of artificial intelligence and the design of fixtures for automated manufacturing. *IEEE ICRA*, April 1986.

[6] E. G. Hoffman. *Modular Fixturing.* Manufacturing Technology Press, Lake Geneva, Wisconsin, 1987.

[7] K. H. Kim. *A System for Automated Fixture Planning with Modular Fixtures.* PhD thesis, Carnegie Mellon University Robotics Institute, May 1993.

[8] B. Mishra, J. T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2(4):641–558, 1987.

[9] H. Sakurai. *Automated Setup Planning and Fixture Design for Machining.* PhD thesis, MIT Department of Mechanical Engineering, January 1990.

[10] A. Wallack and J. Canny. Planning for modular and hybrid fixtures. *IEEE ICRA*, May 1994.