SANDIA REPORT

SAND95–2411 • UC–705
Unlimited Release
Printed January 1997

Automatic Design of 3-d Fixtures and Assembly Pallets

Randy C. Brost, Ralph R. Peters

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release; distribution is unlimited.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A05
Microfiche copy: A01

SAND-95-2411
Unlimited Release
Printed January 1997

Distribution
Category UC-705

Automatic Design of 3-d Fixtures and Assembly Pallets

Randy C. Brost
Ralph R. Peters
Intelligent Systems and Robotics Center
Sandia National Laboratories
Albuquerque, NM 87185-5800

Abstract

This paper presents an implemented algorithm that automatically designs fixtures and assembly pallets to hold three-dimensional parts. All fixtures generated by the algorithm employ round side locators, a side clamp, and cylindrical supports; depending on the value of an input control flag, the fixture may also include swing-arm top clamps. Using these modular elements, the algorithm designs fixtures that rigidly constrain and locate the part, obey task constraints, are robust to part shape variations, are easy to load, and are economical to produce. For the class of fixtures that are considered, the algorithm is guaranteed to find the global optimum design 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 a minute, and identifies the global optimum design in just over an hour. The algorithm is also capable of solving difficult design problems where a single fixture is desired that can hold either of two parts.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Contents

1 Introduction 1

2 Previous Work 7

3 Fixture Kit 9
- 3.1 Baseplate 9
- 3.2 Lateral Locator 9
- 3.3 Side Clamp 9
- 3.4 Support Pad 11
- 3.5 Top Clamp 12

4 Design Algorithm 13
- 4.1 Problem Statement 13
  - 4.1.1 Assumptions 13
  - 4.1.2 Input 14
  - 4.1.3 Output 15
- 4.2 Fixture Generation 16
  - 4.2.1 Input Pre-Processing 16
  - 4.2.2 Generation of xy-Constraints . . 17
  - 4.2.3 Generation of z-Constraints . . . 23
  - 4.2.4 Generation of 3-d Fixtures . . . 26
- 4.3 Robust Form Closure 26
- 4.4 Branch-and-Bound Pruning 29

5 Quality Functions 31
- 5.1 Force-Based 31
- 5.2 Position-Based 32
- 5.3 Loading-Based 34
  - 5.3.1 Geometric Loading Analysis . . . 34
  - 5.3.2 Physical Loading Analysis 35
- 5.4 Overall Score 36

6 Algorithm Variations 38
- 6.1 Off-Grid z-Constraints 38
- 6.2 Fixtures Without Top Clamps 39

7 Case Studies 42
- 7.1 Final Machining of Complex Workpieces 42
- 7.2 Light Mechanical Assembly 51

8 Discussion 57

Acknowledgments 60

A Pose Calculation 60

B Force Calculations 62

C Form Closure Numerical Issues 68

D Push-Into-Place Loading Analysis 69
- D.1 Generating Sample Workpiece Positions 70
- D.2 Testing for Jamming 70
- D.3 Testing for Clamp Stall 71

References 76

1 Introduction

Fixture design is a practical problem. When manufacturing products, it is often necessary to hold a workpiece in place during the course of several manufacturing tasks, such as machining, assembly, or inspection operations. The fixtures used to hold the workpiece must prevent undesired part motions and avoid interfering with these tasks, often with the additional requirement that the workpiece must be held in an accurate, repeatable position. These conditions must be maintained even in the face of small variations in workpiece 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.

Thus the fixture design problem is characterized by a blend of technical and pragmatic issues that must be considered together to identify a successful fixture design. In this paper, we present an implemented algorithm that automatically designs optimal fixtures for a particular class of problems. The resulting fixtures provide rigid constraint and deterministic location of the workpiece, obey all associated task constraints, are robust in the face of workpiece shape variations, are easy to load and unload, and offer good economic performance in a variety of business scenarios.

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 with a flat or self-aligning top, and swing-arm top clamps with a flat or self-aligning contact pad. Locating and clamping elements in this class are widely available, and are often employed in fixtures based on the 3-2-1 location scheme [34].

These elements are used by the algorithm to design fixtures that hold the workpiece 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 plane. This scheme guarantees kinematic constraint in all six degrees of freedom, and follows common 3-2-1 fixturing practice. Further following common practice, the algorithm only places top clamps directly above support points to avoid clamp-induced workpiece deformations.

Given a fixturing problem specified by a workpiece description and a set of task constraints, the algorithm uses these elements to generate a series of feasible fixture designs that provide form closure while obeying the constraints. Each fixture is analyzed for loadability and passed to a user-defined quality metric which rates the fixture design. This quality metric can consider arbitrary aspects of the fixture in assigning a quality score, and may apply thresholds to determine whether any of these aspects is unacceptably poor. The fixture is discarded if it fails to pass some threshold; otherwise, it is assigned a scalar quality value. The algorithm employs branch-and-bound pruning to identify the global optimum fixture design while exploring only a small portion of the feasible fixture design space. If pruning is turned off, the algorithm enumerates all possible fixture designs. Either way, the algorithm is guaranteed to find the optimal fixture design, for the given set of optimality criteria and associated thresholds.

The resulting optimum fixture design is the best possible design that can be constructed with the available fixture elements, subject to several small approximations described below. The most important of these approximations is that the user may instruct the algorithm to limit its search for side locator height combinations, for example considering only height combinations where each locator is at its lowest available height. This restriction often has a minor effect on fixture quality, but allows a significant reduction in search time. These approximations may be removed at the expense of additional computation. We have chosen to include these approximations in our implementation because, in our judgment, the improvements in run time are worth more than the marginal increase in fixture quality that could be obtained by replacing each approximation with a complete computation.

Our implementation provides a quality metric that considers three fixture aspects: the fixture's ability to resist expected forces without exerting large reaction forces on the workpiece, the fixture's ability to repeatably locate critical features of the workpiece, and the ease of loading the fixture. This metric is sensitive to the specific manufacturing operations that will be applied to the workpiece, and applies independent thresholds to each of these criteria. Each quality criterion returns a value between 0.0 and 1.0, and the total quality is determined by either a weighted sum or minimum of the three resulting values, based on user preference. This metric is appropriate for a variety of fixture applications, but other metrics may also be supplied to the program.

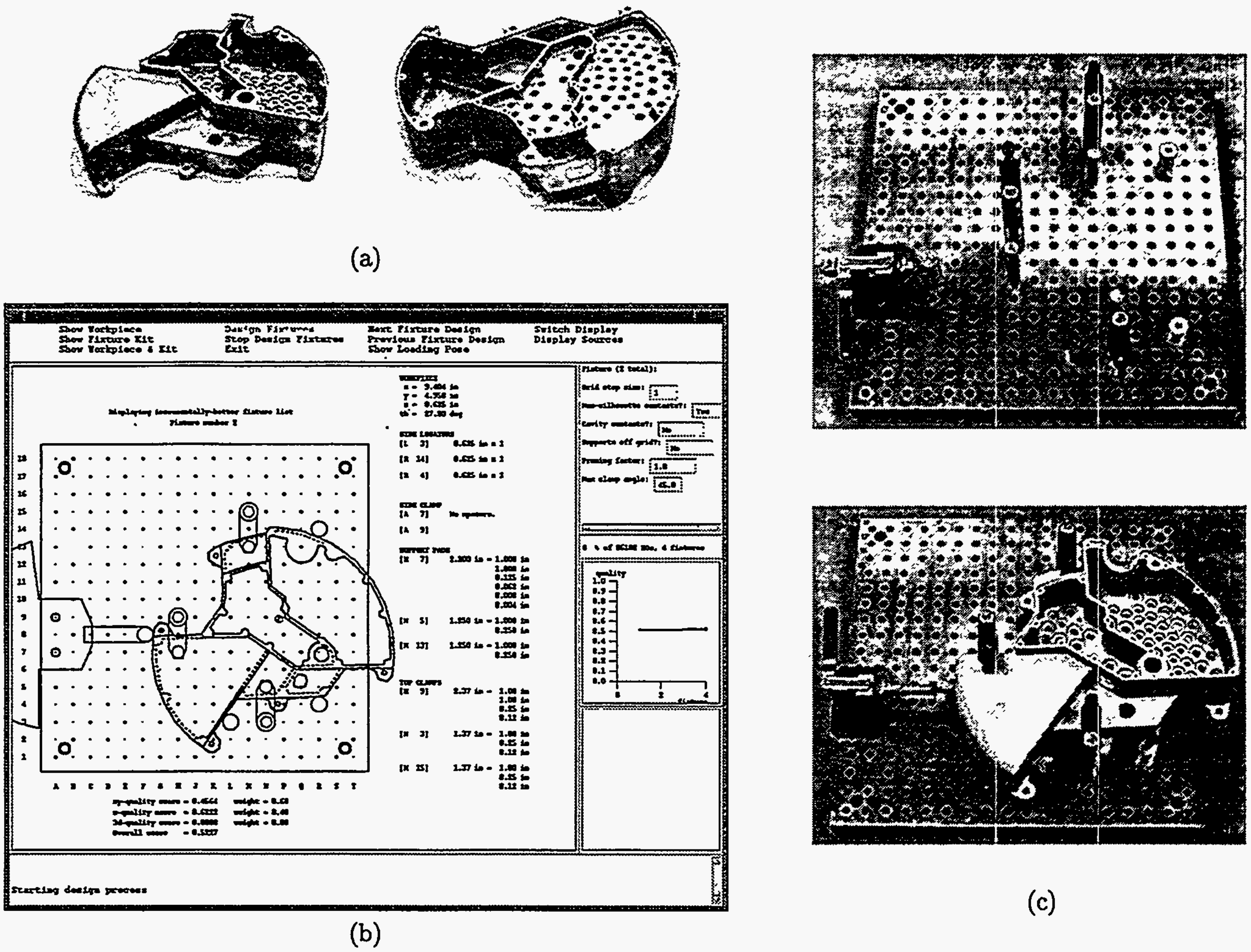


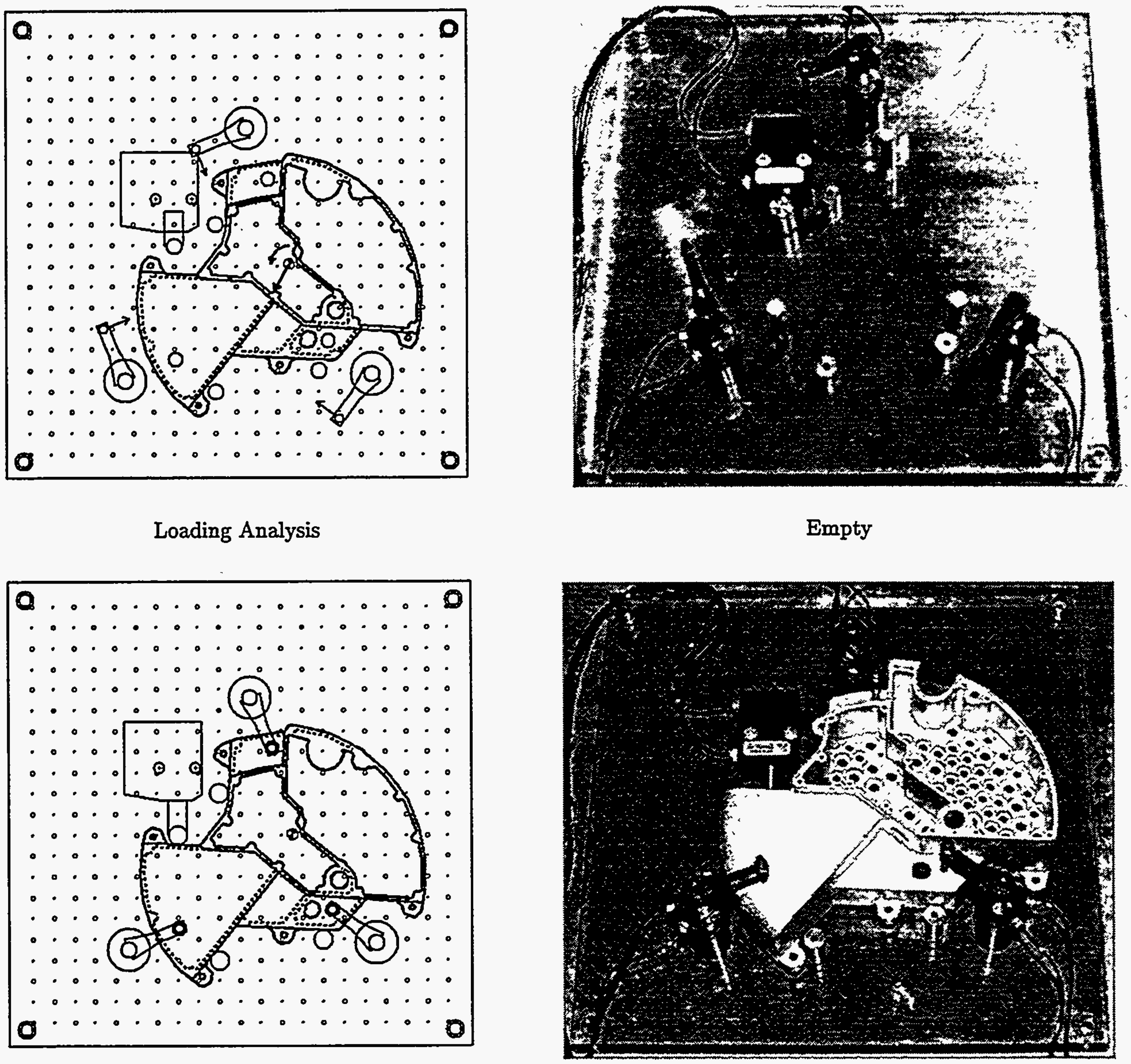
Figure 1: (a) Top and bottom views of a cast housing that requires finish machining. (b) Fixture design for machining a prototype housing, shown with the design tool user interface. (c) The fixture, shown empty and loaded.

Our implementation also includes an interface that interactively displays the fixture designs as they are generated. This allows the user to apply subjective evaluation criteria that are not included in the program's quality metric, and also to stop the computation once an acceptable fixture design is generated. Thus the user may accept an early fixture design to reduce program run time, or obtain the global optimum by letting the program run to completion. The algorithm employs branch-and-bound pruning to restrict the enumeration and quickly find the global optimum; the user can reduce the effect of this pruning as desired.

Figures 1–4 show several examples.

Figure 1 shows a fixture designed by the algorithm for a job-shop machining problem, comprised of modular elements that may be rapidly assembled in various configurations. In this problem, a near-net-shape casting requires several finish machining operations. This problem is difficult to solve because the complex pockets and large number of holes greatly constrain the placement of support elements. The input description of this problem includes the workpiece shape, regions swept by the cutter, expected machining forces, tolerance limits on critical workpiece features, and the components of the fixture kit. Every fixture design returned by the algorithm is guaranteed to resist the expected machining forces without causing permanent deformation of the workpiece at the fixture contacts. For this scenario, we selected a high-quality fixture that appeared early in the computation; this fixture design was computed in 18 seconds on an SGI workstation.

Figure 2 shows a fixture designed to produce the same part on an automated machining line. This fixture employs hydraulic clamps that may be automati-



Loading Analysis

Empty

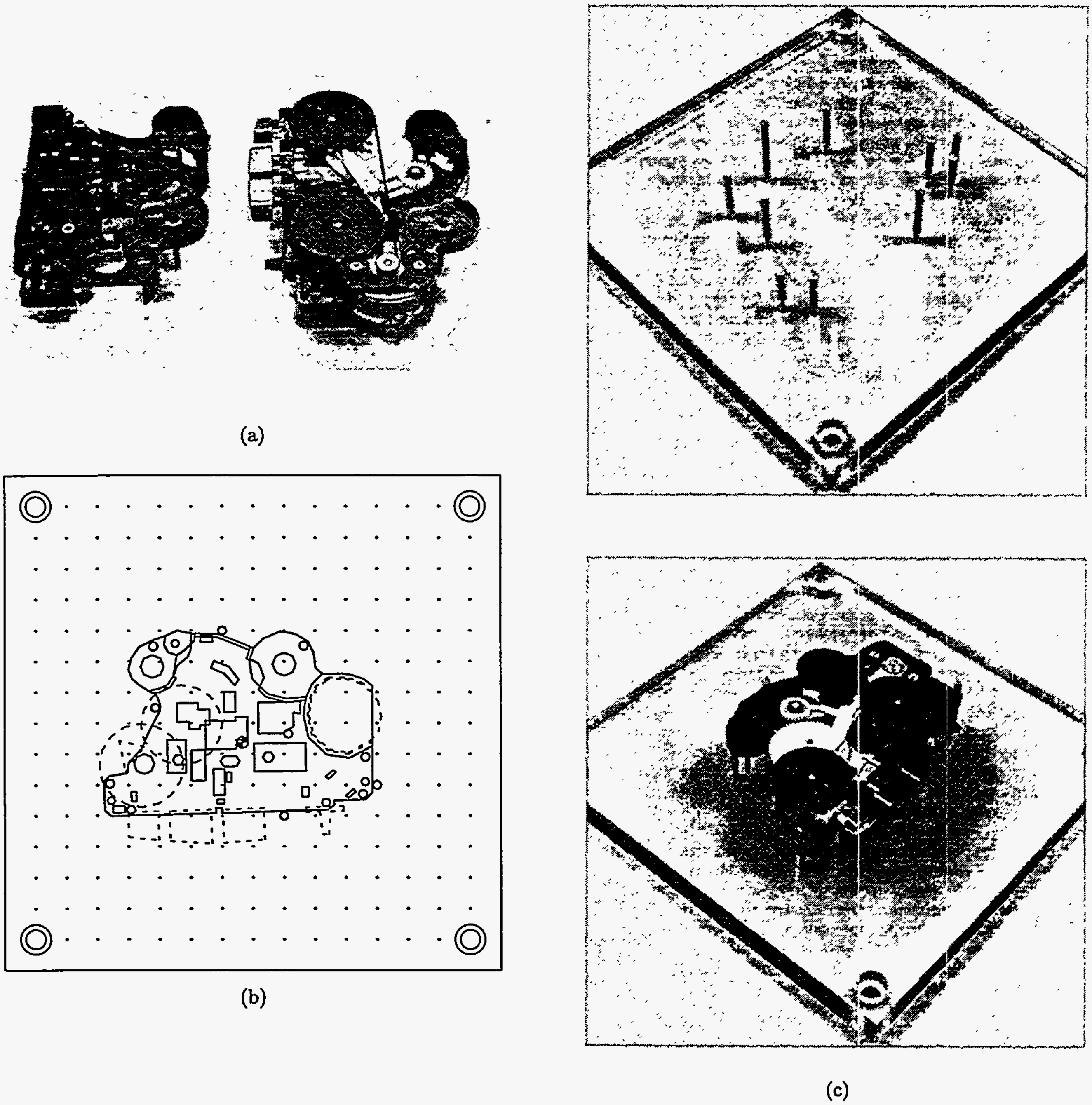
Loaded

Loaded

Figure 2: Optimal fixture design for mass-production of the housing. The arm closing directions were automatically selected to avoid collisions.

cally actuated. The problem is described by the same information used in the job-shop machining scenario, except for differences in the fixture elements and loading requirements. For this problem, every fixture design returned by the algorithm provides all of the guarantees assured in the manual case, plus the additional guarantee that the workpiece may be easily loaded using a simple vertical loading strategy that includes bounded motion uncertainty. The algorithm returns initial valid designs within 66 seconds. However, because more time is available for fixture design in a mass production application, we let the algorithm run to completion, identifying the global optimum design. This computation was performed in 61 minutes.

Figure 3 shows a fixture designed for automated assembly, sometimes called an assembly pallet. Because the pallet will be used in an assembly line, the fixture

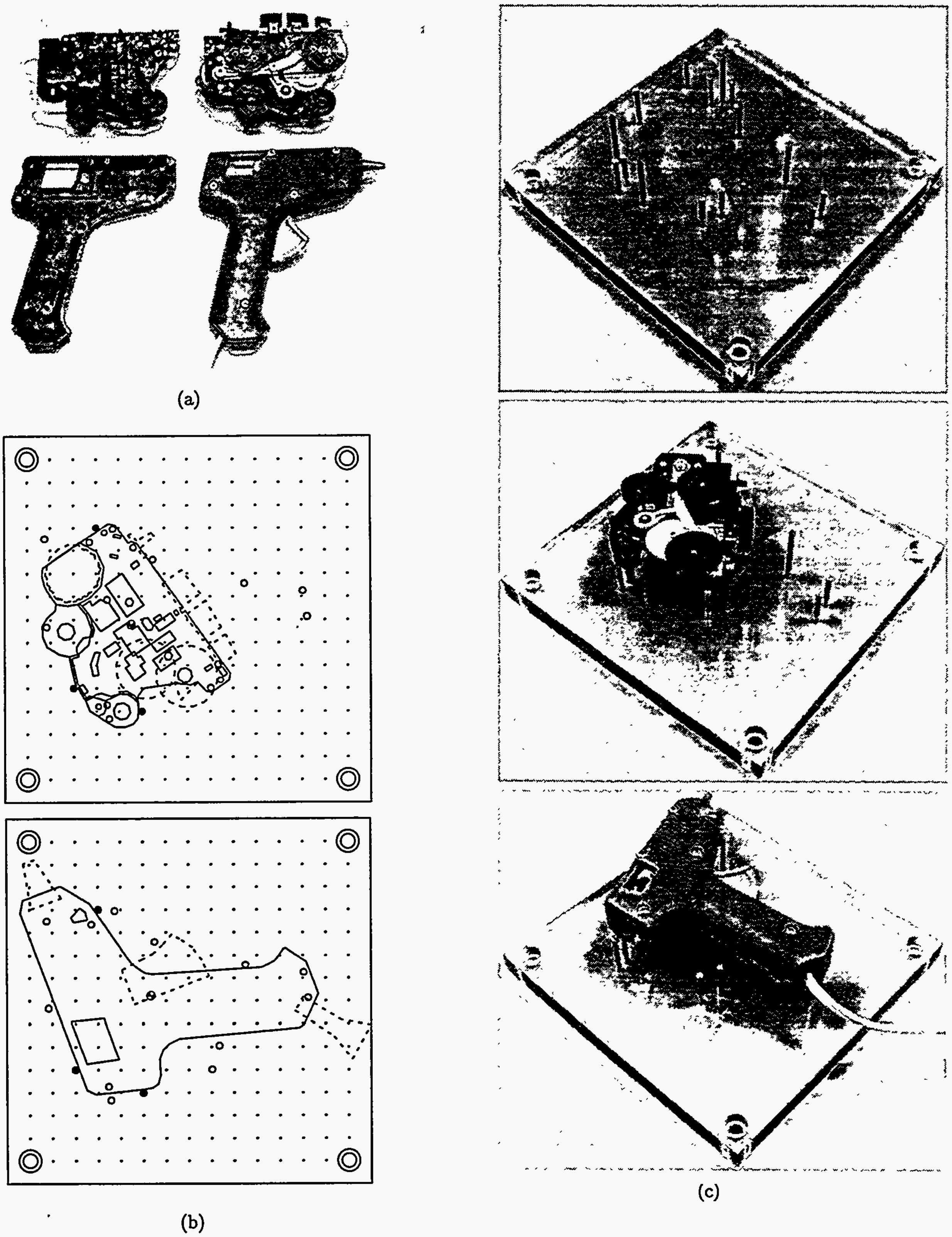

(a)

(b)

(c)

Figure 3: (a) The chassis of a personal cassette player, before and after assembly operations. (b) Optimal assembly pallet design for the cassette player. (c) The pallet, shown empty and loaded.

elements are designed to be inexpensive to fabricate and easy to load vertically. In this problem, a personal cassette player mechanism is assembled by vertically inserting several parts onto a main chassis. The underside of the chassis has several fragile pins extending downward, requiring the pallet to hold the chassis off the pallet surface. A number of the attached parts extend past the boundary of the chassis, constraining the placement of fixture elements. The input description of this problem includes the shape of the chassis, the



(a)

(b)

(c)

Figure 4: A pallet allowing the assembly of two different products. (a) A pair of dissimilar products chosen to demonstrate the capabilities of the algorithm. (b) Optimal mixed-part pallet design. Pins in common to both designs are darkened for clarity. (c) The mixed-part pallet, shown empty and loaded with each product.

regions occupied by added parts, expected assembly and pallet transfer forces, position tolerances associated with assembly sites, and the pallet fixture elements. Every fixture design returned by the algorithm is guaranteed to prevent part motion during insertion and pallet transfer operations, and also to repeatably locate critical assembly sites within their associated position tolerances. The algorithm produced an initial fixture design for this problem in 1.7 minutes, and found the global optimum design in 9.4 minutes.

Figure 4 shows a fixture designed to support the assembly of two different products on the same line. Fixtures of this type are often required by companies that manufacture a family of related products with varying options. In this case, we chose an extreme example of two dissimilar products to demonstrate the capabilities of the algorithm. To design this fixture, we implemented a procedure that runs the algorithm on each problem individually, generating two sets of single-product fixture designs. The procedure then forms all pairs of the resulting designs, discarding designs that are not mutually compatible. This produces a series of mixed-product fixture designs that include one, two, or three pins in common between the constituent single-product designs. The figure shows the design pair with three common pins that had the highest aggregate quality score. This fixture satisfies all of the assembly requirements of both products, and allows the manufacturer to easily switch between products to adjust production volumes in response to varying market conditions. This entire computation was performed in 146 minutes.

Section 7 will explain each of these examples in detail.

The style of fixtures generated by our algorithm implicitly restricts the set of fixturing problems that may be solved. For example, the algorithm only places supports and top clamps on horizontal surfaces, and only places side locators and clamps on vertical surfaces. Thus while the algorithm may be applied to workpieces of arbitrary shape, it will perform poorly on problems where the workpiece has few of these surfaces. Further, because 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 workpieces, which require friction to constrain [52].

Another important limitation is that our algorithm only provides constraint for a single part. In some assembly problems, it is necessary to hold several parts in a desired relative position before fastening them together. For example, the motor shown in Figure 3(a) must be supported from below before it is fastened to the chassis with two screws. Thus the pallet shown in Figure 3(c) requires an additional manually-designed support for the electric motor, or the motor must be assembled to the chassis before it is loaded into the pallet. Section 7.2 explains how the algorithm input may be configured to leave room for a manually-designed support.

Thus our fixture design tool only addresses a subclass of all possible fixture design problems, namely those that involve a single workpiece with an abundance of horizontal and vertical surfaces, that is not a surface of revolution. In Section 8 we show that for problems in this class, the algorithm often returns a large number of high-quality solutions. However, for some problems the details of the workpiece geometry or task constraints may foil the algorithm, causing it to return no solutions or only solutions of poor quality. Extending the fixture design algorithm to employ a broader repertoire of locating and clamping primitives would eliminate some of these limitations, and expand the set of problems for which the algorithm returns useful solutions. In the future we hope to extend this work to address a broader range of problems within the current class, and also address problems outside the class. We will return to this topic in Section 8.

The remainder of this paper will explain this algorithm in detail. We begin by reviewing the background literature in Section 2. Then we present a detailed explanation of the algorithm in Sections 4–6. Next we present several case studies in Section 7 which show how the algorithm may be applied to practical problems. Section 8 concludes by discussing lessons learned and opportunities for future work.

2 Previous Work

This paper draws on a number of prior results in fixture design, grasp planning, and mechanics analysis. The literature in these areas is vast, and in this section we will discuss only those results that are directly relevant. For surveys, see [57, 28].

The algorithm reported in this paper is a direct descendant of the Brost and Goldberg algorithm for designing planar fixtures [10]. In this paper, we have extended the prior algorithm to three dimensions, added the notion of robust form closure, extended the fixture quality metric to consider 3-d forces, position repeatability, and ease of loading, and extended the algorithm to design fixtures for mass production applications as well as small-lot applications. Further, we developed an anytime-algorithm user interface which allows the user to obtain initial fixture designs very quickly; this interface also returns information indicating why a given fixture design problem is difficult, allowing the user the option of modifying the input problem specification accordingly.

Wallack and Canny [76] independently produced an algorithm that is very similar to the 2-d fixture design algorithm reported in [10], and have since extended it to include non-linear object features [77]. Wallack and Canny's algorithm is based on a modular vise with pins placed in a grid of holes on the top of each vise jaw, rather than the single plate with detachable clamps and pins treated in this article. In related work in our group, we have extended Wallack and Canny's planar algorithm to design 3-d robot grippers [13].

This work attempts to bridge the gap between past results addressing the practical aspects of fixture analysis and design, and related results regarding the algorithmic nature of automatic fixture and grasp planning.

In the area of fixture analysis and design, Asada and By [1] published a landmark paper which reported methods for analyzing the constraint and instantaneous loadability of a given fixture design. Further, Englert, Hayes, Sakurai, and Kim each produced excellent Ph.D. theses which developed strong fixture analysis methods for machining applications, and basic planning algorithms that considered the task of taking a workpiece from raw stock to its final form [21, 30, 66, 35]. Other papers developing analysis methods and heuristic planning procedures include [25, 38, 34, 7, 17, 46, 6, 32, 18, 36, 39, 47, 16, 69]. These papers all developed methods for analyzing the performance of a given fixture, and some also described fixture design procedures based on heuristics or incomplete algorithms. In this paper, we develop a complete synthesis algorithm for workpieces of arbitrary shape, and apply many of the analysis techniques developed in these prior papers. However, we do not apply the full process plan analysis or workpiece deformation analysis techniques considered in some of these papers, such as [19, 38, 46, 30, 66, 67, 16].

There is current work in this vein that is closely related to this result, and in many ways complementary. The Agile Fixturing Project pursued by the Machine Tool Agile Manufacturing Research Institute (MT-AMRI) has several activities in this area. These include analysis of fixture performance using rigid-body and compliant workpiece models, automatic prediction of cutting forces from machining process plans, and fixture optimization. These activities complement our work in automatic fixture synthesis by developing richer analysis methods and automatic generation of required input. The principal investigators of this project are R. E. DeVor at the University of Illinois at Urbana-Champaign, E. C. De Meter at Penn State University, and J. Hu at the University of Michigan, Ann Arbor. While much of this work is still in progress, a few publications have appeared, including [48, 15, 29, 33, 62, 50].

In the field of computer science, several complete grasp synthesis algorithms have been reported, as well as proofs of completeness for abstract grasping problems. Example synthesis algorithms include [3, 52, 42, 53, 51, 59, 60] which approach the grasp planning problem from a variety of perspectives. Each of these results addresses an abstract formulation of the grasping problem — for example, identifying a set of contact points that will hold a given object in form closure. In addition, there have been several algorithms that design fixtures using a variety of proposed hardware primitives [75, 58, 56]. Finally, there have been a number of papers developing metrics to assess grasp and fixture quality [68, 55, 44, 24, 72]. Our work is distinguished from these past results in two main ways: First, we explicitly represent task requirements such as expected applied forces or regions that must remain clear to avoid interference with the task. This task-specific information influences both the algorithm's basic synthesis procedure and its fixture quality analysis. Second, we guarantee that the fixture can be easily built using proven and available hardware components, whereas these prior algorithms synthesized grasp configurations using idealized point contacts or unproven hardware systems. A notable exception is the M.I.T. Handey project [40], which produced a grasp-planning algorithm that simultaneously considered task geometric constraints and gripper characteristics for planning robot pick-and-place tasks.

From a fundamental perspective, there have been a number of proofs addressing necessary and sufficient conditions for constraining a rigid body with unilateral contacts. These include Somoff and Lakshminarayana's independent proofs that seven points are necessary to constrain a rigid body in form closure [71, 37], and the result by Mishra, *et al* and Markenscoff, *et al* proving that twelve and seven points respectively are sufficient for bodies that are not surfaces of revolution [52, 43]. Our work stands in contrast to these results, primarily because of the practical issues surrounding real fixture design problems. First, we employ more than the minimum number of required contacts so that we can employ standard locating and clamping elements and avoid clamp-induced workpiece deformations. Second, in many cases a form-closure fixture design is unavailable for a given problem, because of geometric constraints imposed by the task. The assembly pallets designed by our algorithm provide one example. Thus we have taken the approach of centering our algorithm on the capabilities of proven fixture hardware while explicitly considering task constraints.

It is worth noting that the hardware that we have selected satisfies numerous practical constraints, but is not capable of solving all fixture problems. In addition to excluding surfaces of revolution and objects without horizontal and vertical surfaces available for contact, the fixture hardware is fundamentally incapable of holding objects that are very small relative to the fixture grid size, or which closely approximate a surface of revolution [81]. Wentink, *et al* provide a detailed evaluation of the capabilities of various types of fixture kits [78].

This paper addresses the problem of constraining a single part. This part may be a workpiece that requires machining, or a base part to which other parts are assembled. Some assembly problems require the fixture to hold multiple parts in a relative position before fastening operations are performed; designing these fixtures requires a multi-part constraint analysis. This problem has been studied by several authors. De Meter presents a detailed method of analyzing multi-part assembly pallets in [49], while Baraff, *et al* and Wolter and Trinkle present analysis results and synthesis methods for multi-part fixtures using abstract fixture elements [5, 79].

This section has provided a brief overview of related work. There are several other prior results which relate to specific aspects of our fixture design algorithm; we refer to these throughout the remaining text at the points where they are most relevant. Our work has been previously reported in conference form in [12].

The remainder of this paper will present the fixture design algorithm and show its application to several examples. We begin by describing the class of fixture elements that are considered.

3 Fixture Kit

The fixture design program designs fixtures consisting of a baseplate, three lateral locators, a side clamp, support pads, and, in some cases, top clamps. The choice of components used in the design may significantly affect the appearance and the capabilities of the fixture. Each of these components is discussed in the following sections.

3.1 Baseplate

The baseplate is a rectangular plate with holes drilled into it for attaching other fixture components. The size of the baseplate is specified by the dimensions $x_{\text{min-plate}}$, $x_{\text{max-plate}}$, $y_{\text{min-plate}}$, and $y_{\text{max-plate}}$. The holes may be either fully-threaded holes that are used to securely attach components such as top clamps, or dowel holes that are used to precisely locate components such as lateral locators; the lower portion of each dowel hole is threaded. The tolerance in lateral position of the dowel holes ϵ_{hole} is specified. These holes are arranged in a square pattern with a distance between holes of d_{grid} and grid pattern dimensions specified by $x_{\text{min-grid}}$, $x_{\text{max-grid}}$, $y_{\text{min-grid}}$, and $y_{\text{max-grid}}$. The dowel-hole grid-pattern may be either `all` or `alternating`. These holes may be pre-drilled, in the case of a completely modular kit, or drilled on an as-needed basis. In the latter case the baseplate may have holes for the support pads and top clamps drilled off-grid.

3.2 Lateral Locator

Three lateral locators and a side clamp provide lateral constraint of the workpiece. Figure 5 shows a lateral locator and its associated dimensions.. This type of lateral locator is plugged into a dowel hole in the baseplate for precise positioning. Contact with the workpiece occurs only on the qualified surface of the locator, indicated by the label h_{contact}. Each lateral locator also has an associated tolerance $\epsilon_{\text{locator}}$, indicating the maximum possible distance between the nominal locator surface and the true locator surface. Spacers are used to set the locator at various heights above the baseplate. We assume that the spacer radius is no larger than $r_{\text{locator}} - \epsilon_{\text{locator}}$, assuring that a vertical workpiece face will touch the locator and not the spacer. Spacers may be available in a limited set of discrete heights.

Assembly pallets use lateral locators that are rods pressed into the baseplate. Figure 6 shows such a lateral locator and its associated dimensions. The conical tips of these lateral locators ease loading of the workpiece into the assembly pallet. Because the rod can be cut to the appropriate length, spacers are not used. Also, the parameter h_{contact} is not relevant because contact with the workpiece may occur at any position below the conical tip.

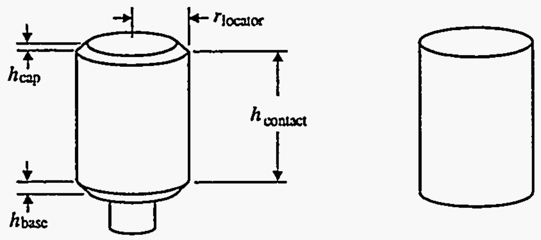


Figure 5: Modular lateral locator with dimensions. The upper and lower chamfers are unusable for contact. For interference checks, the locator is modeled using the conservative volume shown at right.

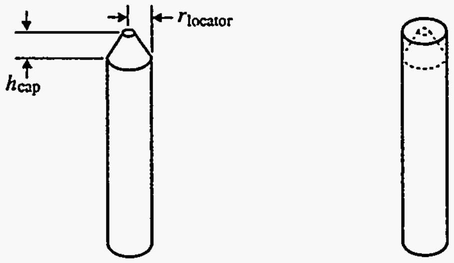


Figure 6: Assembly pallet lateral locator. The upper cone is unusable for contact. For interference checks, the locator is modeled using the conservative volume shown at right.

3.3 Side Clamp

The side clamp holds the workpiece against the three lateral locators. Figure 7 shows a side clamp and its characteristic vertical dimensions. The side clamp body and associated fittings are represented as a prism of height h_{body}.

A number of parameters describe the side clamp plunger. The cylindrical plunger tip is required to have the same radius as the lateral locators. The remainder of the plunger is represented conservatively as a prism with a height equal to the total height of the plunger tip. The position of the plunger tip relative to the side clamp body is described by the parameters x_{travel} and y_{offset}, shown in Figure 8. The tip position may vary over the range $[x_{\text{travel}_{\text{min}}}, x_{\text{travel}_{\text{max}}}]$ when the clamp is closed, and retracts to $x_{\text{retracted}}$ when the clamp is open. The constraint provided by the side clamp may be modeled in one of two ways, depending on the type of clamp. Some manual clamps have overcenter mechanisms that lock the plunger in place; we treat these

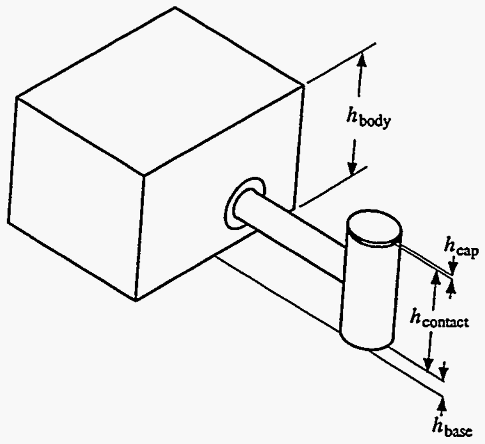


Figure 7: Side clamp vertical dimensions.

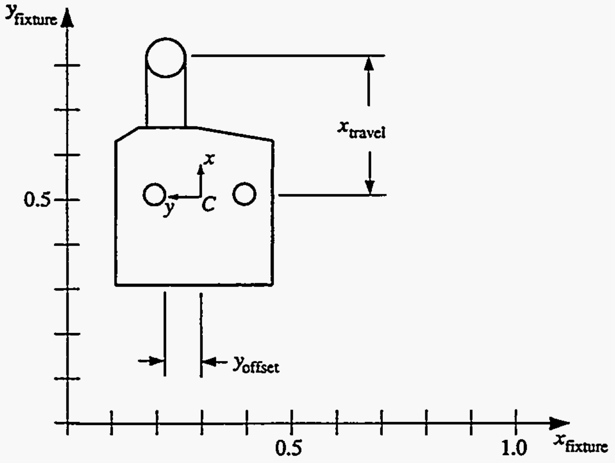
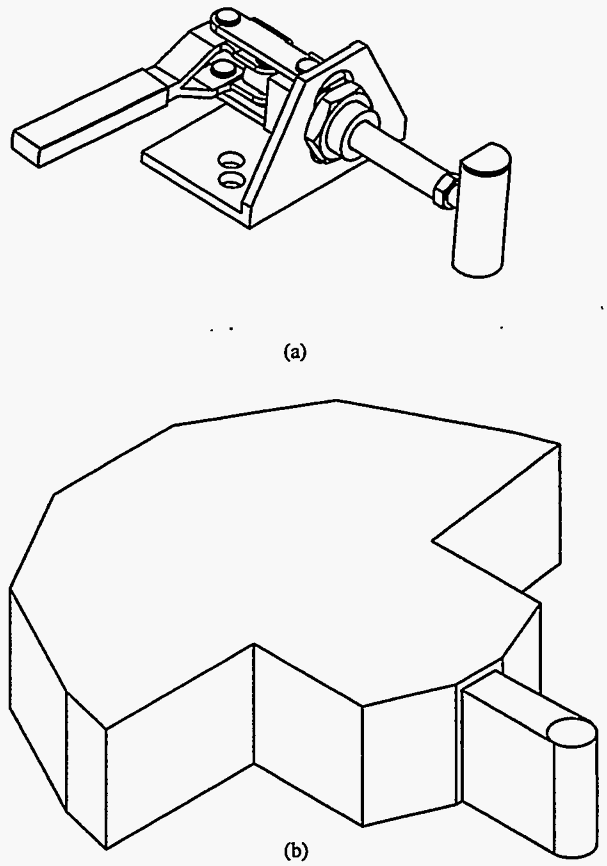


Figure 8: Side clamp coordinates. The xy coordinate frame C defines the position of the clamp on the fixture; the position $(0.3, 0.5, 90°)$ is shown here. The clamp tip translates along a line defined by the input parameter y_{offset}; the position of the tip along this line is x_{travel}.

as rigid kinematic constraints. For hydraulic or pneumatic clamps, we assume that the clamp plunger exerts a fixed force F_{plunger}.

An example manual side clamp is shown in Figure 9. The upper panel shows a drawing of the actual clamp. The lower panel shows the body prism, the plunger prism, and the plunger tip used in designing a fixture. The body prism includes the side clamp body and the region swept by the handle as the clamp is opened and closed. Because we seek to assure positive clearance in various collision checks, all non-locating boundaries of fixture elements are described using conservative volumes that enclose their maximum material condition. These enclosing shapes are simplified for efficiency.


(a)

(b)

Figure 9: (a) A typical manual side clamp, manufactured by Qu-Co. (b) Our conservative model of the clamp. The fan-shaped area includes the region swept by the handle during clamp actuation.

Similarly, an example hydraulic side clamp is shown in Figure 10. In this case, the body prism has been expanded on the left side to allow for hydraulic lines and fittings.

The side clamp used in assembly pallets may be some type of simple spring-loaded device, or it may be a lateral locator pressed into an off-grid position in the base plate. This latter case, referred to as a "side-clamp-fixel," is used for the assembly pallet computations discussed in this report. The side-clamp-fixel has no body or plunger prisms, $x_{\text{travel}_{\text{min}}=0}$, $x_{\text{travel}_{\text{max}}} = d_{\text{grid}}$, and $y_{\text{offset}} = \frac{1}{2}d_{\text{grid}}$ to avoid interference with the lateral locators.

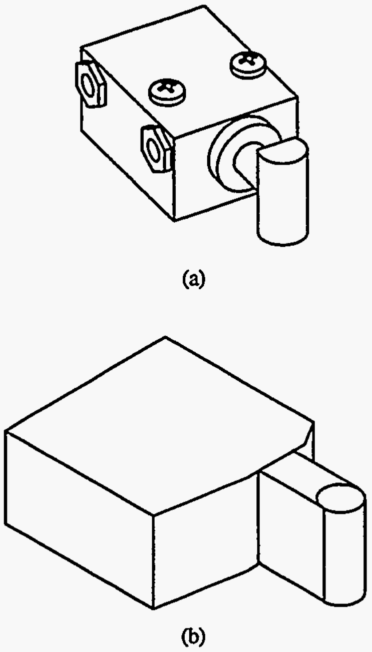


Figure 10: (a) A typical hydraulic side clamp, manufactured by Vektek. (b) Our conservative model of the clamp. The left side has been enlarged to account for hydraulic lines and fittings.

3.4 Support Pad

Support pads lift the workpiece off of the baseplate. Figure 11 shows two types of modular support pads and their associated dimensions. The support pad in the upper panel has a self-aligning contact area that is one-half the diameter of the support pad, while the support pad in the lower panel is a simple cylinder with a flat top and a contact area that is the same diameter as the support pad. The underside of both of these support pads has a threaded hole to allow mounting using threaded rods. Vertical heights are set using support spacers. A dedicated fixture may use support pads of the appropriate height and no spacers. An assembly pallet support pad is shown in Figure 12. It is a flat-topped rod of appropriate length that is pressed into the baseplate.

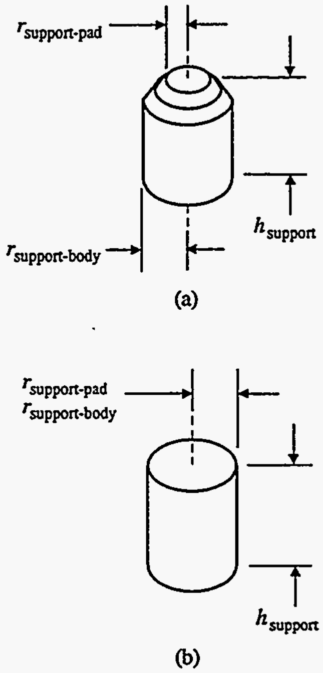


Figure 11: Modular support pads. (a) Support pad with self-aligning top (b) Support pad with flat top.

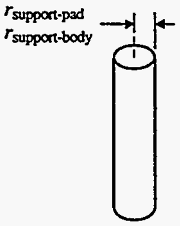


Figure 12: Assembly-pallet support pad

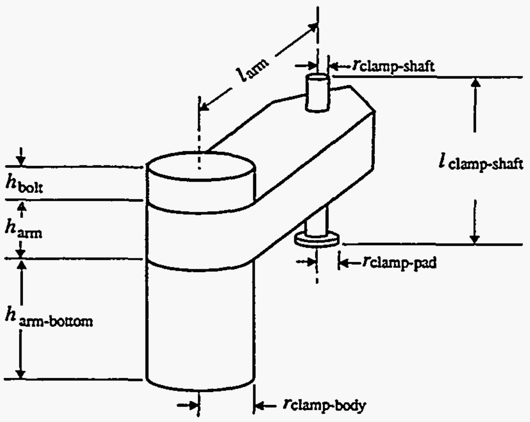


Figure 13: Top clamp dimensions. In this paper we assume a cylindrical clamp body and an angular clamp retraction motion; the algorithm may be generalized to include polyhedral body shapes as well as linear retraction motions.

3.5 Top Clamp

Top clamps restrain the workpiece from moving up. A top clamp with associated dimensions is shown in Figure 13. All components are represented as cylinders of appropriate height except for the arm which is represented as a prism with a thickness of h_{arm}. The manual top clamp used in computations presented in this report is shown in Figure 14. The upper panel is a drawing of the clamp, the middle panel shows the representation used in computations, and the lower panel shows the top-clamp arm retractions that were allowed. The manual top clamp can be retracted to any position in either direction, but this simple set of retractions proved sufficient for our work. The hydraulic top clamp used in computations is shown in Figure 15. The hydraulic top clamp can retract one-quarter of a revolution in both the clockwise and counter-clockwise directions. Each top clamp also has an optional associated clamping force F_{top}. The assembly pallets discussed in this report do not use top clamps.

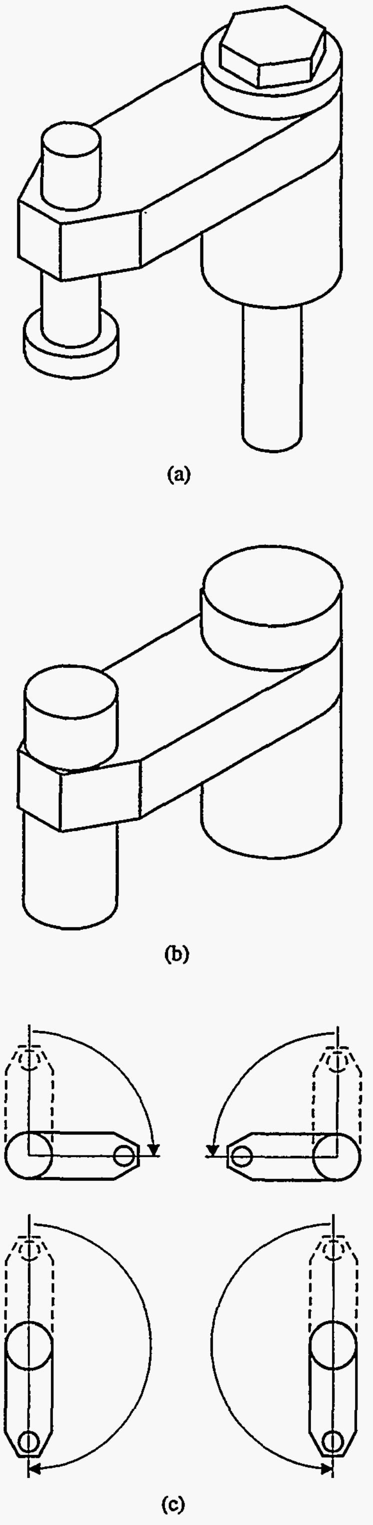


Figure 14: (a) A typical manual top clamp, manufactured by Qu-Co. (b) Our conservative model of the clamp. (c) Arm retractions.

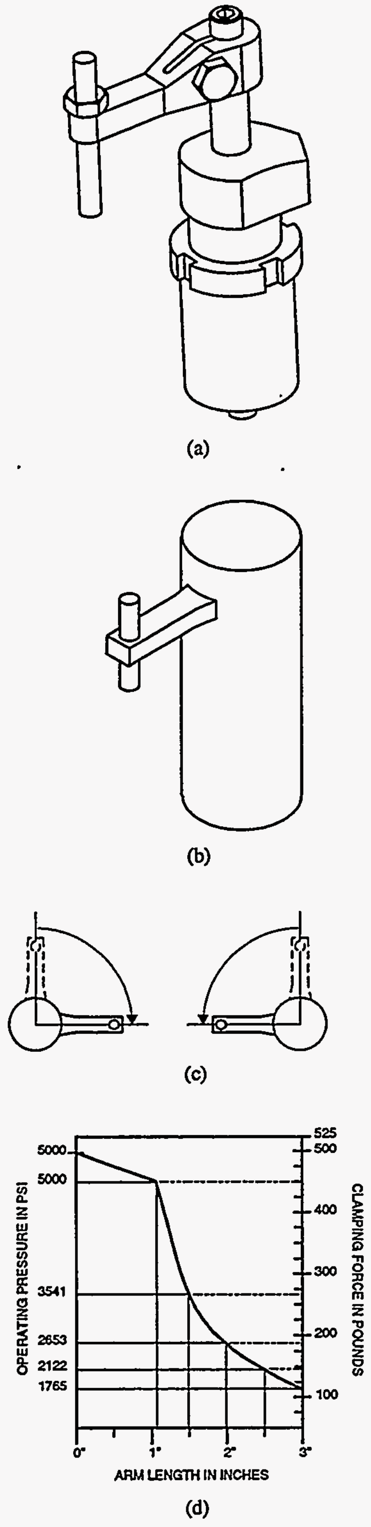


Figure 15: (a) A typical hydraulic top clamp, manufactured by Vektek. (b) Our conservative model of the clamp. (c) Arm retractions. (d) Allowable operating pressure and clamping force versus arm length.

4 Design Algorithm

The fixture design program has two major modules: The user interface and the fixture design procedure. The interface accepts user input, starts and stops the design procedure, displays results as they are generated, and allows interactive study of the results. The fixture design procedure generates fixture designs, assures that they are valid, and applies the quality metric. In the following subsections, we will focus on the details of the fixture design algorithm. Application of the quality metric in the design procedure will be discussed in this section; the details of the quality metric will be discussed in Section 5.

4.1 Problem Statement

4.1.1 Assumptions

The primary assumption we make is that the workpiece is a rigid body. The algorithm provides no explicit deformation analysis, although the form-closure robustness analysis described in Section 4.3 does reject some fixtures that are vulnerable to local shape deformations. A second key 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 8.

The fixture generation algorithm also makes the minor assumption that the workpiece should be placed at the lowest available height, provided that it does not touch the fixture plate. Raising the workpiece above this height generally degrades fixture quality by reducing the effective stiffness of the fixture elements. We disallow placement of the workpiece directly on the plate to assure deterministic workpiece location on three points of support. This restriction follows common 3-2-1 fixturing practice, but we acknowledge that in some situations it is desirable to place the workpiece directly on the fixture support surface. Extending the algorithm to allow this possibility is not particularly difficult; we leave this for future work.

Our default quality metric makes three additional assumptions. First, we assume frictionless contacts between the workpiece and the fixture elements when calculating reaction forces for the force analysis discussed in Section 5.1. This assumption is conservative in the sense that if friction is present, the available constraint will increase. Note that another part of our algorithm explictly considers friction — the loading analysis discussed in Section 5.3. Second, our position repeatability analysis assumes that out-of-plane location errors are negligible. Section 5 will discuss the implications of these assumptions. A third minor assumption is that the contact pressure distributions at fixture side contacts will minimize the induced tipping moment; this assumption is explained in Appendix B.

4.1.2 Input

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

- Workpiece $\mathcal{W}$, expressed in a volumetric shape representation that allows fetching a list of individual boundary features, generation of the workpiece silhouette when viewed along the z-axis, and which supports the `z-limit?` interference checks described below. Each boundary feature should have the following associated information:
  - A tolerance ϵ_{shape}, indicating the possible discrepancy between the position of the nominal workpiece surface and the true workpiece surface. For example, if the workpiece shape variation is represented using a minimum/maximum material condition representation, then the nominal workpiece shape should lie midway between the minimum and maximum material condition shapes, and ϵ_{shape} should be half the maximum distance between the minimum and maximum material conditions for a given feature.
  - A tolerance ϵ_{normal}, indicating the possible angular discrepancy between the nominal surface normal at a point on the feature and the true normal. Specification of ϵ_{normal} is optional, because it may be calculated from ϵ_{shape} and the locator radius.

  We also require the workpiece center of mass $(x_{\text{com}}, y_{\text{com}}, z_{\text{com}})$ and weight $w_{\mathcal{W}}$. For assembly applications this is augmented by additional information as explained in Section 6. The workpiece material is also specified, which is used in combination with the fixture element materials to lookup the friction coefficients required for loading analysis.

- Geometric access constraints $\mathcal{C}$, expressed in the same volumetric representation as $\mathcal{W}$. These constraints are defined relative to the workpiece coordinate frame.

  When analyzing fixture loadability it is convenient to break $\mathcal{C}$ into three components: $\mathcal{C}_{\text{only-loading}}$, $\mathcal{C}_{\text{only-loaded}}$, and $\mathcal{C}_{\text{always}}$, which apply only during fixture loading and unloading, only while the fixture is loaded, and at all times, respectively. An example $\mathcal{C}_{\text{only-loading}}$ constraint is the volume occupied by the gripper fingers of the manipulator used to load and unload the fixture; if the gripper is released and removed before the clamps close, then the movable clamp parts may pass through this volume. However, the clamps must not intersect this volume in the open position. An example $\mathcal{C}_{\text{only-loaded}}$ constraint is the volume swept by a cutting tool; the clamps may intersect this volume in the open position, but cannot intersect the volume once closed. An example $\mathcal{C}_{\text{always}}$ constraint is the volume occupied by a part added to an assembly; the clamps cannot intersect this volume when closed or open, because this would corrupt successful insertion or unloading from the fixture. In order to simplify the discussion below, we will define $\mathcal{C}_{\text{loading}} = \mathcal{C}_{\text{always}} \cup \mathcal{C}_{\text{only-loading}}$, and $\mathcal{C}_{\text{loaded}} = \mathcal{C}_{\text{always}} \cup \mathcal{C}_{\text{only-loaded}}$.

- A fixture kit, comprised of a base plate, lateral locators, support pads, a side clamp, and possibly top clamps. See Section 3 for further information.

- Minimum clearance limits d_{xy} and d_z, indicating the minimum distance allowed between the workpiece and non-locator fixture elements, when the workpiece is loaded. We assume that $d_{xy} \gg \epsilon_{\text{shape}}$ and $d_z \gg \epsilon_{\text{shape}}$ for all features in $\mathcal{W}$.

- Minimum contact height h_{required}, indicating the minimum vertical overlap required at a side contact.

- A quality metric comprised of three functions $\mathtt{Q}_{xy}$, $\mathtt{Q}_z$, and $\mathtt{Q}_{3d}$, which accept an xy-constraint, z-constraint, and a 3-d fixture design, respectively. These terms are defined in Section 4.2. Each function returns a scalar in $[0, 1]$ or $\emptyset$, indicating that the fixture is unacceptable. There is also an associated flag *q-combination-method* which indicates how these scores should be combined to produce an overall quality score. Possible values are `minimum` and `weighted-sum`; if the weighted sum option is selected, then weights w_{xy}, w_z, and w_{3d} are also supplied. We require that $\mathtt{Q}_{xy}$ and $\mathtt{Q}_z$ be insensitive to element spacer height, the absolute height z of $\mathcal{W}$ above the fixture plate, and the placement of the top clamp bodies; any quality analysis that considers these parameters should be placed in $\mathtt{Q}_{3d}$.

  Significant control information is embedded in this metric. For example, if a force analysis is to be performed, then this metric should include a description of the expected applied forces, as well as the maximum allowable contact reaction force.

If a tolerance analysis is to be performed, then this metric should include the locations of critical workpiece features and their associated tolerances. If a loading analysis is to be performed, then this metric should include the desired loading clearance d_{load} and a flag indicating whether the center of mass should be supported during loading. If more than one type of analysis is to be performed, then this metric should include a means of combining individual quality scores to produce the overall score. Details of our quality metric will be explained in Section 5.

- *top-clamps?*, a Boolean flag indicating whether the fixture design should include top clamps. Assembly pallets are typically designed without top clamps.

- *side-contact-height-preference*, a multi-valued flag that indicates how height values should be selected for the side locators and side clamp. Legal values are `minimum`, `coplanar`, and `optimum`. These are explained in Section 4.2.2.

- *z-constraint-on-grid?*, a Boolean flag indicating whether the supports and top clamps are restricted to regular grid locations, or whether they may be placed at arbitrary positions. If the fixture will be fabricated using a pre-drilled modular fixture plate, then this parameter should be set to true; if the fixture will be fabricated by drilling holes in a blank plate, then this parameter may be false.

- Pruning factor p, a scalar in $[0, 1]$. This parameter controls the branch-and-bound pruning employed by the algorithm, and is explained in Section 4.4.

- Several control parameters designed to eliminate the generation of poor-quality fixture designs. These include:

  - *use-non-silhouette-walls?*, which indicates whether to use all vertical faces of $\mathcal{W}$ for side contacts, or only those that correspond to the object silhouette. Setting this parameter to false reduces the number of vertical faces that must be considered by the algorithm and also limits contacts to the object periphery, which has a number of advantages.
  - d_{trim}, which defines the minimum allowable distance between a side locator or clamp contact and the edge of its bounding face. Contacts near edges are often undesirable because the workpiece is more vulnerable to deformations near its corners, shape model errors are more likely due to edge breaks and chamfers, and the corner reduces the set of initial positions from which the workpiece slips easily into place during fixture loading.
  - *max-clamp/normal-angle,* which limits the angle between the side clamp travel direction and the surface normal at the contact point. Large values of this angle can lead to amplified contact forces and jamming of the side clamp plunger.

4.1.3 Output

Given this input, the fixture design algorithm generates a series of feasible fixture designs, with associated quality scores. If $p = 0$, then this series will include all possible fixture designs; otherwise the series will correspond to a subset of the possible fixtures which is guaranteed to include the global optimum. Fixture designs are output as they are generated; once all fixture designs have been generated, the fixtures are sorted according to their overall quality scores. Every fixture design returned by the algorithm satisfies all of the following conditions:

1. The workpiece is held in form closure.

2. The form closure condition is robust in the face of local workpiece shape variations (see Section 4.3).

3. The workpiece location is deterministic in the sense proposed by [1]; that is, there is no configuration of the nominal workpiece that is close to the proper configuration where the workpiece is in contact with all of the locators. Thus, if the workpiece is placed in approximately the proper position and then moved into contact with all of the locators, its position is precisely determined.

4. No part of the fixture intersects the geometric access volumes $\mathcal{C}$.

5. All top clamps are directly above support pads.

6. The fixture may be assembled using the available fixture elements. Thus all elements are placed at legal attachment points on the plate and at heights consistent with the spacer capabilities, and all clamps make contact within their travel limits.

7. All non-locating fixture elements clear the nominal workpiece laterally and vertically by at least d_{xy} and d_z, respectively. If $\epsilon_{\text{shape}} = 0$, this condition assures that the workpiece position is determined solely by locator contacts, and not by

inadvertent contacts with other parts of the fixture. Inadvertent contact may occur if $\epsilon_{\text{shape}} \neq 0$, but as explained in Section 5.2, this does not occur in practice if $d_{xy} \gg \epsilon_{\text{shape}}$ and $d_z \gg \epsilon_{\text{shape}}$.

8. The fixture obeys all minimum-quality thresholds specified in the quality metric. For our metric, this implies:

   (a) For each expected applied force, the resulting locator, support, side and top clamp reaction forces are less than $F_{\text{limit}_{\text{locator}}}$, $F_{\text{limit}_{\text{support}}}$, $F_{\text{limit}_{\text{side}}}$, and $F_{\text{limit}_{\text{top}}}$, respectively.

   (b) For every critical workpiece feature, the worst-case position deviation is less than the associated position tolerance, considering the variability in the side locator positions and workpiece side surfaces.

   (c) The fixture may be easily loaded, meaning that: *(i)* there is a workpiece loading position that clears all fixture elements by a lateral distance of at least d_{load}, *(ii)* in the loading position, the top clamp arms may be retracted to a position that allows the workpiece to be loaded and removed vertically, clearing all fixture elements by at least d_{load}, *(iii)* the workpiece center of gravity is contained within the support triangle in both the loading position and the final loaded position, *(iv)* the side clamp can push the workpiece into the final loaded position in the fixture, without jamming due to friction or stalling the actuator. The user may choose to relax these requirements by neglecting conditions *(iii)* and/or *(iv)*, or by skipping the loading analysis entirely.

   These criteria may be selectively turned off if desired. The details of the default quality functions are discussed in Section 5.

In addition to assuring that all output fixture designs satisfy these conditions, the algorithm is guaranteed to generate the global optimum design with respect to the input quality metric and the parameter *side-contact-height-preference.* This global optimum appears at the front of the final sorted list of fixture designs. If *side-contact-height-preference* is set to `optimum`, then the returned global optimum will be the true global optimum. If *side-contact-height-preference* is set to `minimum` or `coplanar`, then the returned global optimum will be the global optimum solution within the subspace of fixture designs that are considered.

4.2 Fixture Generation

The style of fixtures considered by the algorithm allows us to decompose a fixture design into two parts: the *xy-constraint,* which is responsible for preventing motion within the xy-plane, and the *z-constraint,* which is responsible for preventing motion out of the xy-plane. As we shall see, this decomposition is very convenient for the purposes of automatic design.

The decomposition suggests a straightforward synthesis algorithm: First, generate all possible xy-constraint designs. For each resulting xy-constraint, generate all possible z-constraints that are compatible with the xy-constraint. Finally, score and sort the designs, returning the sorted list of 3-d fixture designs.

This strategy will work, but has the undesirable property that no output is returned until the full set of fixture designs is generated. In order to return initial designs as quickly as possible, our algorithm interleaves these synthesis steps so that as soon as an xy-constraint is generated, z-constraint synthesis begins for that xy-constraint. As soon as a valid z-constraint is found, the xy- and z-constraints are merged to produce a full 3-d design, which is scored and placed in the output queue. Constraints are applied as early as possible to avoid investing further synthesis effort in a doomed preliminary design. These measures complicate the control structure somewhat, but reduce the total run time and allow early output of initial designs.

In the sections that follow, we will avoid the complications of this interleaved control structure and instead simply explain each major step in the synthesis of a valid fixture design. Section 4.2.1 will describe initial pre-processing steps required to setup the synthesis algorithms. Section 4.2.2 will explain the xy-constraint synthesis method, and Section 4.2.3 will describe the z-constraint synthesis method. Finally, Section 4.2.4 will explain how xy-constraints and z-constraints are merged to form 3-d fixture designs. Discussion of fixture quality scoring will be deferred until Section 5.

4.2.1 Input Pre-Processing

The first step of the algorithm is to identify features that may be used for locating and clamping the workpiece. These features are extracted from the workpiece $\mathcal{W}$ by visiting each face $\mathcal{F}$ of $\mathcal{W}$ and constructing the subset of $\mathcal{F}$ that is suitable for location or clamping. These subsets fall into one of three categories:

- *Bottom surfaces,* which are planar surface patches which have a surface normal pointing in the $-z$ direction, and which are visible from below the object. Support pads may contact these surfaces.

- *Top surfaces,* which are planar surface patches which have a surface normal pointing in the $+z$ direction. The pads of top clamps may contact these surfaces.
- *Side surfaces,* which are vertical surface patches which have a surface normal that lies in the xy-plane, and which are visible from below the object. Lateral locators and the side clamp may contact these surfaces.

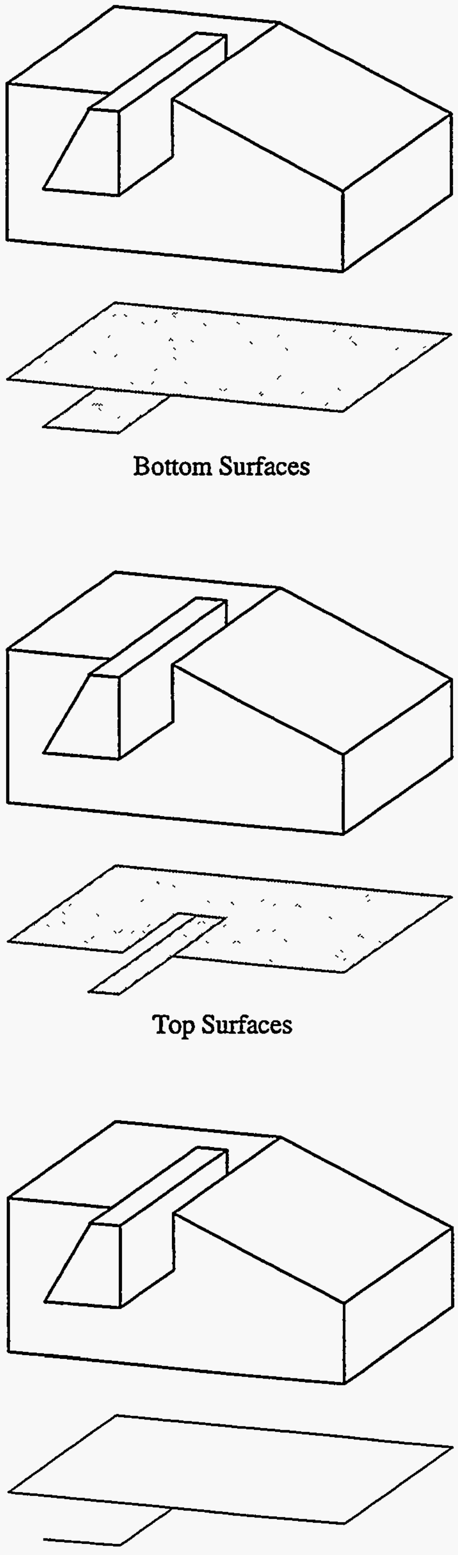
These surface patches are then projected onto the xy-plane to produce a collection of planar regions and curves (Figure 16). Any non-linear elements of these regions and curves are then converted to piecewise linear approximations, producing a collection of polygonal regions and line segments. These regions and line segments are then passed to the fixture design procedure, with appropriate pointers back to the original features in $\mathcal{W}$ so that height information can be recovered later. If *use-non-silhouette-walls?* is false, then projected edges that do not correspond to the workpiece silhouette are discarded appropriately. Similarly, projected edges that correspond to vertical faces with bounding z-values $z_{\max} - z_{\min} < h_{\text{required}}$ are also discarded.

4.2.2 Generation of xy-Constraints

An xy-constraint is comprised of three side locators and a side clamp; this ensemble of contacts prevents all motion in the xy-plane. Because the side locators and side clamp tip are all cylinders with a common radius, we refer to these contact elements as *fixels* for convenience. We synthesize xy-constraints using the planar fixture synthesis algorithm reported in [10]. We review this algorithm below, highlighting the differences between the original planar algorithm and our 3-d algorithm that synthesizes xy-constraints.

Planar Fixture Synthesis

The first step of the algorithm is to transform the projected edges by the fixel radius, which allows us to treat the fixels as ideal points (Figure 17). This process differs slightly from the growing operation reported in [10]. Because the previous algorithm accepted a simple polygon as input, it is possible to remove unreachable grown edge segments by cutting edges at all intersection points, and discarding those segments that lie on the interior of the grown polygon. For 3-d problems, the projected edges generally do not form a simple polygon, and edges cannot be deleted without analyzing reachable heights. This could be accomplished by constructing the Minkowski sum of $\mathcal{W}$ with a cylin-


Bottom Surfaces

Top Surfaces

Side Surfaces

Figure 16: Projecting candidate contact surfaces onto the xy-plane.

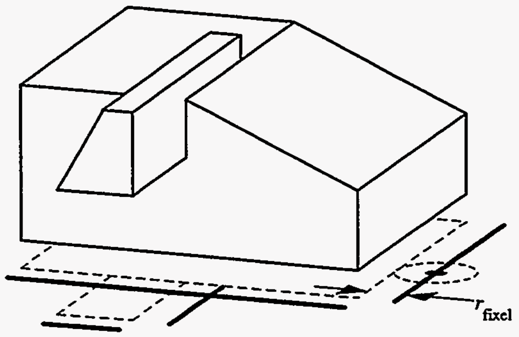


Figure 17: Growing projected edges.

der; this geometric operation is directly analogous to the growing operation employed in [10]. However, we choose instead to simply transform the projected edges by the fixel radius, and discard contacts with unreachable edges later during fixel height analysis. This approach simplifies the implementation, avoids the robustness problems that plague Minkowski sum operations, and allows the algorithm to accept CAD input that is not topologically perfect. This last feature is of significant practical importance, because many current CAD systems sometimes produce topologically incorrect representations.

As in the previous algorithm, the final step of this transformation is to reduce the length of each transformed edge by d_{trim} at both ends. This ultimately produces a grown projected edge e'_i for each projected edge e_i that is longer than $2d_{\text{trim}}$; we establish pointers between e_i and e'_i for convenience.

At this point we employ a heuristic that was not used in [10]. In order to encourage high-quality xy-constraints to appear early in the enumeration, we sort the list of projected edges according to length and direction. This sorting process first partitions the list of projected edges into four sets, according to edge direction. For example, edges with directions in $[0°, 90°)$ are placed in set 1, edges with directions in $[90°, 180°)$ are placed in set 2, etc. These direction intervals define four 90° quadrants; the orientation of this quadrant system is adjusted dynamically to distribute the edges as evenly as possible among the four quadrants. After the edges are partitioned into the four sets, each set is sorted in order of decreasing edge length. The algorithm then re-assembles the edge list, taking the longest remaining edge from each set in a round-robin fashion. This biases long edges toward the front of the list, and assures that edge directions in each of the four quadrants occur regularly.

After transforming and sorting the projected edges, we next enumerate all possible form-closure fixtures that employ three ideal point side locators, and an ideal point side clamp. This process is summarized in Figures 18–19.

The first step is to enumerate all triples of grown projected edges where a given edge appears at most twice. Then for each triple (e_a, e_b, e_c), the algorithm generates all possible *locator setups,* where each edge in the triple contacts a locator. The set of possible locator setups is constructed by placing the first locator at the origin of an infinite grid, and then using geometric information to constrain the possible locations of the second and third locators. Given that edge e_a is in contact with the first locator, the meaningful second locator positions lie within an annulus defined by l_{min} and l_{max}, the minimum and maximum distance between the grown edges e_a and e_b. We confine our attention to the first quadrant, because the other quadrants lead to symmetric fixture designs. For each resulting second locator position, the possible locations for the third locator can be similarly constrained by the intersection of two annular sectors defined by the inter-edge distances and visibility angles for the grown edge pairs (e_a, e_c) and (e_b, e_c). See Figure 18.

The above enumeration generates all edge triples, and then for each edge triple generates all possible grid positions for locator 2, and then for each position of locator 2, generates all possible grid positions for locator 3. Once all three locator positions have been determined, the algorithm calculates the (x, y, θ) position of the workpiece in contact with the locators. This is accomplished by the following function:

$$\text{pose}(e_a, e_b, e_c, x_1, y_1, r_1, x_2, y_2, r_2, x_3, y_3, r_3)$$

where (x_1, y_1), (x_2, y_2), and (x_3, y_3) are the positions of locators 1, 2, and 3, respectively, and $r_1 = r_2 = r_3$ is the fixel radius. This function returns a set of (x, y, θ) poses where each edge is in contact with its corresponding locator. If the edges are linear and $\epsilon_{\text{shape}} = 0$, then there may be up to two such poses, each of which corresponds to a distinct locator setup. We will defer discussion of the case where $\epsilon_{\text{shape}} \neq 0$ until Section 5.2. Our method of calculating these poses differs from the method reported in [10]; this new method is explained in Appendix A.

After enumerating all possible locator setups, the algorithm generates the set of all possible form-closure clamp positions for each locator setup. This is accomplished by direct construction, using an analysis in the $(F_x, F_y, \frac{\tau}{\rho})$ space of planar forces. The param-

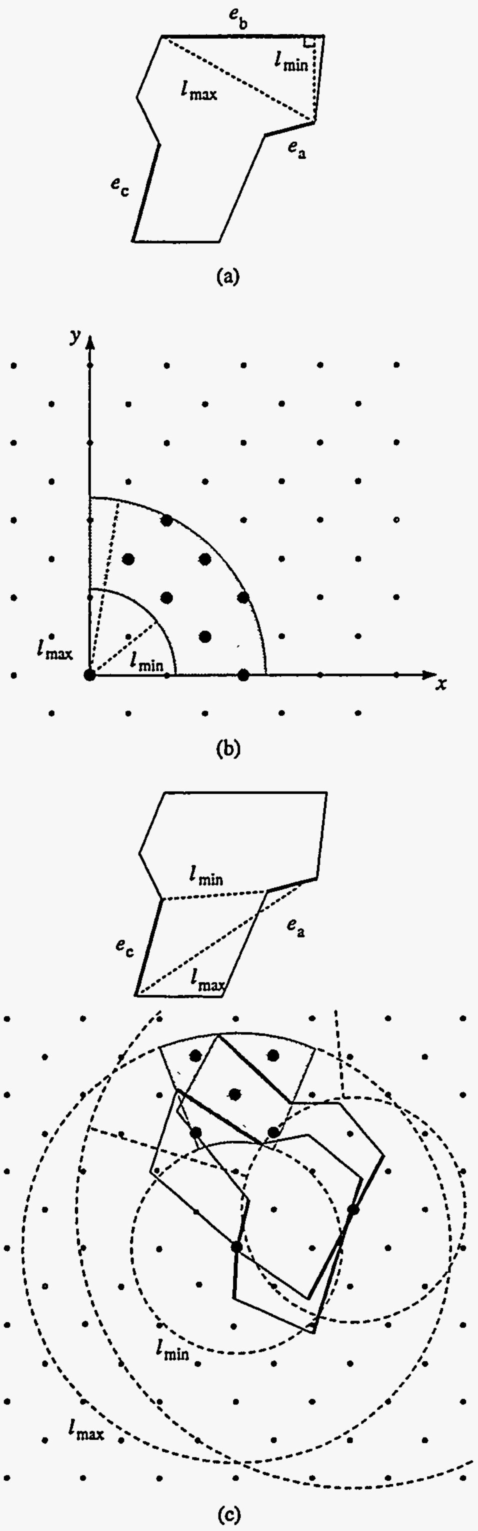


Figure 18: Generating locator setups. (a) An edge triple. (b) Possible positions for locator 2. (c) Possible positions for locator 3, given a particular locator 2 position. See [10] for a detailed explanation of this construction.

eter ρ is an arbitrary positive scalar used to assure homogeneous units for all three axes; we use one-third of the object diameter to produce well-behaved geometric calculations. The algorithm synthesizes form-closure side clamp positions by projecting the forces and torques of the locator contact normals onto the surface of a unit sphere, forming a triangle of possible forces that may be exerted on the workpiece by the locator setup. The algorithm then centrally projects this triangle onto the opposite side of the force sphere, constructing the set of forces that would produce planar form closure if exerted by the side clamp. This set is then intersected with the locus of forces that are possible through contact with a projected edge; the resulting intersection characterizes the set of all placements of the side clamp fixel that will produce form closure. These placement loci are then intersected with the side clamp x_{travel} lines determined by the grid and the side clamp y_{offset}. This produces a discrete set of candidate side clamp positions, each of which will provide form closure. See Figure 19. The algorithm then discards candidate placements that violate the *max-clamp/normal-angle* or clamp travel limits $[x_{\text{travel}_{\min}}, x_{\text{travel}_{\max}}]$, or cause interference between the clamp body or plunger and the side locators. The algorithm also discards fixtures that fail the robust form closure test described in Section 4.3; this test did not appear in [10]. Appendix C discusses some important numerical subtleties that must be addressed to perform these calculations robustly.

Height Range Analysis

At this point the algorithm in [10] checks to see if the side clamp body and plunger interfere with the input polygonal workpiece. For the 3-d problem, this test is more complicated. Because features of the workpiece may project over the top of the side clamp, interference between the workpiece and side clamp cannot be checked until the height of the xy-constraint elements is determined. Thus the next step in the algorithm is to perform a height analysis which determines the heights of the workpiece, side locators, and side clamp above the fixture base plate.

The algorithm begins by determining valid placement heights for the side locators and the side clamp tip. Each fixel has a contact surface, and possibly chamfered areas above and below the contact surface. We denote the heights of these features as h_{base}, h_{contact}, and h_{cap}, as shown in Figure 5. In the first step of the analysis, the algorithm determines the ranges of valid heights for each fixel, measured relative to the workpiece reference frame. These height ranges may

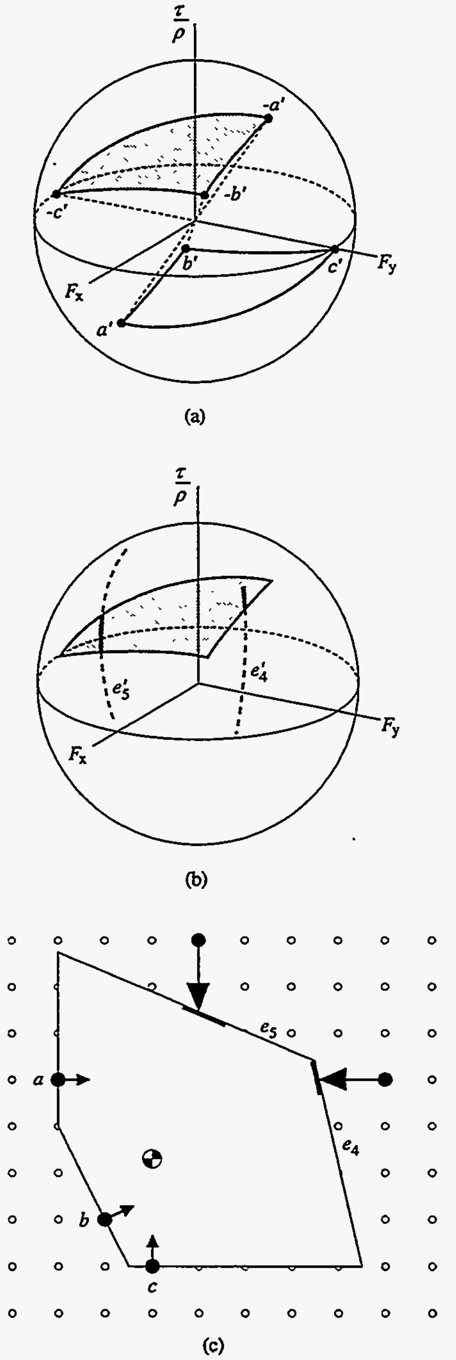


Figure 19: Computing valid side clamp positions. (a) Projecting the locator setup contact normals onto the surface of a unit sphere in force/torque space and then negating them. (b) Identifying the set of possible form-closure contact points on the force/torque sphere. (c) Reconciling the form-closure contact intervals with the grid. Locators are at positions a, b, and c.

contain negative values. Given these height ranges, the algorithm then selects a particular height for the workpiece and each fixel, assuring that the workpiece clears the plate and that each fixel spacer height is non-negative.

The first step of this height analysis is to identify the range of allowable heights for each fixel. Recall that each fixel must satisfy the minimum contact height h_{required} and minimum vertical clearance d_z. For a given fixel, this step will construct a set $\mathcal{H}$ of z-intervals that correspond to valid placements of the fixel, considering both contact overlap and interference. These z-intervals refer to the top of the fixel contact surface, measured relative to the workpiece reference frame.

At this point the fixel contact is represented as a contact between an ideal point and a grown projected edge. Using the fixel radius, the algorithm constructs the contact point between the finite-radius fixel and the original non-grown projected edge. The resulting contact point then defines a vertical line; this line is intersected with the vertical faces of $\mathcal{W}$ that generated the projected edge, producing a set of line segments delineating possible contact with $\mathcal{W}$. See Figure 20. Each line segment determines a $[z_{\text{min}}, z_{\text{max}}]$ interval of possible contact. The algorithm then converts this to an interval of possible fixel placement heights by forming $[z_{\text{min}}, z'_{\text{max}}]$, where $z'_{\text{max}} = z_{\text{max}} + h_{\text{contact}}$. This interval describes the possible fixel heights which will produce a non-zero contact overlap between the fixel and the vertical face of $\mathcal{W}$. Because we require contact overlap of at least h_{required}, we shrink this interval to form $[z'_{\text{min}}, z''_{\text{max}}]$ where $z'_{\text{min}} = z_{\text{min}} + h_{\text{required}}$ and $z''_{\text{max}} = z'_{\text{max}} - h_{\text{required}}$. Intervals that vanish during this shrinking operation are discarded.

Performing this construction produces a set $\mathcal{H} = \{[z'_{\text{min}_1}, z''_{\text{max}_1}], [z'_{\text{min}_2}, z''_{\text{max}_2}], \ldots, [z'_{\text{min}_n}, z''_{\text{max}_n}]\}$ of z-intervals describing fixel placement heights that assure sufficient contact overlap. From here on we will drop the primes, and simply use $\mathcal{H} = \{[z_{\text{min}_1}, z_{\text{max}_1}], \ldots, [z_{\text{min}_n}, z_{\text{max}_n}]\}$ to refer to the current values of $\mathcal{H}$, regardless of the number of transformations that have been applied.

We now prune this set of intervals to avoid interference with $\mathcal{W} \cup \mathcal{C}_{\text{loaded}}$ (analysis of $\mathcal{C}_{\text{loading}}$ is explained in Section 5.3). For brevity we'll abbreviate $\mathcal{W} \cup \mathcal{C}_{\text{loaded}}$ as $\mathcal{V}$, the volume to avoid. Because fixels are cylinders that rise from the fixture base plate, we can view these elements as cylinders that start at $-\infty$ and rise to the desired z height. Let's define the maximum attainable height z_{limit} as the height where the cylinder first intersects $\mathcal{V}$, neglecting intended contacts.

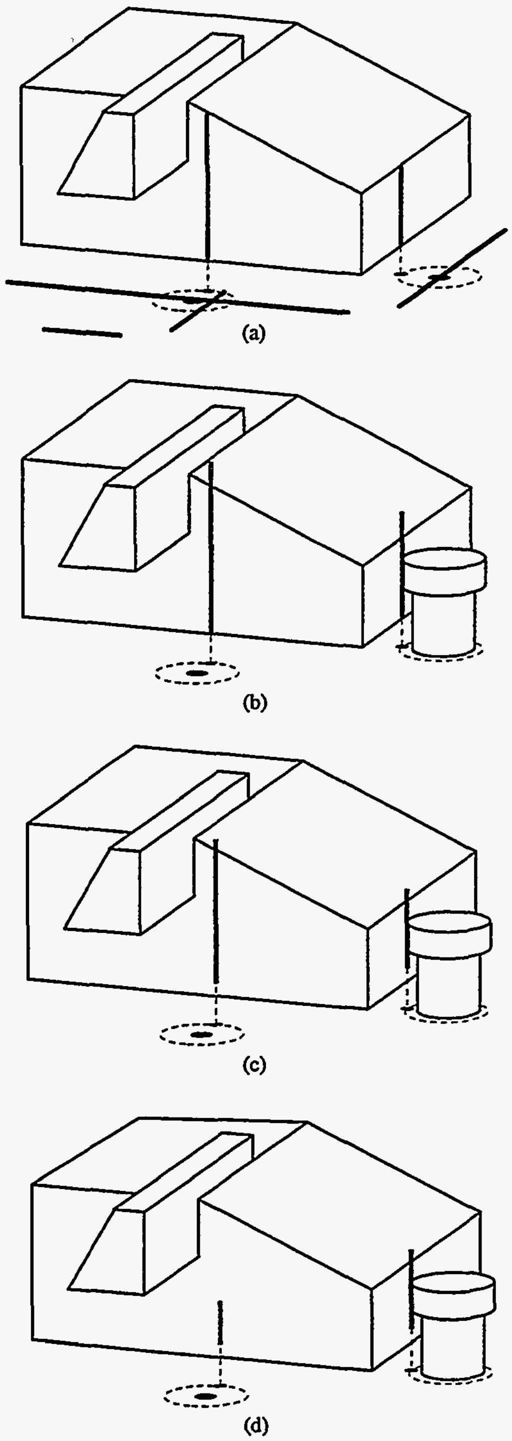


Figure 20: Determining fixel side-contact heights. (a) Possible fixel contact locations. (b) Possible fixel heights. (c) Fixel heights pruned for minimum required contact height; in this case, $h_{\text{required}} = \frac{1}{2}h_{\text{contact}}$, or half of the fixel contact surface. (d) After pruning for workpiece interference.

If we calculate z_{limit}, we obtain the height where the cylinder first collides with $\mathcal{V}$. Because we are interested in constraints on fixel placement height, we form $z'_{\text{limit}} = z_{\text{limit}} - (h_{\text{cap}} + d_z)$, which gives us the fixel placement height where interference first occurs. We can now compare z'_{limit} with z_{max_n} to determine whether $\mathcal{H}$ needs to be pruned. If $z'_{\text{limit}} \geq z_{\text{max}_n}$, then $\mathcal{H}$ remains unchanged. Otherwise, we prune $\mathcal{H}$ by intersecting it with the interval $[-\infty, z'_{\text{limit}}]$. This produces the set of valid fixel placement heights, considering both contact overlap and fixel interference.

The algorithm performs this height analysis using an equivalent but faster computation. Instead of determining z_{limit}, the algorithm checks to see whether interference occurs below z_{max_n}. This is accomplished with the following function:

$$\texttt{z-limit?}(x, y, r, z_{\text{desired}}, \mathcal{V})$$

where x, y, and r define the vertical cylinder, and z_{desired} is the maximum height of interest. This function only returns a value when interference occurs below z_{desired}; if so, then z_{limit} is returned. This avoids geometric intersection tests with features of $\mathcal{V}$ that are above z_{desired}.

To apply this function, the algorithm first transforms the fixel location into the workpiece coordinate frame, using

$$\begin{bmatrix} x'_f \\ y'_f \\ 1 \end{bmatrix} = \mathbf{T}^{-1}(x, y, \theta) \begin{bmatrix} x_f \\ y_f \\ 1 \end{bmatrix}$$

where (x_f, y_f) is the fixel position in the xy-constraint, and $\mathbf{T}^{-1}(x, y, \theta)$ is the inverse of the 2-d homogeneous transform corresponding to the workpiece pose (x, y, θ). This places the cylinder axis in the proper place in the workpiece coordinate frame. The algorithm also modifies the cylinder radius to avoid detecting intended contacts. This is accomplished by forming $r_f - \epsilon$, where r_f is the fixel radius and ϵ is slightly larger than the collision tolerance used by the `z-limit?` function. Thus the algorithm checks for fixel interference by calling:

$$\texttt{z-limit?}(x'_f,\ y'_f,\ r_f - \epsilon,\ z_{\text{max}_n} + h_{\text{cap}} + d_z,\ \mathcal{V})$$

If a non-null z_{limit} value is found, the algorithm forms $z'_{\text{limit}} = z_{\text{limit}} - (h_{\text{cap}} + d_z)$ and then replaces $\mathcal{H}$ with $\mathcal{H} \cap [-\infty, z'_{limit}]$ as described above. Otherwise, $\mathcal{H}$ does not need to be pruned.

Additional checks are required for the side clamp fixel. We model the side clamp body and plunger using a pair of conservative bounding prisms, as shown in Figures 9 and 10. The clamp body $\mathcal{B}$ is represented by a simplified enclosing volume, and the plunger $\mathcal{P}$ is represented by an enclosing volume that includes the region swept by the clamp tip during clamp actuation. Spacers may be placed underneath the clamp, raising the clamp body and plunger simultaneously. As with the side locators, these volumes may be thought of as prisms that start at $-\infty$ and rise vertically, sweeping a polygonal cross-section to some maximum height. $\mathcal{B}_c$ and $\mathcal{P}_c$ denote these cross-section polygons.

This model of clamp shape allows the algorithm to determine the maximum allowable side clamp height by detecting the first interference between $\mathcal{V}$ and these rising prisms. This is accomplished by extracting $z_{\max_n}$ from $\mathcal{H}$ and determining whether this height can be attained without interference with $\mathcal{V}$. This interference check is performed as follows:

$$\mathbf{T}_{\mathcal{B}} \leftarrow \mathbf{T}^{-1}(x, y, \theta)\mathbf{T}(x_c, y_c, \theta_c)$$

$$\mathcal{B}'_c \leftarrow \texttt{transform}(\mathcal{B}_c, \mathbf{T}_{\mathcal{B}})$$

$$z_{\text{body}} \leftarrow z_{\max_n} - h_{\text{contact}} - h_{\text{base}} + h_{\text{body}} + d_z$$

$$\texttt{prism-z-limit?}(\mathcal{B}'_c,\ d_{xy},\ z_{\text{body}},\ \mathcal{V})$$

and

$$\mathbf{T}_{\mathcal{P}} \leftarrow \mathbf{T}^{-1}(x, y, \theta)\mathbf{T}(x_c, y_c, \theta_c)\mathbf{T}(x_{\text{travel}}, y_{\text{offset}}, 0)$$

$$\mathcal{P}'_c \leftarrow \texttt{transform}(\mathcal{P}_c, \mathbf{T}_{\mathcal{P}})$$

$$z_{\text{plunger}} \leftarrow z_{\max_n} + h_{\text{cap}} + d_z$$

$$\texttt{prism-z-limit?}(\mathcal{P}'_c,\ d_{xy},\ z_{\text{plunger}},\ \mathcal{V})$$

where (x_c, y_c, θ_c) is the position of the side clamp body in the xy-constraint, x_{travel} is the clamp plunger position for this particular xy-constraint, and $\texttt{prism-z-limit?}(\mathcal{Q}, \epsilon, z_{\text{desired}}, \mathcal{V})$ is a function that returns the first z value where a prism with cross-section $\mathcal{Q}$ passes within ϵ of $\mathcal{V}$, if such an event occurs before z_{desired}. If any of these calls to `prism-z-limit?` returns a value, the algorithm appropriately prunes the set $\mathcal{H}$ for the side clamp fixel.

The algorithm thus constructs the set of valid placement heights $\mathcal{H}_i$ for each fixel F_i. If this produces any $\mathcal{H}_i = \emptyset$, then the xy-constraint is discarded.

Height Selection

Next the algorithm selects a workpiece height z above the fixture plate so that (1) the workpiece and constraint volumes clear the fixture plate by at least d_z, and (2) each fixel can be attached to the plate with a spacer of non-negative thickness. This is accomplished by identifying the minimum z value that satisfies each condition (1) and (2), and taking the maximum of these results.

To identify the minimum z that assures plate clearance, the algorithm identifies z_{bottom}, the z-value of the lowest point on $\mathcal{W} \cup \mathcal{C}_{\text{loaded}} \cup \mathcal{C}_{\text{loading}}$. The desired minimum height is then $z_1 = d_z - z_{\text{bottom}}$.

To determine the minimum z value that assures non-negative spacers at each fixel, the algorithm visits each fixel and calculates the minimum z value required for a non-negative spacer. For a given fixel F_i, the algorithm extracts $z_{\max_i}$ from the fixel's valid placement heights $\mathcal{H}_i$, which is the maximum allowable height for the top of the fixel, expressed in the workpiece coordinate frame. This value is then converted to $z'_{\max_i} = z_{\max_i} - (h_{\text{contact}} + h_{\text{base}})$, which is the maximum allowable height for the bottom of the fixel base. The minimum workpiece height which allows this fixel to be feasibly placed is then $z_{2i} = -z'_{\max_i}$. The desired overall z_2 value is then the maximum of these z_{2i} values.

The algorithm can now obtain the desired workpiece height $z = \max(z_1, z_2)$. Given this value, the algorithm constructs the set of valid spacer heights for each fixel. For each fixel F_i, this is

$$\mathcal{S}_i = [\mathcal{H}_i \oplus (z - h_{\text{contact}} - h_{\text{base}})] \cap [0, +\infty]$$

where $\mathcal{H}_i \oplus s$ denotes the operation of adding the scalar s to both endpoints of every interval in $\mathcal{H}_i$. In some situations the elements of the modular fixture kit allow only a discrete set of spacer thicknesses; in this case the algorithm replaces each $\mathcal{S}_i$ with $\mathcal{S}_i \cap \mathcal{S}_{\text{fixel}}$, where $\mathcal{S}_{\text{fixel}}$ is the set of fixel spacer thicknesses attainable using the available fixture elements. The algorithm may now set the fixel heights by simply selecting a spacer thickness from each $\mathcal{S}_i$.

Making this selection involves significant trade-offs, which depend on the user's intentions. In some cases the user may wish to minimize the height of the fixels, thus maximizing locator stiffness and position accuracy. In other cases the user may prefer fixel heights that share a common xy-plane of contact; later we shall see that this configuration allows faster search for the global optimum design. Or, the user may prefer to let the algorithm identify the combination of heights that maximizes the fixture's overall quality score. These three scenarios are specified by setting the *side-contact-height-preference* parameter to `minimum`, `coplanar`, or `optimum`, respectively.

If the input preference is `minimum`, the algorithm simply selects the minimum value from each $\mathcal{S}_i$. Note that it is acceptable for the fixels to be placed at different heights, because the resulting moments about the x- and y-axes will be opposed by the z-constraint. If the input preference is `coplanar`, then the algorithm intersects the feasible contact overlap z-intervals to identify the lowest possible common xy-plane of contact, or if no such plane exists, the plane that minimizes the maximum distance between a fixel contact and the plane. The algorithm then selects values from $\mathcal{S}_i$ to place each fixel in contact with this plane or as close to it as possible. If the input preference is `optimum`, then the algorithm discretizes each $\mathcal{S}_i$ according to a pre-set resolution, and considers all combinations of the resulting spacer choices. This creates a number of copies of the xy-constraint, each with different fixel height assignments. All of these copies are passed on to the subsequent stages of computation; this will ultimately produce the global optimum fixture design, including consideration of fixel height.

Once heights have been chosen for the workpiece and fixels, the xy-constraint is fully specified. The algorithm then passes the xy-constraint to the function $\mathtt{Q}_{xy}$ for quality scoring or elimination. If a non-$\emptyset$ value is returned, the xy-constraint is passed along to the z-constraint synthesis procedure.

4.2.3 Generation of z-Constraints

The z-constraint synthesis algorithm varies slightly depending on the values of the control parameters *top-clamps?* and *z-constraint-on-grid?*. In this section, we will focus on the case where both of these parameters are true. This corresponds to the case where the fixture is assembled with a pre-drilled modular fixture plate, and kinematic form closure is desired in all six degrees of freedom. The algorithm variations required for other values will be explained in Section 6.

When top clamps are present, a z-constraint is comprised of three supports and three top clamps, where each top clamp contacts the workpiece directly above a corresponding support. The algorithm generates z-constraints by first generating a list of all valid placements of a support/top-clamp pair, and then considering all triples formed from the resulting list. For convenience, we will denote a single support/top-clamp placement as a *z-contact*; a triple of z-contacts forms a z-constraint.

Identifying Valid z-Contacts

To identify the set of valid z-contacts, the algorithm begins by constructing the set of valid support placements, and then discarding placements that do not allow a top clamp to be placed.

When *z-constraint-on-grid?* is true, candidate support placements are readily identified, because they must coincide with the base plate grid. In order to be valid, a candidate support placement requires that the entire support pad contact a bottom surface of $\mathcal{W}$ while the support body avoids interference with $\mathcal{V} = \mathcal{W} \cup \mathcal{C}_{\text{loaded}}$. The algorithm identifies such points by visiting each bottom surface $\mathcal{F}_b$ of $\mathcal{W}$, identifying the grid points underneath $\mathcal{F}_b$, and testing the validity of each grid point.

Given a bottom surface $\mathcal{F}_b$ at height z_b, one method of generating the grid points under $\mathcal{F}_b$ would be to transform $\mathcal{F}_b$ into the xy-constraint reference frame using $\mathbf{T}(x, y, \theta)$ of the xy-constraint, find the bounding box of the resulting $\mathcal{F}'_b$, and then use two nested for loops to visit each (x_g, y_g) grid point enclosed in the bounding box, checking each point to see if it is inside $\mathcal{F}'_b$.

Our algorithm uses an equivalent method that is typically faster. Rather than incur the expense of transforming $\mathcal{F}_b$, the algorithm instead constructs a circle (x_b, y_b, r_b) enclosing $\mathcal{F}_b$; this is accomplished during pre-processing. Given an xy-constraint, this circle is transformed by $\mathbf{T}(x, y, \theta)$ to (x'_b, y'_b, r_b); this transformed circle encloses $\mathcal{F}'_b$. The algorithm then forms the bounding box $\langle [x'_b - r_b, x'_b + r_b], [y'_b - r_b, y'_b + r_b] \rangle$, and uses two nested for loops to generate enclosed grid points (x_g, y_g). Then each candidate (x_g, y_g) is transformed into the workpiece coordinate frame using the inverse pose transform $\mathbf{T}^{-1}(x, y, \theta)$, and the resulting (x'_g, y'_g) is checked against the original, untransformed $\mathcal{F}_b$. This approach is applied to all interference checks during z-constraint synthesis, saving considerable transformation of the complicated geometric features of $\mathcal{W}$ and $\mathcal{C}$.

Having thus identified a candidate support location (x'_g, y'_g), the algorithm checks to see if (x'_g, y'_g, z_b) is strictly contained in $\mathcal{F}_b$ by a distance of at least $r_{\text{support-pad}}$; if not, then the support pad does not make full contact with $\mathcal{F}_b$ and the candidate is discarded. If so, then the algorithm checks for support body interference by calling

$$\texttt{z-limit?}(x'_g,\ y'_g,\ r_{\text{support-body}} + d_{xy},\ z_b,\ \mathcal{V}).$$

If this function returns a value, then the candidate is discarded due to interference. Otherwise, (x_g, y_g) is a valid support placement.

After identifying valid support locations, the algorithm proceeds to analyze top clamp placement. We require top clamp contacts to lie directly above support points, to avoid clamp-induced workpiece deformations. Thus each valid support location corresponds to a candidate top clamp placement.

For each support location (x_g, y_g), the algorithm first determines if there exists a top surface above the support that allows valid placement of the top clamp contact pad. Using the corresponding point (x'_g, y'_g) expressed in the workpiece coordinate frame, the algorithm visits each top surface $\mathcal{F}_t$ with height z_t, and checks to see if (x'_g, y'_g, z_t) is strictly contained in $\mathcal{F}_t$ by a distance of at least $r_{\text{clamp-pad}}$. If so, then the algorithm checks for shaft interference by determining whether the bounded vertical cylinder

$$(x'_g, y'_g, r_{\text{clamp-shaft}} + d_{xy}, [z_t + \epsilon, z_t + l_{\text{clamp-shaft}} + d_z])$$

intersects $\mathcal{V}$, where ϵ is slightly larger than the collision tolerance. If no intersection is found, then (x_g, y_g) corresponds to a placement of the top clamp contact pad that satisfies both the contact and non-interference requirements.

The search for valid support pad and top clamp contact pad placements could be made more efficient by performing additional pre-processing steps: First, project each bottom and top surface $\mathcal{F}_b$ and $\mathcal{F}_t$ onto the xy-plane, producing corresponding $\mathcal{F}'_b$ and $\mathcal{F}'_t$ regions. Then for each pair $(\mathcal{F}'_b, \mathcal{F}'_t)$, construct $\mathcal{F}'_b \cap \mathcal{F}'_t$, and shrink this result by $\max(r_{\text{support-pad}}, r_{\text{clamp-pad}})$. The resulting regions would delineate all points which simultaneously satisfy the support pad and top clamp pad contact requirements. We decided not to employ this method due to the difficulty of making the required region intersection and shrinking operations robust.

At this point the algorithm has identified all grid locations which allow simultaneous placement of a support and the contact-pad shaft of the top clamp. The last step in z-contact synthesis is to discard those candidate locations for which there is no valid placement of the top clamp body and arm.

The top clamp body must be placed at a grid location; thus for every candidate z-contact (x_g, y_g), there are k possible clamp body locations (x_{g2}, y_{g2}), where k depends on the available clamp arm lengths. The algorithm considers each candidate location in turn, checking to see if the top clamp body may be placed at (x_{g2}, y_{g2}) without causing interference with the clamp body or arm. Candidate locations are visited in order of decreasing distance to the nearest workpiece edge and the first interference-free location is selected. This test is described below. As soon as any interference is detected, the candidate body placement is discarded and the next candidate placement is considered.

This interference check is performed using the conservative model shown in Figure 13. First the algorithm checks to see if the candidate body position (x_{g2}, y_{g2}) interferes with the side locators or side clamp of the xy-constraint. Because all of these elements are volumes of constant cross-section rising from the fixture base plate, these are simply 2-d intersection tests. If an intersection is found, the candidate body placement is discarded; otherwise, the algorithm proceeds to check for interference with $\mathcal{V}$.

Checking for interference between the top clamp body and $\mathcal{V}$ requires height analysis. Given that the top clamp contact-pad is placed at height z_t, the top clamp arm may be placed at a range of possible heights $[z_{\text{arm}_{\min}}, z_{\text{arm}_{\max}}] = [z_t + z_{\text{arm/pad}_{\min}}, z_t + z_{\text{arm/pad}_{\max}}]$. This in turn determines a range of possible top clamp body heights

$$\begin{aligned} z_{\text{body}_{\min}} &= z_{\text{arm}_{\min}} + h_{\text{arm}} + h_{\text{bolt}} \\ z_{\text{body}_{\max}} &= z_{\text{arm}_{\max}} + h_{\text{arm}} + h_{\text{bolt}}. \end{aligned}$$

The algorithm now determines whether the clamp body may be successfully placed by calling

$$\texttt{z-limit?}(x'_{g2},\ y'_{g2},\ r_{\text{clamp-body}} + d_{xy},\ z_{\text{body}_{\max}} + d_z,\ \mathcal{V}).$$

where (x'_{g2}, y'_{g2}) is the result of transforming (x_{g2}, y_{g2}) into the workpiece coordinate frame using $\mathbf{T}^{-1}(x, y, \theta)$. If this function returns $\emptyset$, then body placement is possible for the full range of arm heights. If instead the function returns a value $z'_{\text{body}_{\max}}$, then the algorithm compares this value to $z_{\text{body}_{\min}}$. If $z'_{\text{body}_{\max}} < z_{\text{body}_{\min}}$, then this candidate body placement is discarded. Otherwise, clamp body placement is possible for a limited range of arm heights, and the algorithm replaces $z_{\text{arm}_{\max}}$ with $z_{\text{arm}_{\max}} - (z_{\text{body}_{\max}} - z'_{\text{body}_{\max}})$.

The interval $[z_{\text{arm}_{\min}}, z_{\text{arm}_{\max}}]$ now defines a range of arm heights that allows valid placement of the clamp body. Next the algorithm checks for valid arm placement, testing for interference with the xy-constraint and $\mathcal{V}$.

The first step in this interference check is to transform the clamp arm cross-section $\mathcal{A}_c$ into the workpiece coordinate frame, using $\mathbf{T}^{-1}(x, y, \theta)\mathbf{T}(x_{g2}, y_{g2}, \theta_{\text{arm}})$, where θ_{arm} is the arm orientation determined by (x_{g2}, y_{g2}) and (x_g, y_g). This produces a transformed arm cross-section $\mathcal{A}'_c$.

The algorithm checks for interference between the clamp arm and the xy-constraint by checking for intersections between $\mathcal{A}'_c$ and the previously-transformed xy-constraint element cross-sections. If any intersection is found, then $[z_{\text{arm}_{\text{min}}}, z_{\text{arm}_{\text{max}}}]$ is pruned as required to avoid interference between the bottom of the clamp arm and the top of the fixture element.

Arm interference with $\mathcal{V}$ is now checked by calling

$z_{\text{min}} \leftarrow z_{\text{arm}_{\text{min}}} - d_z$

$z_{\text{max}} \leftarrow z_{\text{arm}_{\text{max}}} + h_{\text{arm}} + d_z$

`prism-z-ranges`$(\mathcal{A}'_c,\ d_{xy},\ z_{\text{min}},\ z_{\text{max}},\ \mathcal{V})$.

The function `prism-z-ranges`$(\mathcal{Q}, \epsilon, z_{\text{min}}, z_{\text{max}}, \mathcal{V})$ returns a set of z-intervals $\mathcal{Z} = \{[z_{\text{min}_1}, z_{\text{max}_1}], \cdots, [z_{\text{min}_n}, z_{\text{max}_n}]\}$ that describes the subset of $[z_{\text{min}}, z_{\text{max}}]$ where a prism with cross-section $\mathcal{Q}$ does not pass within ϵ of $\mathcal{V}$. Like the `prism-z-limit?` function, this function saves substantial computation by avoiding consideration of features of $\mathcal{V}$ outside the z range of interest.

Each resulting $[z_{\text{min}_i}, z_{\text{max}_i}]$ interval in $\mathcal{Z}$ is then shrunk by forming $[z_{\text{min}_i} + d_z, z_{\text{max}_i} - (h_{\text{arm}} + d_z)]$; this produces a set of intervals $\mathcal{Z}_{\text{arm}}$ describing valid arm placement heights. If $\mathcal{Z}_{\text{arm}} = \emptyset$, then this candidate body placement is discarded, and the algorithm proceeds to the next candidate body location (x_{g2}, y_{g2}). If there are no more candidate body locations, then the z-contact is discarded and the algorithm moves on to analyze the next candidate support location (x_g, y_g).

If $\mathcal{Z}_{\text{arm}} \neq \emptyset$, then the body placement is valid. In this case, the algorithm stores the body placement $\langle(x_{g2}, y_{g2}), \mathcal{Z}_{\text{arm}}\rangle$ in the z-contact data structure, and moves on to analyze the next candidate support location (x_g, y_g). Unexplored candidate body locations are also stored with the z-contact for use later as described in Section 6.1.

After completing analysis of all candidate support locations (x_g, y_g), the algorithm is left with a list of z-contacts, each of which has a valid clamp body placement. The algorithm now proceeds to formulate all triples of these z-contacts to produce z-constraints.

Triple Analysis

The algorithm generates all triples of z-contacts, testing each triple for validity. A triple is valid if the z-contact points are not collinear, and if the z-contacts do not interfere with each other.

Before generating triples of z-contacts, the algorithm employs a heuristic to encourage high-quality z-constraints to appear early in the enumeration. This heuristic is similar to the heuristic used before generating triples of grown projected edges. The algorithm partitions the z-contact list into three sets, according to the direction of the z-contact point (x'_g, y'_g) from the workpiece center of gravity $(x_{\text{com}}, y_{\text{com}})$. Each direction partition is 120° wide, but the reference direction is adjusted dynamically to distribute the z-contacts evenly among the three sectors. The resulting z-contact sets are then sorted according to distance from $(x_{\text{com}}, y_{\text{com}})$, with the furthest z-contacts appearing first. The algorithm then re-assembles the z-contact list, taking the first z-contact from each set in a round-robin fashion. This biases triples with widely separated z-contacts to appear early in the enumeration, and causes z-contact triangles that contain the center of gravity to occur regularly.

Using this sorted z-contact list, the algorithm generates all triples of z-contacts. Triples with collinear (x_g, y_g) points are discarded immediately; otherwise, the algorithm checks for interference between the z-contacts.

To determine whether the z-contacts interfere with each other, the algorithm simply checks for interference between the 2-d footprints of the z-contact elements. Interference occurs if (a) cylindrical elements A and B from two z-contacts (either supports or clamp bodies) have centers closer together than $r_A + r_B$, or (b) the distance between the center of a cylindrical element A and the cross-section of a clamp arm from another z-contact is less than r_A, or (c) two arm cross-sections intersect. If any of these conditions occurs, then the triple is discarded.

This analysis of z-contact interference may discard some triples that are actually feasible. This may occur if two clamp arms pass over each other, or if there is an alternative selection of clamp body placements that prevents interference between z-contacts. Acceptance of these cases could be supported by extending our interference test to include height information, and by using the z-contact's list of unexplored candidate body positions to generate and analyze alternative body placements. We decided not to include these features in the algorithm, because these conditions only occur when z-contacts are clustered close together. Because typical quality metrics prefer widely separated z-contacts, adding this analysis capability would cause the algorithm to invest more time analyzing triples that would later be discarded by $\mathtt{Q}_z$ or given poor quality scores. If a situation arises where $\mathtt{Q}_z$ prefers tightly clustered z-contacts, then these features should be added.

Given a triple of non-collinear z-contacts with no z-contact interference, the algorithm constructs a z-constraint and passes it to the function $\mathtt{Q}_z$ for quality scoring or elimination. If a non-$\emptyset$ value is returned, the xy-constraint and z-constraint are passed to the procedure which merges these results to form a 3-d fixture design.

4.2.4 Generation of 3-d Fixtures

At this point the algorithm has a valid xy-constraint and a valid z-constraint. Heights in the xy-constraint are expressed relative to the fixture plate, for a given workpiece height z. Heights of elements in the z-constraint are expressed relative to the workpiece coordinate frame. The xy-constraint and z-constraint are merged by simply adding z to the heights of all z-constraint elements, and choosing element spacer heights accordingly.

There are two slight complications to this construction. The first complication occurs when raising the z-constraint by z leaves some z-constraint element with a negative spacer height h_i. For example, this may occur when a support touches a bottom surface of $\mathcal{W}$ with height z_b, where $z_b + z < h_{\text{support}}$. When this occurs, the workpiece and xy-constraint fixture elements must be raised by a correction height of at least $dz = -h_i$.

If all fixture elements have continuously adjustable heights, then the algorithm simply raises $\mathcal{W}$ by dz, adjusting all spacer heights accordingly. If the fixel spacers only allow a discrete set of placement heights, then the algorithm raises $\mathcal{W}$ by $dz' = k \cdot h_{\text{spacer}}$, where h_{spacer} is the minimum fixel spacer thickness, and $k = \lceil \frac{dz}{h_{\text{spacer}}} \rceil$.

A second complication occurs if the top clamp spacers only allow a discrete set of placement heights. To determine the range of possible spacer heights for a top clamp, the algorithm transforms the range of possible clamp arm heights $\mathcal{Z}_{\text{arm}}$ by forming $\mathcal{S}_j = \mathcal{Z}_{\text{arm}} \oplus -h_{\text{arm-bottom}}$. The resulting $\mathcal{S}_j$ describes the continuous range of possible spacer heights for the top clamp. The algorithm then forms $\mathcal{S}'_j = \mathcal{S}_j \cap \mathcal{S}_{\text{top-clamp}}$, where $\mathcal{S}_{\text{top-clamp}}$ is the set of top clamp spacer thicknesses attainable using the available fixture elements. $\mathcal{S}'_j$ describes the possible range of spacer heights, including the constraints imposed by spacer thickness. The algorithm then selects a value from $\mathcal{S}'_j$ to set the top clamp height.

The complication arises when the $\mathcal{S}_j \cap \mathcal{S}_{\text{top-clamp}}$ construction causes $\mathcal{S}'_j$ to vanish. When this occurs, there are several possible responses:

- The algorithm could perform a height analysis that simultaneously considers the $\mathcal{S}_i$ and $\mathcal{S}_j$ spacer ranges of the fixels and top clamps, seeking a workpiece height z that allows valid placement of all elements,

- The algorithm could seek an alternative placement for the top clamp body that has a range of possible arm heights that produces $\mathcal{S}'_j \neq \emptyset$, or

- The algorithm could grow the clamp arm height h_{arm} by h_{spacer} during body placement analysis. This would assure that if $\mathcal{S}_j \neq \emptyset$, then for any given continuous spacer height h_s chosen from $\mathcal{S}_j$, there would exist a nearby discrete spacer height $h'_s > h_s$ which is guaranteed to avoid interference.

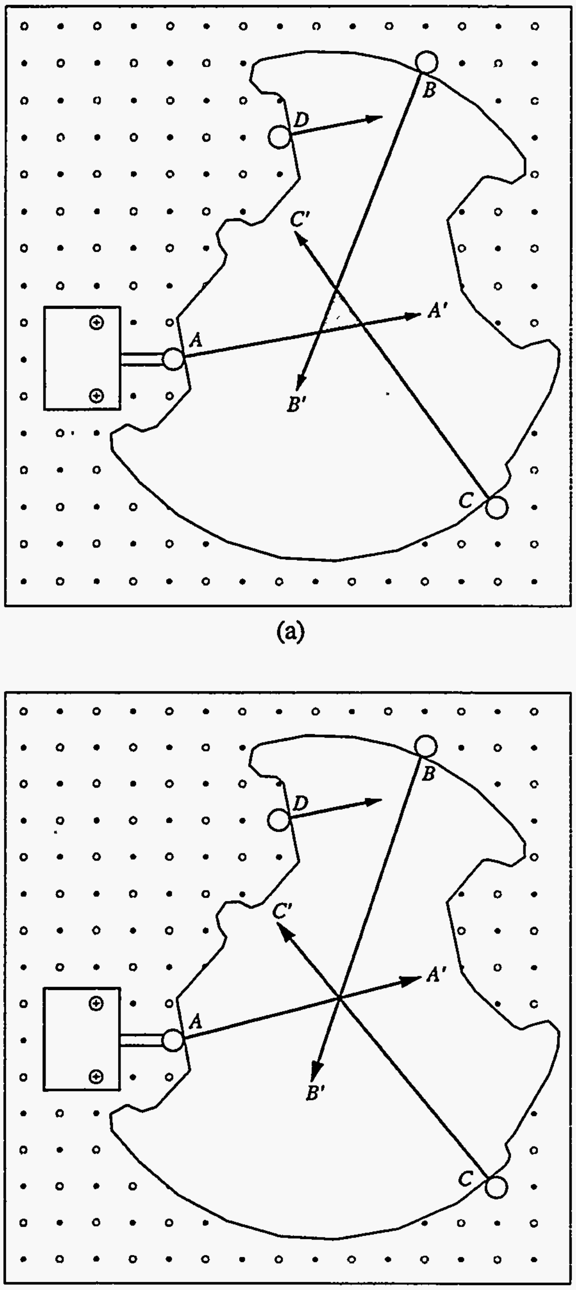
Our implementation employs the third option. Note that this issue does not arise for supports, because we require support heights to be continuously adjustable.

Once valid heights are determined for all fixture elements, a 3-d fixture is constructed that combines the xy-constraint and z-constraint. This 3-d fixture is then passed to $\mathtt{Q}_{3d}$ for final scoring or elimination. If the returned value is non-$\emptyset$, then the fixture is output to the fixture queue.

Thus far we have explained how the algorithm generates fixture designs that satisfy conditions 1 and 3–7 listed in Section 4.1.3. The remaining conditions 2 and 8 are assured by applying specific tests that we have glossed over for clarity. These tests are described in the following sections; Section 4.3 describes the robust form closure test, and Section 5 describes the tests applied during quality scoring.

4.3 Robust Form Closure

Form closure is a condition that results when the constraints imposed by a collection of contact normals combine to span the space of all possible motions. This condition is satisfied when the lines of action defined by the contact normals have certain arrangement properties [63]. This test for total constraint of a rigid body has been widely used in the robotics literature [37, 52, 42, 43].



(a)

(b)

Figure 21: A form closure condition that is not robust. (a) The nominal arrangement of contact normals. (b) The true arrangement, resulting from a discrepancy between the modeled workpiece surface normals and the true normals.

Unfortunately, the form closure condition is vulnerable to some surprising failures when there is even a small discrepancy between the ideal workpiece shape and the true workpiece shape, as shown in Figure 21. Part (a) shows a fixture design returned by an early version of our implementation, with the arrangement of the contact normals drawn for clarity. A simple analysis shows that this is clearly a form-closure fixture design: Contacts A, B, and C constrain the part so that the only possible motions are counter-clockwise rotations about points that lie inside the triangle formed by the contact normals A', B', and C'. However, contact D clearly resists these counter-clockwise rotations, and the workpiece is held in form closure.

If you build this fixture, you would expect to find the fixture to resist all planar motions of the workpiece. But in fact, the workpiece can easily rotate clockwise! Why?

The answer is shown in part (b) of Figure 21. Because of the difference between our model of the workpiece and its true shape, the directions of the true contact normals vary slightly from the expected normal directions. Because the distances between contact points are so large, this causes a drastic change in the arrangement of the contact lines of action. The triangle shown in (a) has collapsed, and now the contact normals A', B', and C' intersect at a common point. Under this arrangement of lines of action, clockwise rotations about the intersection point are not opposed.

This failure to constrain the workpiece is a consequence of the form closure test's inherent insensitivity to where the contact point lies along a given line of force. For a particular arrangement of lines of force, all constraint problems are viewed equivalently, even though the distances between contact points may vary greatly. Thus the form closure test is fundamentally insensitive to cases where the distance between contacts causes the constraint to become brittle in the face of small changes in the contact normal direction.

This is unacceptable in practice, because variations in workpiece shape and shape model errors are common. To avoid accepting a fixture design that is vulnerable to these errors, our algorithm applies a form closure robustness test to all xy-constraint designs.

The contact normal directions at each contact may vary within a range of $\pm\epsilon_{\mathrm{normal}}$; because there are four contacts, there are $2^4 = 16$ combinations of extremal contact normals. The algorithm constructs each of these sixteen combinations, testing each for form closure by mapping the contact normals onto the force sphere and checking to see if the spherical convex hull of the resulting points spans the sphere. If any combination does not provide form closure, the xy-constraint is discarded. Otherwise, the algorithm concludes that the xy-constraint is robust in the face of the anticipated shape variations.

This test of the extremal contact normals is both necessary and sufficient for robust planar form closure. It is clearly necessary, because if there exists some choice of extremal contact normals that does not provide form closure, then this is a witness that form closure is not robust. It is also sufficient, because if all combinations of extremal contact normals provide form closure, then all choices of intermediate contact normals also provide form closure.

This can be seen by considering the arrangement of contact normals on the unit sphere in (F_x, F_y, τ) space. If a contact normal is applied through a fixed point over a range of possible directions $[\nu - \epsilon, \nu + \epsilon]$, then the resulting locus of contact normal forces corresponds to a great-circle arc on the force sphere [11, 10]. Thus all instances of real contact normals correspond to points selected from four such loci. If all sixteen combinations of extremal contact normals provide form closure, then for all combinations of the endpoints of these loci, the convex-combination of the points spans the entire sphere.

Now suppose that all sixteen extremal combinations do provide form closure, but there exists some selection of intermediate contact normals which does not. This implies that there must exist some selection of contact normals which forms the critical boundary between form closure and not form closure; the convex-combination of these normals will span a hemisphere, where three of the four normals correspond to coplanar points on the force sphere. The boundary of this hemisphere is a great circle, so the great circle arc loci containing the coplanar contact normals must either be coincident with the boundary, or cross it monotonically. Neither choice is possible, because that would imply that at least one of the sixteen endpoint combinations would not provide form closure. This contradiction implies that all intermediate choices of normals must provide form closure, proving the sufficiency condition.

There is an analogous question regarding the z-constraint. These contacts constrain the workpiece to an xy-plane because of the triangle of bilateral constraints formed by the opposed support/top clamp pairs. Because these constraints force the $\hat{z}$ direction of the workpiece to coincide with the $\hat{z}$ direction of the base plate, we refer to this constraint condition as *$\hat{z}$-closure.*

This triangle of bilateral constraints may also be viewed as two triangles of unilateral contacts, one comprised of upward-pointing contact normals, and one comprised of downward-pointing normals (Figure 22).

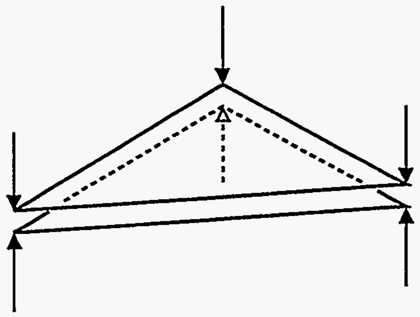


Figure 22: Contacts exerted by a z-constraint.

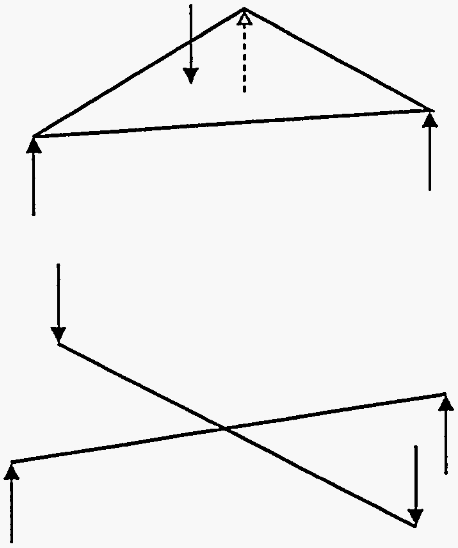


Figure 23: Contact conditions which produce $\hat{z}$-closure.

This configuration of contacts is redundant, because the minimum contacts required to produce $\hat{z}$-closure is a triangle of constraints opposed by a single constraint in the triangle interior, or a pair of opposing constraints whose connecting edges intersect (Figure 23).

From this we can gain an intuitive understanding of what would be required for $\hat{z}$-closure failure of a z-constraint design. Roughly speaking, the z-constraint will fail when the projections of the upward- and downward-pointing constraint triangles no longer intersect. This intuition fails in the general case, but adequately characterizes situations where ϵ_{normal} is small.

This intuitive view demonstrates that for reasonable values of ϵ_{normal}, our fixture designs are only vulnerable to z-constraint failure when the inner diameter of the support/clamp triangle is small compared to the vertical distance separating the supports and top clamps. Because most Q_z metrics reject fixtures with small-diameter support triangles, we have chosen not to include a $\hat{\mathbf{z}}$-closure robustness check in our implementation. If a situation arises where $\mathtt{Q}_z$ prefers tightly clustered z-contacts, then this test should be added.

4.4 Branch-and-Bound Pruning

The fixture design algorithm described thus far will enumerate all possible fixture designs, score them, and then return the sorted list. This allows the algorithm to always find the global optimum fixture design, regardless of the details of the input quality metric.

This can amount to a fairly expensive computation. Section 8 will present a detailed analysis of the algorithm's complexity, but for now let's consider a simplified model. If a given problem has N_{xy} xy-constraint designs, and an average of N_z z-constraint designs for every xy-constraint, then the algorithm will produce a total of $N_{xy}N_z$ fixture designs. This can result in hundreds of thousands of solutions for practical problems. One wonders, is it possible to do better than this brute force computation?

Producing a general-purpose algorithm that directly constructs the global optimum solution seems impossible. This is because the nature of the user's quality metric is unknown *a priori*. For example, we have encountered users that want to optimize heat dissipation properties during welding processes, or minimize the amount of paint wasted by overspray on the fixture during spray painting operations. Even in the context of our more typical default metric, variations in the set of applied forces or critical feature points greatly complicate the problem of developing general-purpose closed-form optimality models that capture the practical task requirements.

This is further compounded by the interaction of the fixture configuration and the workpiece geometry. Figure 24 shows a plot of fixture qualities generated during a typical run of the algorithm. Although the algorithm's heuristics bias high-quality fixtures to appear early in the enumeration, the sequence of quality scores appears random.

This apparent randomness suggests an approach to speeding the search for the global optimum through branch-and-bound pruning. The key idea is to avoid developing fixture designs where the quality score for the xy-constraint makes it clear that no completion of the fixture design can ever produce the global optimum. This is the case when

$$w_{xy}q_{xy} + w_z{\cdot}1 + w_{3d}{\cdot}1 \leq q_{\text{best}}$$

where q_{best} is the quality score of the best fixture output so far, and equation (4) is used to compute fixture quality, as explained later in Section 5.4. When this condition is satisfied, then no fixture design produced from this xy-constraint can beat the previous best fixture, even if $\mathtt{Q}_z$ and $\mathtt{Q}_{3d}$ return perfect quality scores. Thus, we can skip constructing z-constraints and 3d-fixture designs based on this xy-constraint, and still preserve the guarantee that the algorithm will find the

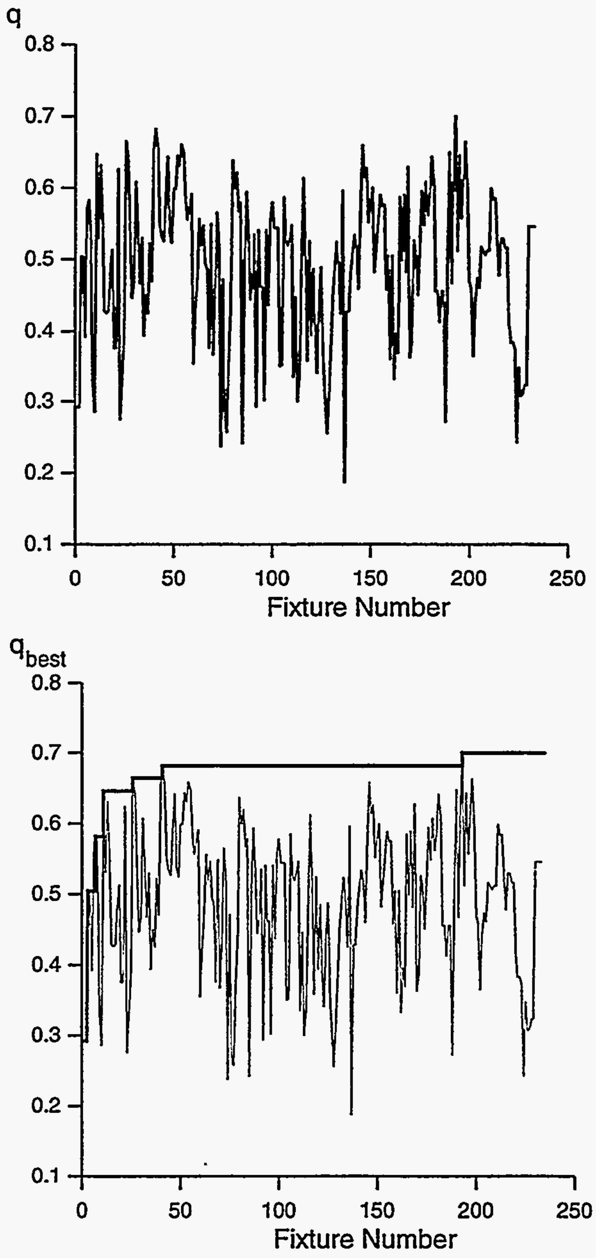


Figure 24: Quality values of fixtures generated by the synthesis algorithm. The upper plot shows the quality values over a typical run, plotted in the order that the fixtures are generated. The lower plot shows the best quality seen so far, with the fixture quality history shown for reference.

global optimum fixture design.

This pruning technique does not reduce the algorithm's worst-case complexity, but can produce significant computational savings in the average case. Section 6.1 will show how this pruning may be strengthened if z-constraint elements are allowed at off-grid locations. We have observed significant reductions in run time using these methods.

Before departing this section, we will remark in closing that our philosophy in designing this algorithm is to provide a method for generating a series of fixture designs that satisfy all pre-specified requirements, and then present the user with a set of good designs for further subjective evaluation of criteria that are not encoded in the algorithm. In some cases the pruning method is so effective that the algorithm returns a very small number of designs, thus limiting the user's choices during subjective evaluation. To address this problem, the algorithm allows the user to adjust the pruning control parameter p over the range $[0, 1]$. An xy-constraint is then pruned whenever

$$w_{xy}q_{xy} + w_z \cdot 1 + w_{3d} \cdot 1 \leq p q_{\text{best}}.$$

This allows the user to apply full pruning with $p = 1$, to gradually increase the number of returned designs by selecting values in $(0, 1)$, or to completely disable pruning by selecting $p = 0$.

5 Quality Functions

The fixture design algorithm accepts an arbitrary quality metric, comprised of three functions $\mathtt{Q}_{xy}$, $\mathtt{Q}_z$, and $\mathtt{Q}_{3d}$. The result returned by $\mathtt{Q}_{3d}$ is the ultimate score used to sort designs; in determining this score, $\mathtt{Q}_{3d}$ has access to the results of $\mathtt{Q}_{xy}$ and $\mathtt{Q}_z$, which are cached in the 3-d fixture's xy- and z-constraints, respectively. Each of these functions accepts the appropriate fixture design and returns a scalar in $[0, 1]$, or $\emptyset$, indicating that the fixture does not satisfy minimum quality standards.

A general method for producing multi-criteria quality metrics is to combine individual quality metrics into a single function which applies each quality metric in turn, returning $\emptyset$ as soon as any individual metric returns $\emptyset$. If all of the individual metrics return scalar values, then these values are combined to produce an overall score in $[0, 1]$. Plausible combination methods include weighted sum or the minimum of all individual scores.

The following sections describe our default quality metric, which analyzes the fixture aspects outlined in Section 4.1.3. This quality metric considers a fixture's ability to resist applied forces without exerting excessive reaction forces on the workpiece, the position repeatability of critical workpiece features, and the ease of loading the fixture. The following sections describe each of these analyses.

5.1 Force-Based

Our force-based quality metric seeks to minimize the reaction force required to resist expected applied forces. Embedded in this metric is an input list of forces expected during task execution, expressed as $\mathbf{F}_i = [F_x\ F_y\ F_z\ \tau_x\ \tau_y\ \tau_z]_i$ vectors, as well as the input force limits $F_{\text{limit}_{\text{locator}}}$, $F_{\text{limit}_{\text{side}}}$, $F_{\text{limit}_{\text{support}}}$, and $F_{\text{limit}_{\text{top}}}$, which describe the maximum allowable reaction force at each fixture element.

Given a fixture design to evaluate, the quality calculations within $\mathtt{Q}_{xy_F}$, $\mathtt{Q}_{z_F}$, and $\mathtt{Q}_{3d_F}$ are essentially the same. These functions consider each expected applied force, and compute the contact reaction force at each fixture element required to resist the expected force. $\mathtt{Q}_{xy_F}$ considers only the (F_x, F_y, τ_z) components of the applied force, and $\mathtt{Q}_{z_F}$ considers only the (F_z, τ_x, τ_y) components. In some cases $\mathtt{Q}_{3d_F}$ must repeat this analysis, considering all six components of force. These calculations are explained in Appendix B.

After calculating the reaction forces required to resist each expected applied force, the metric identifies the maximum reaction force F_{max} occurring at each fixture element. The metric then compares this force with the appropriate force limit F_{limit} and ideal force F_{ideal}, which is a lower bound on F_{max} which we will explain momentarily. If $F_{\text{max}} > F_{\text{limit}}$, the quality metric returns $\emptyset$. Otherwise, the force-based quality score is determined by:

$$q_F = 1 - \left(\frac{F_{\text{max}} - F_{\text{ideal}}}{F_{\text{limit}} - F_{\text{ideal}}}\right)$$

The value of this function is 1 when $F_{\text{max}} = F_{\text{ideal}}$, and 0 when $F_{\text{max}} = F_{\text{limit}}$.

The value of the lower bound F_{ideal} is determined during pre-processing. The value used for F_{ideal} must be smaller than the minimum possible value of the maximum contact reaction force.

One adequate choice is to set $F_{\text{ideal}} = 0$. Because contact reaction forces are never negative, this is a correct lower bound on the maximum contact reaction force. However, if $\mathbf{F}_i \neq 0$, we know that this lower bound will never be attained. This will produce artificially low quality scores, skewing the information fed back to the user and reducing the effectiveness of the pruning techniques explained in Section 4.4.

An alternative strategy is to base F_{ideal} on the expected applied forces. The basic idea is to characterize the contact reaction force that would occur in the best-case fixture design for each applied force, and then take the maximum of the resulting values. For example, if $\mathbf{F}_i$ is a downward force of 100lb exerted at some point $\mathbf{p}_i$, then the best-case z-constraint would be a tripod of supports distributed evenly around $\mathbf{p}_i$, each having a contact reaction force of 33lb. The existence of this applied force therefore implies that no fixture design will yield a maximum support reaction force of less than 33lb, so $F_{\text{ideal}_{\text{support}}} = 33\text{lb}$ is a correct lower bound.

If we consider only the translational components of the expected applied forces, then we can easily compute the required lower bounds:

$$\begin{aligned}
F_{\text{ideal}_{\text{locator}}} &= \frac{F_{xy_{\text{max}}}}{3} \\
F_{\text{ideal}_{\text{side}}} &= \frac{F_{xy_{\text{max}}}}{3} \\
F_{\text{ideal}_{\text{support}}} &= \frac{F_{-z_{\text{max}}}}{3} \\
F_{\text{ideal}_{\text{top}}} &= \frac{F_{+z_{\text{max}}}}{3}
\end{aligned}$$

where $F_{xy_{\text{max}}}$ is the maximum magnitude of a vector $[F_x\ F_y]$, drawn from a single input force $\mathbf{F}_i$, and $F_{-z_{\text{max}}}$

and $F_{+z_{\max}}$ are the magnitudes of the maximum negative and positive F_z components of an input force $\mathbf{F}_i$. If desired, we could extend these calculations to include the applied moments as well, by factoring in the maximum workpiece diameter.

The user has the option of selecting $F_{\text{ideal}_{\text{side}}} = 0$ to indicate a preference for fixtures that minimize the side clamp's role in resisting applied forces. This is good practice in some situations, because the locators are generally more rigid than the side clamp.

These expected forces, force limits, and ideal forces are embedded in the functions $\mathtt{Q}_{xy_F}$, $\mathtt{Q}_{z_F}$, and $\mathtt{Q}_{3d_F}$, which become part of the overall quality metric functions explained in Section 5.4.

In $\mathtt{Q}_{xy_F}$, two values of q_F are calculated, one for the side locators, and one for the side clamp; the minimum of these values is returned. Similarly, $\mathtt{Q}_{z_F}$ obtains separate q_F values for the supports and top clamps, and returns the minimum of the two results. When $\mathtt{Q}_{3d_F}$ performs a full 6-dof force analysis, the resulting element-wise q_F values are combined using a method that is equivalent to the method used to calculate and combine the $\mathtt{Q}_{xy_F}$ and $\mathtt{Q}_{z_F}$ results.

The force analysis described in Appendix B assumes frictionless contacts; this has both positive and negative implications. On the plus side, the presence of friction does not compromise the fixture's form closure constraint of $\mathcal{W}$, but instead augments the constraint. On the other hand, friction can lead to significant internal forces, which could give rise to larger contact reaction forces than those anticipated by the quality metric. Resolving this problem is fundamentally impossible, because contact reaction forces in the presence of friction are statically indeterminate for the class of fixtures we consider.

5.2 Position-Based

Our position-based quality metric seeks to minimize the maximum variation in position of critical workpiece features when the workpiece is properly loaded in the fixture. Embedded in this metric is an input list of critical workpiece features and associated position tolerances, expressed as $\mathbf{P}_i = \langle (x,y,z)\ \epsilon_{xy}\ \epsilon_z \rangle_i$ vectors, where (x,y,z) is the critical workpiece point, ϵ_{xy} is the maximum allowable lateral position deviation, and ϵ_z is the maximum allowable vertical position deviation.

When the workpiece is loaded into the fixture, the true locations of these critical points will not exactly match the locations predicted by the workpiece pose (x,y,θ). These position deviations result from variations in workpiece shape, and variations in the position of the locator contact surfaces. This metric estimates the maximum possible magnitude of these position deviations, using a purely planar analysis. Because the analysis is planar, it may be accomplished entirely by $\mathtt{Q}_{xy}$ if the locator position tolerance $\epsilon_{\text{locator}}$ is independent of locator height. If $\epsilon_{\text{locator}}$ varies with locator height, then this analysis must be performed in $\mathtt{Q}_{3d}$, because locator heights may change when the xy-constraint and z-constraint are merged.

During pre-processing, the metric computes a lower bound on the maximum position deviation of a critical point. In our implementation, this is simply $d_{\text{ideal}} = \epsilon_{\text{locator}}$, which indicates the position deviation that would result if contact variations caused $\mathcal{W}$ to purely translate. A tighter bound would be $d_{\text{ideal}} = \epsilon_{\text{locator}} + \epsilon_{\text{shape}_{\min}}$, where $\epsilon_{\text{shape}_{\min}}$ is the smallest ϵ_{shape} in $\mathcal{W}$.

These critical points and d_{ideal} are embedded in the functions $\mathtt{Q}_{xy_P}$ or $\mathtt{Q}_{3d_P}$, which become part of the overall quality metric functions in Section 5.4.

To determine a quality score that captures a fixture's position repeatability, the metric first calculates a set of workpiece poses $\{(x,y,\theta)_j\}$ that are instances of worst-case variations in workpiece position. This is accomplished by first determining the contact tolerance $\epsilon_{\text{contact}_k} = \epsilon_{\text{locator}_k} + \epsilon_{\text{shape}_k}$ for each locator, where $\epsilon_{\text{shape}_k}$ is the shape tolerance of the workpiece feature contacting locator k. Determining a separate contact tolerance for each locator allows the metric to discriminate between contacts with more precise and less precise workpiece surfaces, while simultaneously considering the effect of fixture configuration and workpiece geometry. The set of example worst-case workpiece poses is then constructed by calling

$$\mathtt{pose}(e_a, e_b, e_c, x_1, y_1, r_1', x_2, y_2, r_2', x_3, y_3, r_3')$$

for all eight choices of $r_k' = r_k \pm \epsilon_{\text{contact}_k}$. This produces a set of eight $(x,y,\theta)_j$ poses that are instances of worst-case deviations in the pose of $\mathcal{W}$. If $\mathtt{pose}$ returns multiple values, the metric uses the pose closest to the nominal location of $\mathcal{W}$, because the other poses correspond to improper loading of the fixture.

Next the metric considers each critical point $\mathbf{P}_i$, and computes its nominal location $(x,y)'_i$, using the nominal workpiece pose transformation $\mathbf{T}(x,y,\theta)$. The algorithm similarly computes the critical point location for each of the eight worst-case poses, producing a set of deviated positions $\{(x,y)'_{ij}\}$. The algorithm then finds the maximum distance $d_{\max_i}$ between $(x,y)'_i$ and the points in $\{(x,y)'_{ij}\}$. If $d_{\max_i} > \epsilon_{xy_i}$, then the metric exits immediately, returning $\emptyset$. Otherwise, the metric calculates

$$q_{P_i} = 1 - \left(\frac{d_{\max_i} - d_{\text{ideal}}}{\epsilon_{xy_i} - d_{\text{ideal}}} \right)$$

The value of this function is one when $d_{\max_i} = d_{\text{ideal}}$, and zero when $d_{\max_i} = \epsilon_{xy_i}$. After considering each critical point $\mathbf{P}_i$, the ultimate value q_P returned by the metric is the minimum of the resulting q_{P_i} values.

This calculation method is an approximation, for several reasons. First, the analysis is purely planar, which causes the z and ϵ_z components of each critical point to be ignored. Consequently we do not consider uncertainty in the support/workpiece contacts. Uncertainty in these contacts can cause out-of-plane pose deviations, producing critical point deviations in both xy and z. These effects have a small influence on xy position if the workpiece height is small relative to the inner diameter of the z-constraint support triangle.

The second approximation arises from our finite sampling of possible workpiece poses. Because we only sample a discrete set of example worst-case contact deviations, it may be possible to find a combination of r'_k values that yields a larger $d_{\max_i}$ for some critical point.

Even with these approximations, this metric provides valuable information by discarding fixture designs that are known to violate the position tolerances, and by providing a means to compare fixture designs based on position repeatability. Further improvements to this metric are an appropriate area for future work.

This metric also illuminates a characteristic of the fixture generation algorithm. During fixture generation the clearances d_{xy} and d_z are assured using the nominal workpiece pose (x, y, θ), and the nominal workpiece shape. In reality, the true workpiece position and shape will vary from these nominal values, which may cause the true clearance between the workpiece and non-locating fixture elements to be less than d_{xy} and d_z, possibly even leading to inadvertent contacts. This is why condition 7 of Section 4.1.3 only asserts clearance for the nominal workpiece.

It is tempting to propose pre-processing analysis which will compute enlarged d'_{xy} and d'_z values, such that if the algorithm assures that the nominal workpiece clears non-locating elements by the enlarged distances, then the true workpiece will clear non-locating elements by the original clearances d_{xy} and d_z. While appealing, this approach is very difficult, and may be impossible. Consider the example shown in Figure 25; in this fixture, very small $\epsilon_{\text{contact}}$ values produce very large deviations in workpiece position; these deviations result from a subtle interaction of the workpiece geometry and fixture configuration.

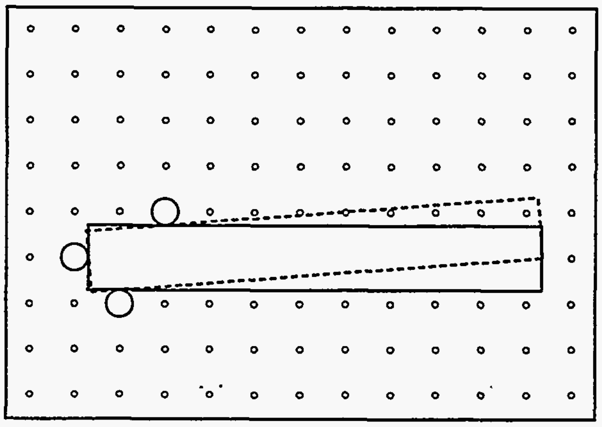


Figure 25: A case where small errors in workpiece shape cause large errors in workpiece position. The workpiece drawn with dashed-lines is slightly narrower than the workpiece drawn with solid-lines.

However, the good news is that the position-based quality metric can prevent these problems from occurring in practice. Because fixtures like the example in Figure 25 produce large $d_{\max_i}$ values, they are discarded by the metric, which prefers fixtures with small $d_{\max_i}$ values. Thus if $d_{xy} \gg \epsilon_{\text{shape}}$ and $d_z \gg \epsilon_{\text{shape}}$, then inadvertent contacts between the true workpiece and non-locating fixture elements are very unlikely for fixture designs that are given high quality scores. This can be rigorously enforced by augmenting the quality metric input to include additional $\mathbf{P}_i$ critical points $\langle (x,y)\ d_{xy}\ d_z \rangle$, where the (x,y) points are sampled densely on a surface $\mathcal{W}'$ that conservatively encloses $\mathcal{W}$. Fixture designs will then only be accepted by the quality metric if these $\mathcal{W}'$ surface points move less than d_{xy} and d_z due to workpiece and fixture variations. Because $\mathcal{W}$ lies entirely on the interior of $\mathcal{W}'$, all points in $\mathcal{W}$ must then move less than d_{xy} and d_z. This strategy is guaranteed to provide positive workpiece clearance, up to the precision of our method for calculating $d_{\max_i}$ and the sampling of points on $\mathcal{W}'$.

Our implementation does not employ this strategy, because we felt that the extra rigor was not worth the additional computational complexity. Instead, a more practical approach might be to apply a post-processing filter that performs this and other expensive verification tests after a preferred fixture has been selected by the user.

5.3 Loading-Based

Our loading-based quality metric prefers fixtures that are easy to load over those that are not. This quality metric returns a quality score q_L of 1 if the fixture is easy to load, and $\emptyset$ or zero if it is not, depending on whether the user strictly requires easy loading or merely prefers it.

The loading analysis considers a canonical loading strategy comprised of three steps: First, a manipulator lowers the workpiece vertically onto the supports with the clamps open. Second, the side clamp closes, pushing the workpiece into place against the three side locators. Third, the top clamps close, swinging into place and then moving downward to clamp the workpiece. This strategy is appealing in that it can be accomplished by simple pick-and-place automation hardware, and is also convenient for humans to perform.

With this strategy in mind, our metric considers a fixture design easy to load if the following conditions are met:

(i) There exists a loading pose $(x, y, \theta)_L$ for which each locator clears its corresponding side surface by a lateral distance d_{load}. Further, when the workpiece is placed in the fixture in the loading pose, the z-constraint supports remain in contact with their corresponding bottom surfaces of $\mathcal{W}$, and $\mathcal{V}_L = \mathcal{W} \cup \mathcal{C}_{\text{loading}}$ clears all fixture elements laterally by at least d_{load}.

(ii) There exists a selection of top clamp arm retractions that (a) allows the workpiece to be lifted vertically out of the fixture from the loading pose, assuring that $\mathcal{W} \cup \mathcal{C}_{\text{loading}}$ clears all clamp arms laterally by at least d_{load}, (b) allows the clamp arms to move to the closed position when the workpiece is in the (x, y, θ) loaded pose, clearing $\mathcal{W} \cup \mathcal{C}_{\text{loaded}}$ laterally by at least d_{xy}, and (c) avoids mutual interference of the arm swept volumes.

(iii) The workpiece center of gravity is within the support triangle in both the (x, y, θ) loaded position and the $(x, y, \theta)_L$ loading position. The user may choose to ignore this condition if, for example, the manipulator retains its grasp of $\mathcal{W}$ until the clamps are closed.

(iv) For a sampling of intermediate contact configurations ranging from the loading position $(x, y, \theta)_L$ to the final position (x, y, θ), the workpiece does not encounter a condition where the clamp is incapable of pushing the workpiece into place. That is, no configuration is found where jamming due to friction could prevent motion for arbitrary clamp thrust forces, or where the friction is large enough to stall the clamp given its actual thrust force.

Notice that the first two conditions are geometric, while the second two are driven by the physical requirements of the loading strategy. An alternative strategy would be to have the manipulator move the workpiece into place against the locators using a lateral compliant motion; see [80] for an example for such a strategy. Our implementation allows the user to disable the tests for conditions *(iii)* and *(iv)* if this strategy is preferred.

Given a candidate fixture design, the analysis of these conditions is spread over the functions $\mathtt{Q}_{xy_L}$, $\mathtt{Q}_{z_L}$, and $\mathtt{Q}_{3d_L}$. The following sections describe this computation.

5.3.1 Geometric Loading Analysis

The first step of the loading analysis is to verify that the workpiece can be lowered into the fixture without interference. In this analysis, we neglect the effects of manipulator orientation uncertainty. If d_{load} is chosen to equal the manipulator position uncertainty ϵ_{xy}, then errors in workpiece orientation may cause portions of $\mathcal{V}_L$ to depart from their nominal locations by distances greater than d_{load}. This may be rectified by choosing $d_{\text{load}} = \epsilon_{xy} + d_{\mathcal{V}_L} \cdot \epsilon_\theta$, where ϵ_θ is the manipulator orientation uncertainty and $d_{\mathcal{V}_L}$ is the maximum diameter of $\mathcal{V}_L$. If the grasp point on $\mathcal{W}$ is known, then this value may be refined to $d_{\text{load}} = \epsilon_{xy} + r_{\mathcal{V}_L} \cdot \epsilon_\theta$, where $r_{\mathcal{V}_L}$ is the maximum lateral distance between the grasp point and a point in $\mathcal{V}_L$. These techniques will produce conservative results; an exact test may be implemented by employing the techniques described in [8].

From here on we will assume that d_{load} has been chosen to be large enough to cover the effects of orientation uncertainty. This parameter is embedded in the loading analysis quality functions $\mathtt{Q}_{xy_L}$, $\mathtt{Q}_{z_L}$, and $\mathtt{Q}_{3d_L}$, which become part of the overall quality metric functions in Section 5.4.

Loading Pose

The vertical loading motion is executed with a manipulator in the real world, which means that the motion will be imperfect and subject to variation. Thus, the (x, y, θ) target position of this vertical motion should not be the exact loaded position of the workpiece in the fixture; if this point were chosen, motion errors could cause the part to collide with the side locators.

To avoid these collisions, the algorithm calculates a loading position that clears all side locators by d_{load}.

This loading pose $(x, y, \theta)_L$ is calculated in the $\mathtt{Q}_{xy_L}$ function, which calls

$$\mathtt{pose}(e_a, e_b, e_c, x_1, y_1, r_1', x_2, y_2, r_2', x_3, y_3, r_3')$$

where $r_i' = r_i + d_{\text{load}}$. If more than one pose is returned, the function accepts the pose closest to the loaded position (x, y, θ). The resulting $(x, y, \theta)_L$ loading pose assures that the contact features of $\mathcal{W}$ clear the locators by d_{load}; however, this clearance does not apply to other features of $\mathcal{W}$.

Thus the metric verifies that $\mathcal{V}_L$ in the loading pose clears all of the xy-constraint fixture elements by at least d_{load}, with the side clamp plunger in the retracted position $x_{\text{retracted}}$. This interference test is performed using the `z-limit?` and `prism-z-limit?` functions described above. If any fixture element does not clear $\mathcal{V}_L$ by d_{load}, then $\mathtt{Q}_{xy_L}$ returns $\emptyset$; otherwise, $\mathtt{Q}_{xy_L}$ returns the result of the physical loading analysis described in Section 5.3.2.

The function $\mathtt{Q}_{z_L}$ performs two tests on the loading pose. The first test verifies that the z-constraint support pads are still contained in their bottom surfaces when $\mathcal{W}$ is in the loading pose. The second test verifies that $\mathcal{V}_L$ in the loading pose clears the support bodies by at least d_{load}. A loading pose that passes these tests meets the requirements of *(i)*.

Top Clamp Clearance

Analysis of top clamp clearance and arm retraction is performed in the $\mathtt{Q}_{3d_L}$ function, because as we shall see in Section 6.1, when *z-constraint-on-grid?* is false, the top clamp body positions are not determined until the xy-constraint and z-constraint are merged to form a 3-d fixture design.

The function $\mathtt{Q}_{3d_L}$ verifies that $\mathcal{V}_L$ in the loading pose clears the top clamp bodies by at least d_{load}, and then performs arm retraction analysis. Each top clamp has a list of associated retraction specifications, as shown in Figures 14(c) and 15(c). In the retraction analysis, $\mathtt{Q}_{3d_L}$ selects a retraction for each top clamp, so that the retraction combination satisfies condition *(ii)* above. If no satisfactory combination can be found, then $\mathtt{Q}_{3d_L}$ returns $\emptyset$; otherwise $\mathtt{Q}_{3d_L}$ returns 1.

This retraction analysis method is an approximation in situations where the top clamp may be retracted along a continuous range of angles, as in the case of the Qu-Co manual top clamp. In our implementation we elected to discretize this range of angles for simplicity; a more complete analysis could be developed in the future.

5.3.2 Physical Loading Analysis

The tasks described thus far assure that conditions *(i)* and *(ii)* above are satisfied. In addition to these geometric tests, the algorithm also checks the physical characteristics of the loading operation.

Once the workpiece has been lowered into the fixture in the loading pose, the side clamp closes to push the workpiece into place against the side locators. This strategy will fail if the workpiece tips over before the top clamps lock it down, or if the side clamp fails to push the workpiece into its final loaded position. The following section describes how the loading analysis addresses these problems.

Workpiece CG Support

The loading analysis checks to see if the workpiece will rest stably in both the loading and loaded poses without tipping. The function $\mathtt{Q}_{z_L}$ performs two tests to verify this condition. The first test verifies that when the workpiece is in the loading pose, its center of gravity $(x_{\text{com}}, y_{\text{com}})$ is contained within the z-constraint support triangle, shrunk by d_{load}. This assures that the workpiece will be supported after the vertical loading motion, despite the presence of manipulator position uncertainty. The second test verifies that when the workpiece is in the (x, y, θ) loaded pose, its center of gravity $(x_{\text{com}}, y_{\text{com}})$ is contained within the z-constraint support triangle, without shrinking. If either test fails, then $\mathtt{Q}_{z_L}$ returns $\emptyset$; otherwise, $\mathtt{Q}_{z_L}$ returns 1.

It is tempting to suggest a more liberal version of the CG-support test that checks $(x_{\text{com}}, y_{\text{com}})$ against a support region formed by growing the support triangle by the support pad radius $r_{\text{support-pad}}$. This would be correct for solid support pads, but incorrect for self-aligning support pads whose contact surfaces tip to accommodate surfaces that are not perfectly horizontal. In our implementation, we decided to simply test against the support triangle, thus avoiding the need to distinguish support pad types.

Push-Into-Place Analysis

In a successful loading operation, the side clamp pushes the workpiece into place in contact with the three side locators. This operation may fail for a number of reasons:

- *Jamming.* The workpiece jams due to friction before reaching the desired final position. This condition persists even if the side-clamp force is increased.

- *Clamp stall.* The workpiece stops moving before reaching the desired final position, because the side clamp is not powerful enough to overcome friction. This condition may be eliminated by increasing the side clamp force.
- *Tipping.* The workpiece tips in an out-of-plane rotation before reaching the desired final position.
- *Escape.* The workpiece slips away from the desired final position. The clamp successfully closes, but the workpiece has "slipped out" of the desired grasp.

The analysis of these conditions is fairly involved, so we will defer the details until Appendix D for clarity. This appendix explains how an exact test for these conditions may be implemented, and also describes a simpler approximate test which we use in our implementation. This test focuses particularly on jamming and clamp stall failures, and has successfully discarded fixtures that exhibited these loading problems in our physical experiments.

5.4 Overall Score

The preceding sections have explained several functions that analyze fixture quality from the perspectives of force resistance, position repeatability, and ease of loading. In this section, we explain the method of combining these functions into the $\mathtt{Q}_{xy}$, $\mathtt{Q}_{z}$, and $\mathtt{Q}_{3d}$ functions required by the fixture generation algorithm.

In order to setup the above quality metrics for a fixturing problem, the user must specify the expected forces $\{\mathbf{F}\}$, force limits $F_{\text{limit}_{\text{locator}}} \cdots$, critical feature specifications $\{\mathbf{P}\}$, and a loading clearance d_{load}. These parameters can generally be determined from the physical task characteristics; Section 7 shows several examples. In addition to this input, the user must specify how to combine the incomparable scores of force resistance, position repeatability, and ease of loading into a single overall score. Our metric provides two methods for this: weighted sum and minimum, as indicated by the input *q-combination-method* flag. The weighted sum method attempts to balance these different considerations according to a user-defined set of weights; the minimum method simply returns the minimum score.

Because of the simplicity of the minimum method, in this section we will focus exclusively on the weighted sum method of combining quality scores. Under this method, the overall fixture quality is

$$q = w_F q_F + w_P q_P + w_L q_L \tag{1}$$

where w_F, w_P, and w_L are user-defined scalars in $[0, 1]$ that satisfy $w_F + w_P + w_L = 1$. If the user strictly requires that fixtures be easy to load, then w_L should be set to zero, because all output fixtures will have $q_L = 1$.

Because some tasks have different requirements in the horizontal and vertical directions, we further allow the user to specify the relative weighting of the xy and z aspects of the quality analysis. Thus

$$q_F = w_{F_{xy}} q_{F_{xy}} + w_{F_z} q_{F_z} \tag{2}$$

and

$$q_P = w_{P_{xy}} q_{P_{xy}} + w_{P_z} q_{P_z} \tag{3}$$

where $w_{F_{xy}}$, w_{F_z}, $w_{P_{xy}}$, w_{P_z} are scalars in $[0, 1]$ which satisfy $w_{F_{xy}} + w_{F_z} = 1$ and $w_{P_{xy}} + w_{P_z} = 1$. The terms $q_{F_{xy}}$, q_{F_z}, and $q_{P_{xy}}$ are the quality scores returned by the $\mathtt{Q}xy_F$, $\mathtt{Q}z_F$, and $\mathtt{Q}xy_P$ functions described above; the q_{P_z} term is a constant representing the value returned by a hypothetical non-planar position analysis function that is not yet implemented. Choosing $q_{P_z} = 0$ will cause q_P to range over $[0, w_{P_{xy}}]$, while choosing $q_{P_z} = 1$ will cause q_P to range over $[w_{P_z}, 1]$. We prefer $q_{P_z} = 1$ because this causes high-quality fixtures to have recognizably good scores, and because our lack of a non-planar position analysis amounts to an implicit assumption of perfect non-planar position repeatability.

In certain situations a fixture design must be analyzed using a full 6-dof force analysis. When this occurs, $\mathtt{Q}3d_F$ clobbers the above q_F score and sets $q_F = q_{3d_F}$. Similarly, some situations may require a future $\mathtt{Q}3d_P$ function to set $q_P = q_{3d_P}$. These situations will be recognizable to $\mathtt{Q}_{3d}$, which will compute the ultimate value of q using the appropriate q_F and q_P values.

This model of quality combination allows the user to focus analysis weight on the most critical aspects of a fixture design task, or spread analysis weight evenly among aspects when there is no particular driving criterion. When used in conjunction with the thresholds that discard inadequate fixtures, this allows substantial flexibility in the filtering and scoring of fixture designs.

Thus in addition to providing the physical parameters described above, the user must also determine their quality scoring preferences and provide input weights w_F, w_P, w_L, $w_{F_{xy}}$, w_{F_z}, $w_{P_{xy}}$, w_{P_z}, and a flag *easy-loading-required?*, indicating whether or not fixture designs should be discarded if they are not easy to load.

Given this input, the user interface constructs the $\mathtt{Q}_{xy}$, $\mathtt{Q}_z$, and $\mathtt{Q}_{3d}$ quality functions and w_{xy}, w_z, and w_{3d} weights required by the fixture generation algorithm. These functions call the constituent quality analysis functions described above, returning $\emptyset$ if any constituent function returns $\emptyset$. Otherwise, these functions return quality scores calculated using the formulas derived below. For the moment we will assume that *easy-loading-required?* is true and $w_L = 0$, allowing us to ignore loading quality terms.

Substituting equations (2) and (3) into equation (1) gives

$$q = w_F(w_{F_{xy}} q_{F_{xy}} + w_{F_z} q_{F_z}) + w_P(w_{P_{xy}} q_{P_{xy}} + w_{P_z} q_{P_z}).$$

Rearranging gives

$$q = w_{F_{xy}} w_F q_{F_{xy}} + w_{P_{xy}} w_P q_{P_{xy}} + w_{F_z} w_F q_{F_z} + w_{P_z} w_P q_{P_z}.$$

It is illuminating to assume that $w_{F_{xy}} = w_{P_{xy}} = w_{xy}$ and $w_{F_z} = w_{P_z} = w_z$.[1] If this assumption is met, then we can write

$$q = w_{xy}(w_F q_{F_{xy}} + w_P q_{P_{xy}}) + w_z(w_F q_{F_z} + w_P q_{P_z}).$$

This allows us to express the overall fixture quality score as a weighted sum of the xy- and z-constraint quality scores:

$$q = w_{xy} q_{xy} + w_z q_z$$

where

$$\begin{aligned} q_{xy} &= w_F q_{F_{xy}} + w_P q_{P_{xy}} \\ q_z &= w_F q_{F_z} + w_P q_{P_z}. \end{aligned}$$

We can now include the case where *easy-loading-required?* is false and $w_L \neq 0$. The loading analysis is a predicate that discards fixtures that are not easy to load. This information is not fully available until the complete 3-d fixture is analyzed, so we reflect easy loading in the 3-d quality score q_{3d}. This gives:

$$q = w_{xy} q_{xy} + w_z q_z + w_{3d} q_{3d} \tag{4}$$

where

$$\begin{aligned} q_{xy} &= w_F q_{F_{xy}} + w_P q_{P_{xy}} \\ q_z &= w_F q_{F_z} + w_P q_{P_z} \\ w_{3d} &= w_L \\ q_{3d} &= q_{3d_L}. \end{aligned}$$

The quality functions $\mathtt{Q}_{xy}$, $\mathtt{Q}_z$, and $\mathtt{Q}_{3d}$ return the values q_{xy}, q_z, and q shown above. This derivation has shown how we may convert the user's input intuitive force, position, and loading weights into quality functions specialized to xy- and z-constraint partial fixture designs. This decomposition supports the pruning methods presented in Sections 4.4 and 6.1.

In some situations the $\mathtt{Q}_{3d}$ function must perform a full 6-dof force or position analysis; when this occurs, the corresponding components of the q_{xy} and q_z quality scores are ignored. For example, if a 6-dof force analysis is required, then

$$q = w_F q_{3d_F} + w_P(w_{xy} q_{P_{xy}} + w_z q_{P_z}) + w_L q_{3d_L}.$$

Unfortunately, use of this quality function essentially precludes the use of pruning techniques to speed the search for the global optimum fixture design.

[1] This assumption is not required, because the algorithm can proceed without it by including additional bookkeeping, even with branch-and-bound pruning.

6 Algorithm Variations

The preceding sections have focused on the case where the *z-constraint-on-grid?* and *top-clamps?* parameters are both true. In this section, we will explain the modifications to the algorithm required when these values are set to false.

6.1 Off-Grid z-Constraints

Some fixture design scenarios allow fixture elements to be placed at arbitrary locations. For example, fixtures produced for mass production applications are often fabricated from plain tooling plate. Our algorithm treats this case identically in the xy-constraint enumeration procedure, placing locators and the side clamp on hypothetical grid locations. However, when *z-constraint-on-grid?* is false, the algorithm employs a modified z-constraint synthesis procedure that takes advantage of the additional design freedom.

When z-constraint elements are restricted to grid locations, then z-constraint synthesis must be repeated for each xy-constraint, because the xy-constraint's (x, y, θ) pose determines the relationship between the workpiece and the possible support/top-clamp locations. If z-constraint elements are allowed off the grid, then high-quality z-constraint designs can be found for the workpiece pose $(0,0,0)$, and then transformed onto each xy-constraint design.

When a z-constraint is transformed into the xy-constraint coordinate frame, the algorithm checks for interference between the z-constraint elements and the xy-constraint elements. If there is no interference between elements, the algorithm proceeds to synthesize a 3-d fixture design from the xy-constraint/z-constraint pair and proceeds as above. If the xy-constraint interferes with the z-constraint supports, then the pair is discarded. If the xy-constraint interferes with the z-constraint clamp bodies or arms, then the algorithm performs body placement analysis for additional candidate body placement locations cached in the appropriate z-contact data structures. If an interference-free body placement is found, then the algorithm proceeds to synthesize a 3-d fixture design; otherwise, the pair is discarded.

This modification improves the run time of the algorithm, because it allows significant computation to be moved outside the innermost loop. When *z-constraint-on-grid?* is true, the algorithm must repeat z-constraint construction for each xy-constraint, requiring $N_{xy}N_z$ total z-constraint construction operations. If *z-constraint-on-grid?* is false, then z-constraint construction only needs to be performed once, requiring only N_z construction operations. The total complexity is still $N_{xy}N_z$, because this is the number of possible xy-constraint/z-constraint merging operations.

We obtain further improvement by increasing the pruning capabilities of the algorithm. The algorithm begins by constructing all possible z-constraints during pre-processing, and scores and sorts the resulting z-constraints in order of decreasing quality q_z. These q_z values will remain valid even if the clamp bodies are relocated, because we require $\mathtt{Q}_z$ to be insensitive to clamp body placement.

Once the z-constraints have been constructed, the algorithm proceeds to generate xy-constraints. For each xy-constraint, the algorithm visits each z-constraint in turn, checking the pruning condition:

$$w_{xy}q_{xy} + w_z q_z + w_{3d}\cdot 1 \leq p q_{\text{best}}. \tag{5}$$

When this condition is satisfied, we know that this z-constraint will not produce the global optimum design, and neither will any subsequent z-constraint in the z-constraint list. In this case the algorithm ignores this and all remaining z-constraints, and proceeds to the next xy-constraint. If the condition is not satisfied, then the algorithm attempts to construct a 3-d fixture from the xy-constraint/z-constraint pair, and then proceeds to the next z-constraint on the list.

If $w_{3d} = 0$ and $p = 1$, then expansion of the z-constraint list for a given xy-constraint will always halt as soon as a z-constraint generates a valid 3-d fixture design. Thus each xy-constraint will produce exactly one 3-d fixture design, usually after considering only a very small number of z-constraints. Under these conditions, pruning effectively reduces the total algorithmic complexity to $N_{xy} + N_z$. This corresponds to the important class of fixture design problems where off-grid z-constraint elements are allowed and easy loading is strictly required (see Section 7.1).

There are a few subtleties to the off-grid synthesis method. First, when generating candidate z-contact support locations, the algorithm generates grid points under the bottom surfaces of $\mathcal{W}$ in the pose $(0,0,0)$. This allows us to use the actual surface bounding boxes, rather than bounding boxes developed from bounding circles as described in Section 4.2.3. The algorithm also includes additional candidate points generated by finding the vertices of each bottom surface after shrinking by $r_{\text{support-pad}} + d_{xy}$. These extremal candidate support points may produce higher-quality z-constraints than are possible using only grid points.

A second subtlety is that when z-constraint elements are allowed off the grid, they may be placed arbitrarily close to each other or to elements of the xy-constraint. When the fixture is fabricated by inserting these elements into holes drilled in plain tooling plate, a minimum separation distance may be required due to material strength considerations. The fixture generation algorithm accommodates this by including an additional input parameter $d_{\text{separation}}$, which describes the minimum required separation distance between a pair of fixture elements. These separation distances are then included in the interference check (a) on page 25 of Section 4.2.3 by replacing $r_A + r_B$ with $r_A + r_B + d_{\text{separation}}$. A similar check is also applied to avoid interference with the xy-constraint elements. Different required separations between various types of elements may be supported by adding appropriate bookkeeping.

The third subtlety is that when z-constraint elements may be placed off the grid, the set of candidate body placement locations may be treated differently. When elements are restricted to the grid, a discrete set of candidate locations is determined by the available arm lengths and the grid spacing. When elements are allowed off the grid, then the top clamp body may be placed anywhere along a circle of radius l_{arm} centered at the support/clamp point (x_z, y_z). In this case the algorithm generates a discrete set of candidate placement locations (x_{z2}, y_{z2}) on the circle, sampling at regular θ-intervals (we use 30° in our implementation). The resulting set of placement positions is then sorted according to distance outside the workpiece silhouette, favoring (x_{z2}, y_{z2}) points that are furthest from the silhouette. This heuristic favors body placement positions that maximally clear the workpiece, thereby increasing the likelihood that the first candidate body position will be feasible, and decreasing the likelihood of interference during loading.

6.2 Fixtures Without Top Clamps

Some fixture design scenarios require fixtures without top clamps. This is often the case for fixtures designed for automated assembly operations, where applied forces are either downward or lateral and there is a desire to minimize fixture cost. When the *top-clamps?* flag is false, the xy-constraint synthesis method remains unchanged, but z-constraint synthesis is modified in both the z-constraint generation procedure and in the z-constraint quality metric $\mathtt{Q}_{z_F}$.

Let's focus first on the z-constraint generation procedure. When top clamps are absent, the identification of candidate support locations remains unchanged, for both possible values of *z-constraint-on-grid?*. Of course, all steps related to top clamp placement are eliminated. Thus the generation of candidate z-contacts remains unchanged, except that some steps are eliminated.

The key difference in z-constraint synthesis in the two cases is the way that the list of z-contacts is converted to a list of z-constraints. In the case with top clamps, each non-collinear triple is a candidate z-constraint, so a list of n z-contacts may produce roughly n^3 different z-constraints. Each resulting z-constraint is then comprised of three bilateral constraints that kinematically prevent out-of-plane motion for all possible applied forces.

In the case without top clamps, each z-contact provides only unilateral constraint. Now a triangle of z-contacts only prevents out-of-plane motion for downward forces exerted in the interior of the triangle; downward forces exerted outside the triangle will cause the workpiece to tip. Thus to maximize the z-constraint's ability to prevent out-of-plane motion, we form the convex hull of all possible support points, and place supports at the vertices of this convex hull. Thus a list of n z-contacts will produce exactly one z-constraint when top clamps are absent.

There are a few subtleties required to make this computation work well in practice. First, supports corresponding to convex-hull vertices may interfere with elements of the xy-constraint, either by intersecting an xy-constraint element or violating the $d_{\text{separation}}$ requirement. To avoid this problem, the list of candidate support locations is filtered to discard illegal support placement before the support convex hull is formed. These operations should not be reversed, because discarding hull vertices after the convex hull operation may lead to a greatly reduced support region. This implies that if *z-constraint-on-grid?* is false, the convex hull operation must be delayed until after the off-grid z-constraint elements are transformed to the xy-constraint coordinate frame. Whether on-grid or off-grid, the support convex-hull operation should only be performed on candidate support points that adequately clear the xy-constraint elements.

The convex hull that results from this construction may have many vertices, several of which provide little constraint value. To allow economic considerations to prevent the placement of unnecessary support points, the algorithm accepts a user-specified parameter d_{simplify}, which controls the deletion of nearly-degenerate convex hull vertices. This parameter provides a bound on the distance d_{remove}, which is used to choose convex-hull vertices to eliminate. The meaning of d_{remove} is illustrated in Figure 26; this is the height of

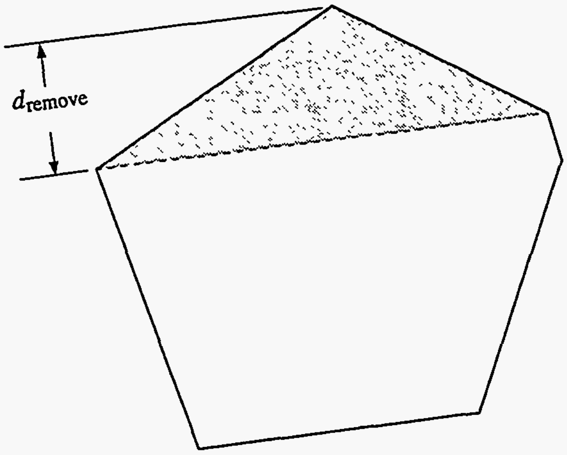


Figure 26: The distance d_{remove}.

the triangle that would be excluded from the convex-hull if the candidate vertex would be removed. The algorithm simplifies the support convex hull by identifying the vertex with the smallest value of d_{remove}, and removing it if $d_{\text{remove}} \leq d_{\text{simplify}}$. This process is repeated until $d_{\text{remove}} > d_{\text{simplify}}$ for all vertices. This allows the number of convex-hull vertices to be reduced in a way that minimizes the reduction in the support region, while allowing the user to control the aggressiveness of the simplification.

After forming the convex hull and removing all nearly-degenerate vertices, it is still possible that the convex hull will itself contain vertices that violate the $d_{\text{separation}}$ requirement. When this occurs, the algorithm finds the closest pair of vertices, and attempts to replace the pair with a single vertex midway between the two. This may not be feasible due to interference considerations; if this is the case, then the algorithm discards the vertex with the smallest d_{remove} value. Either way, the algorithm assures that the pair of too-close vertices is replaced with a single vertex. This process is repeated until all convex hull vertices obey the $d_{\text{separation}}$ requirement.

This completes our description of the z-constraint generation without top clamps. Candidate support locations are generated as before, and then filtered to remove support placements that interfere with the xy-constraint. The convex hull of the remaining support points is then formed, and simplified according to d_{simplify}. The resulting convex hull is then further simplified to eliminate $d_{\text{separation}}$ violations. This produces a z-constraint support region that is nearly maximal, economical, and obeys all interference restrictions.

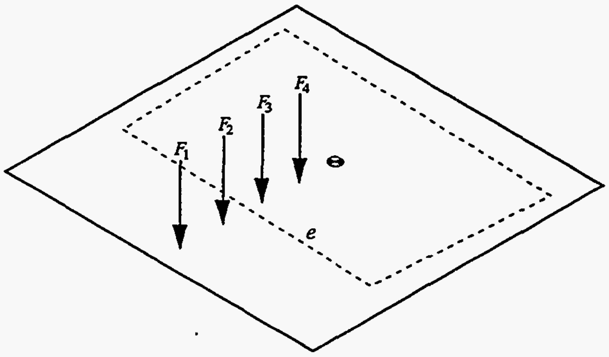


Figure 27: Example applied forces. The support region is indicated by a dashed line.

We now turn our attention to the quality metric $\mathtt{Q}_{z_F}$. In the case with top clamps, $\mathtt{Q}_{z_F}$ calculates the contact reaction force at each of the supports and top clamps arising from the expected applied forces, and compares this with user-specified maximum force limits. This makes sense in scenarios where the user wishes to avoid workpiece deformation resulting from large applied forces.

In the case without top clamps, there are two key differences: First, it is impossible to calculate the contact reaction forces, because a z-constraint containing more than three supports is statically indeterminate. Second, in applications such as assembly the forces are typically small, so workpiece deformation is not a primary concern. Thus the quality metric $\mathtt{Q}_{z_F}$ for the case without top clamps considers the fixture's ability to prevent the workpiece from tipping under the influence of expected applied forces.

We can convert this intuitive notion into a well-behaved metric that may be evaluated for arbitrary applied forces, not just downward forces. To understand this metric, consider the situation shown in Figure 27. This figure shows an example support region and a series of example applied forces. These example forces are all downward; the case of general applied forces is addressed in Appendix B. Also shown is the workpiece center of mass. If $(x_{\text{com}}, y_{\text{com}})$ lies outside the support region, then the z-constraint is rejected because the workpiece will tip between operations. Forces F_1 and F_2 are applied outside the support region; these forces tend to cause the object to tip. Force F_1 lies further outside the support region, so its tipping moment is larger and it is somehow "worse" than F_2. However, neither force will cause the object to tip if the object weighs enough to overcome the tipping moment. Thus the magnitude of the tipping moments exerted by F_1 or F_2 must be compared to the anti-tipping moment exerted by the workpiece center of mass in order to determine whether tipping will occur.

One convenient way to quantify this effect is to calculate the additional workpiece weight w_{excess} that is required to just prevent tipping. For example, suppose the workpiece weight is 1lb and F_1 causes the workpiece to tip, but the workpiece would not tip for all workpiece weights $\geq$ 1.3lb. In this case w_{excess} would be 0.3lb. Inspection of Figure 27 reveals that F_1 would yield the largest value of w_{excess}, F_2 would yield a smaller value, F_3 would yield a small negative value, and F_4 would yield a more negative value than F_3. For the forces F_3 and F_4 we measure w_{excess} relative to the support edge e. Appendix B explains how to calculate w_{excess} for general applied forces.

This parameter provides the basis for our tipping quality metric $\mathtt{Q}_{z_F}$. Values of w_{excess} range over $[-\infty, +\infty]$, with quality decreasing as w_{excess} increases. To convert this into a well-defined quality metric over $[0, 1]$, we need to identify upper and lower bounds delineating meaningful values of w_{excess}.

The upper bound is determined by the user input parameter w_{limit}. This parameter allows the user to vary the degree of tipping robustness required by the algorithm. If $w_{\text{limit}} = 0$, then fixtures will be accepted if they marginally prevent tipping. Selecting $w_{\text{limit}} < 0$ adds a safety factor; this is generally unnecessary because marginal fixtures will be given lower quality scores than fixtures that robustly prevent tipping. Selecting $w_{\text{limit}} > 0$ allows some fixtures to be returned that are vulnerable to tipping; this may be desirable in cases where preventing all tipping is impossible, but the user would like to study the output fixture designs anyway.

The lower bound w_{ideal} is determined by identifying a threshold beyond which additional robustness against tipping is irrelevant. This is a subjective parameter; our implementation uses a default value of $w_{\text{ideal}} = -2w_{\mathcal{W}}$, where $w_{\mathcal{W}}$ is the workpiece weight.

Our tipping quality score is then determined by finding w_{max}, the maximum value of w_{excess} for the set of applied forces. If $w_{\text{max}} > w_{\text{limit}}$, then $\mathtt{Q}_{z_F}$ returns $\emptyset$. If $w_{\text{max}} < w_{\text{ideal}}$, then $q_{F_z} = 1$. Otherwise,

$$q_{F_z} = 1 - \left(\frac{w_{\text{max}} - w_{\text{ideal}}}{w_{\text{limit}} - w_{\text{ideal}}} \right).$$

There is one additional subtlety required to develop this metric for practical problems. In some situations the weight and center of mass of the workpiece changes during task execution; for example, in assembly applications the effective workpiece mass properties change significantly as various parts are added to the subassembly. Because these properties are crucial to the proper calculation of q_{z_F}, we extend the input set of expected applied forces to include the current workpiece center of mass and weight. Thus each $\mathbf{F}_i$ becomes

$$\mathbf{F}_i = \langle [F_x \; F_y \; F_z \; \tau_x \; \tau_y \; \tau_z]_i \;\; (x_{\text{com}}, y_{\text{com}}, z_{\text{com}})_i \;\; w_i \rangle$$

where $(x_{\text{com}}, y_{\text{com}}, z_{\text{com}})_i$ and w_i are the workpiece center of mass and weight that are current when $[F_x \; F_y \; F_z \; \tau_x \; \tau_y \; \tau_z]_i$ is applied.

This completes our explanation of the case where *top-clamps?* is false. One closing remark is that this synthesis procedure is even more efficient than the pruned off-grid synthesis procedure with top clamps. Because there is exactly one z-constraint, the overall complexity is simply N_{xy}.

7 Case Studies

The previous sections have explained how the fixture generation algorithm works, and how the style of fixtures generated by the algorithm may be varied by changing the values of the flags *z-constraint-on-grid?* and *top-clamps?*. In this section, we explore how the algorithm may be applied to solve practical problems. We will revisit the machining and assembly examples presented in Section 1, explaining each in detail.

In each case we also report the performance of our implementation on the problem. This program is written in Common Lisp, with a user interface written in CLIM. Our program currently employs a geometric representation of volumes based on prisms of constant cross-section; we are currently modifying the code to use ordinary CAD models instead, expressed in the widely supported STL format. The performance results reported below reflect run times on an SGI Indigo II workstation.

7.1 Final Machining of Complex Workpieces

Near-net-shape fabrication methods are techniques for efficiently producing parts with complex shapes. Examples include casting and welding, which can efficiently produce parts that would be very costly to machine from raw stock. Due to the lack of precision of these processes, final machining operations are often required to create precise part features such as gasket surfaces, threaded holes, and the like. These machining operations require fixtures that can hold the complicated workpiece while avoiding interference with cutting paths.

Figure 28 shows the finish machining operations required to produce the cast housing shown in Figure 1. These include drilling several holes and milling two gasket surfaces. At least two fixtures are required to perform these operations — one holding the workpiece upright and one holding the workpiece inverted. In the following paragraphs, we will examine the problem of designing the fixture for the workpiece in the upright orientation.

Prototype Fabrication

In the prototype phase of product development, a small number of copies of the part are required for fit checking, performance evaluation, and destructive testing. These prototype parts are often fabricated in a job shop where manual casting and machining methods are employed. Because of the small production quantities, fixture design and fabrication costs can comprise a significant portion of the total cost to produce the prototype parts. Further, storing the fixture for future use can lead to significant fixture storage costs.

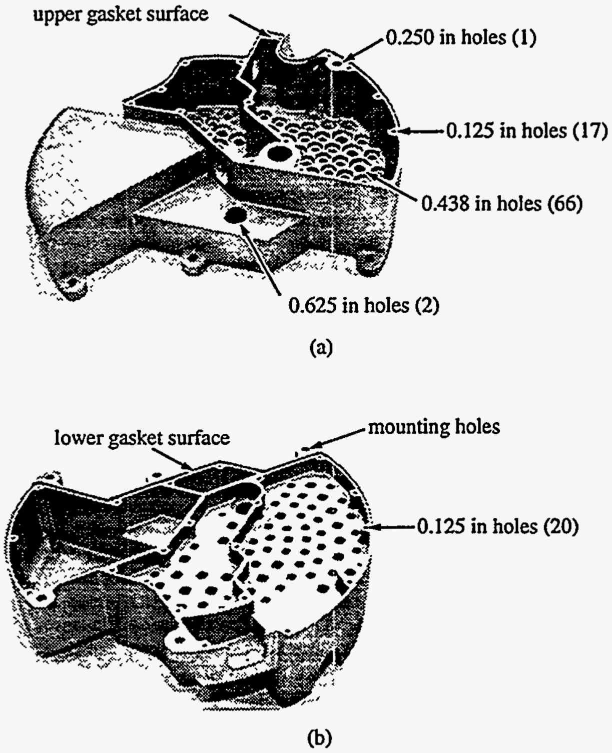


Figure 28: Finish machining operations required for the housing. (a) Operations performed with the housing in the upright orientation. (b) Operations for the inverted orientation.

The costs of fixture fabrication and storage may be reduced by constructing the fixture from re-usable modular elements. Modular fixture systems were first introduced in World War II, and are available from a number of commercial firms worldwide. These systems generally fall into two categories: hole-based and slot-based. The hole-based systems have a base plate with an array of precisely machined holes, and elements that can be attached to the plate at precise locations by inserting dowel pins into the holes. The slot-based systems have a arrangement of slots with a T-shaped cross section, allowing elements to be attached at any position along the slot. Both systems reduce fixture storage costs by allowing re-use of the basic fixture elements.

In order to minimize the time required to fabricate the fixture, we have chosen to focus on hole-based fixture systems. In these systems, a precise fixture is obtained by simply assembling the elements, which is much faster than the careful measurement and adjust-

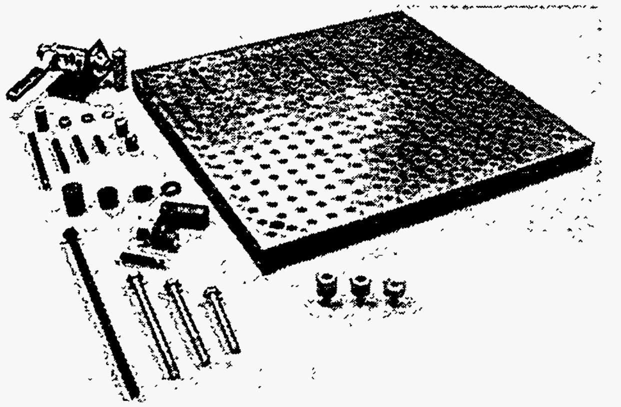


Figure 29: A typical modular fixture kit. Some bolt lengths and spacer thicknesses are omitted to avoid clutter.

ment of fixture element positions required by slot-based systems.

Figure 29 shows an example hole-based modular fixture kit. This fixture kit contains the base plate, side locators, support pads, and side and top clamps required by our algorithm. This particular kit is comprised of commercial elements available from Qu-Co [61]. The base plate has an alternating grid of smooth and threaded holes, spaced 0.75in apart; the lower half of each smooth hole is threaded, so that clamps may be attached to any hole, while side locators may only be attached to alternating holes. The side locators, support pads, and clamps all have associated spacers that allow them to be set at various heights.

We made a few slight modifications to the commercial fixture elements to improve the generality of the overall fixture kit:

- First, we fabricated a cylindrical tip for the side clamp plunger that matched the side locator diameter; this tip allowed us to treat the side locators and side clamp tip as geometrically similar fixels. We then modified the clamp bracket to include specific mounting holes instead of mounting slots. This allowed the clamp to be placed in the proper position on the plate by simply attaching it to the proper holes in the plate; otherwise, careful measurement would be required to assure that the clamp placement location matched the algorithm's expectations. Because this clamp uses an over-center mechanism that locks the plunger into a specific closed position, the travel range $[x_{\text{travel}_{\min}}, x_{\text{travel}_{\max}}]$ of the resulting side clamp is determined by the bolt and locknut that attach the clamp tip to the plunger.
- Second, we requested support pads with a female threaded hole in the bottom, rather than the standard male threaded stud. We then included a set of studs in the kit that are threaded at both ends, covering a wide range of lengths in 0.25in increments. We also included a series of spacers that fit over the studs but had a smaller outside diameter than the support pad. The spacer thickness varies over a binary progression of thicknesses from 1.0in to 0.001in. This arrangement allows the user to quickly and accurately establish any desired support height to within 0.001in, while requiring only a few hardware elements. To produce a cylindrical support of a particular height, the user simply selects a stud of the proper length, screws it into the base plate hole, drops the appropriate sequence of spacers over the stud, and then tightens the support pad onto the top of the stud.
- Finally, we modified the top clamps to employ longer arms. This modification significantly increased the number of solutions which allowed the top clamp to be placed without interference. We also trimmed away unnecessary material protruding behind the arm, which sometimes obstructed vertical loading with the arm in the retracted position.

These modifications produced a fixture kit which allows the user to quickly assemble a fixture that exactly matches the algorithm's expectations. We also studied kits which included additional elements. For example, we built special brackets which allow placement of supports and top clamps at arbitrary positions; these elements allow the assembly of fixtures generated with *z-constraint-on-grid?* set to false. Fixtures employing these elements take much more time to assemble than fixtures that place all supports and top clamps on the grid, so we prefer the simple fixture kit of Figure 29, as long as good-quality solutions can be found with *z-constraint-on-grid?* set to true.

Now suppose that a user has this fixture kit available, and would like to design a fixture to solve the problem shown in Figure 28(a). To do so, the user must provide a CAD model of the workpiece and specify the task geometric access constraints, expected forces, force limits and position tolerances, and set several control flags. The paragraphs that follow will explain how each of these values may be determined.

To specify the task geometric access constraints, the user delineates the volume swept by the cutter during machining operations. This volume is shown in

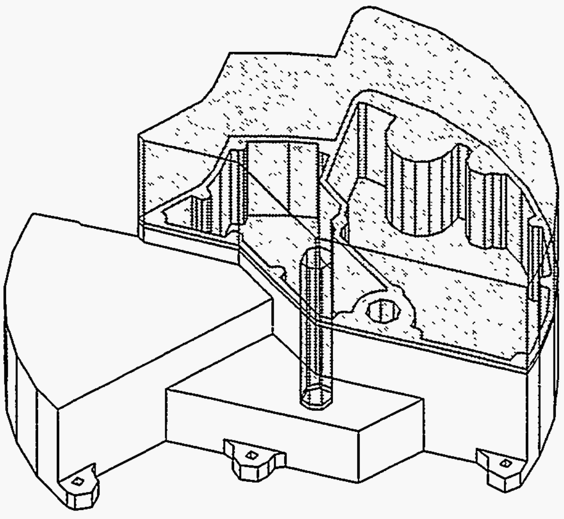


Figure 30: The volume swept by the cutter during machining.

Figure 30. This volume becomes the algorithm input $\mathcal{C}_{\text{only-loaded}}$. No $\mathcal{C}_{\text{only-loading}}$ or $\mathcal{C}_{\text{always}}$ volumes are required in this case.

We envision that in the future this volume would be automatically generated by a system that integrates process planning and fixture design. In the meantime, this volume may be specified using a CAD editor.

The expected forces could also be automatically generated by an integrated process planning and fixture design system. Because this system remains for future work, the following paragraphs will explain how we calculated these forces for this example. See [15, 29] for a more detailed analysis of cutting forces.

The forces expected during cutting may be obtained from the intended machining operations. The two principle machining operations required to finish the housing are milling the gasket surface and drilling several holes.

Let's consider the milling operation first. The gasket surface is formed by side-cutting a 0.125in × 0.125in interior ledge along the upper wall of the housing. We can predict the lateral machining force by applying the basic power equation $P = Fv$, rearranged to

$$F_{\text{mill}} = \frac{P_c}{v_c}$$

where P_c is the cutting power, and v_c is the cutting speed. This equation matches the cutting model given in [70] and used in [35], after appropriate unit conversions.

The cutting power may be obtained from the Machinery's Handbook [54]:

$$P_c = K_p C Q W$$

where K_p is a power constant, C is a factor compensating for feed rate, Q is the metal removal rate in cubic inches per minute, and W is a tool wear factor.

To calculate the cutting force for our example gasket cutting operation, we proceed as follows. The first step is to look up the proper cutting speed and feed rate in the Machinery's Handbook. These parameters have been empirically determined and tabulated for various materials, and are used by most practicing machinists. The housing material is cast aluminum B443.0; to finish cut this material, the tables specify a cutting speed of $600\frac{\text{ft}}{\text{min}}$ and a feed rate of $0.008\frac{\text{in}}{\text{tooth}}$.

This information can be used to calculate the cutter rpm ω, the machine feed rate R, and material removal rate Q. For a 0.5in diameter cutter with two teeth, we obtain

$$\begin{aligned} \omega &= \frac{600\frac{\text{ft}}{\text{min}} \cdot 12\frac{\text{in}}{\text{ft}}}{\pi \cdot 0.5\text{in}} \\ &\approx 4580\frac{1}{\text{min}} \end{aligned}$$

$$\begin{aligned} R &= 0.008\frac{\text{in}}{\text{tooth}} \cdot 2\text{tooth} \cdot 4580\frac{1}{\text{min}} \\ &\approx 73.3\frac{\text{in}}{\text{min}} \end{aligned}$$

$$\begin{aligned} Q &= 73.3\frac{\text{in}}{\text{min}} \cdot 0.125\text{in} \cdot 0.125\text{in} \\ &\approx 1.15\frac{\text{in}^3}{\text{min}} \end{aligned}$$

Next we compute the cutting power P_c. This requires the empirical parameters K_p, C, and W, also obtained from the Machinery's Handbook. For performing a finishing side-cut of aluminum B433.0, the tables yield $K_p = 0.25\frac{\text{hp}\cdot\text{min}}{\text{in}^3}$ and $C = 1.08$. For a sharp cutting tool, $W = 1.0$. This gives

$$\begin{aligned} P_c &= 0.25\frac{\text{hp}\cdot\text{min}}{\text{in}^3} \cdot 1.08 \cdot 1.15\frac{\text{in}^3}{\text{min}} \cdot 1.0 \\ &\approx 0.31\text{hp} \end{aligned}$$

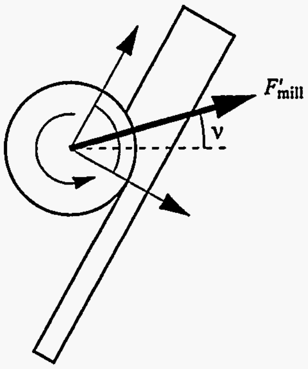


Figure 31: Forces exerted during milling.

Finally we obtain the lateral cutting force

$$\begin{aligned} F_{\text{mill}} &= \frac{0.31\text{hp}}{600\frac{\text{ft}}{\text{min}}} \cdot 33{,}000\frac{\frac{\text{ft}\cdot\text{lb}}{\text{min}}}{\text{hp}} \\ &\approx 17\text{lb} \end{aligned}$$

Thus the tool exerts a lateral force of 17lb on the workpiece during cutting. The exact direction of this force is difficult to ascertain, and probably varies with time as each cutter tooth engages the material. Thus we conservatively assume that the force direction may lie anywhere within the 90° arc shown in Figure 31. To simplify matters, we model this range of forces with a single force pointing midway through the arc, of length $F'_{\text{mill}} = 17\text{lb}\sqrt{2} \approx 24\text{lb}$. This force model is conservative in the sense that if the fixture can adequately resist this force, it can also resist any force of 17lb chosen within the 90° arc.

This force magnitude F'_{mill} and direction ν can then be converted to the required $\mathbf{F}_i = [F_x\ F_y\ F_z\ \tau_x\ \tau_y\ \tau_z]$ force vector by taking into account the cutting point (x, y, z):

$$\begin{aligned} F_x &= F'_{\text{mill}} \cos(\nu) \\ F_y &= F'_{\text{mill}} \sin(\nu) \\ F_z &= 0 \\ \tau_x &= -F_y z \\ \tau_y &= F_x z \\ \tau_z &= F_y x - F_x y \end{aligned}$$

We implemented code to calculate these $\mathbf{F}_i$ for a sampling of points along the gasket contour, assuming a counter-clockwise cutting path. Two forces were constructed at each corner of the gasket contour, to account for the tool approach and departure motion directions. This produced a set of 67 distinct applied forces, shown in Figure 32.

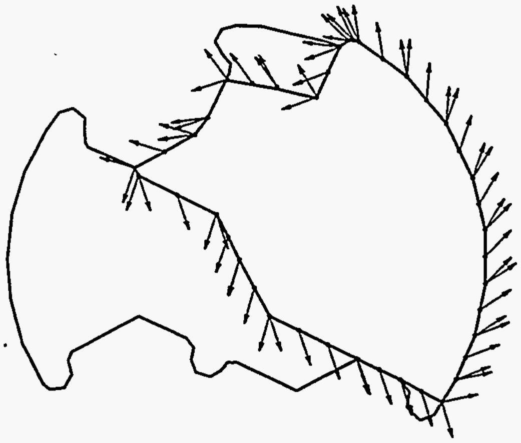


Figure 32: Our model of the forces that arise while milling the gasket surface. Each arrow represents a line of force parallel to the xy-plane, at the gasket z height. The magnitude of each force is 24lb.

We may now turn our attention to the drilling operations. The Machinery's Handbook provides a formula for predicting the thrust force and torque exerted by a drilling operation:

$$\begin{aligned} F_{\text{drill}} &= K_d W (2F_f F_t B + d^2 J) \\ \tau_{\text{drill}} &= K_d W F_f F_m A \end{aligned}$$

where d is the drill diameter, K_d is a factor that depends on the workpiece material, W is the tool wear factor, F_f is a factor that depends on the drill feed rate, and F_t, F_m, A, B, and J are factors that depend on the drill geometry.

Because drill feed rate depends on material and hole diameter, this model implies that if standard drill bits are used, the force and moment exerted during drilling depend only on the workpiece material and drill diameter. For example, for a 0.625in hole drilled in aluminum B433.0 using a sharp standard-geometry bit, the tables specify

$$\begin{aligned} K_d &= 7000.0 \\ F_f &= 0.035 \\ F_t &= 0.687 \\ F_m &= 0.429 \end{aligned}$$

$$
\begin{aligned}
A &= 1.085 \\
B &= 1.355 \\
J &= 0.030 \\
W &= 1.0
\end{aligned}
$$

The Machinery's Handbook does not provide units for these constants, but indicates that d should be given in inches, and the resulting F_{drill} and τ_{drill} values will have units of lb and in·lb, respectively. Thus

$$
\begin{aligned}
F_{\text{drill}} &\approx 540\text{lb} \\
\tau_{\text{drill}} &\approx 115\text{in}\cdot\text{lb}
\end{aligned}
$$

where τ_{drill} describes a clockwise torque. A similar calculation can be performed for all of the holes required by the housing. The results are summarized below:

d (in)	F_{drill} (lb)	τ_{drill} (in·lb)
0.625	540	115
0.500	434	76
0.438	285	43
0.250	170	16
0.125	64	3

To determine the total applied force $\mathbf{F}_i$ produced when drilling a hole located at position (x, y), we form

$$
\begin{aligned}
F_x &= 0 \\
F_y &= 0 \\
F_z &= -F_{\text{drill}} \\
\tau_x &= F_z y \\
\tau_y &= -F_z x \\
\tau_z &= -\tau_{\text{drill}}
\end{aligned}
$$

In addition to the thrust force exerted while drilling the hole, there is an opposite force exerted when the drill breaks through the back side of the material that tends to pull the workpiece out of the fixture. This force is not addressed in the Machinery's Handbook, so we will assume that this upward force is no greater than 20% of the downward thrust force. Thus each through-hole has an additional $\mathbf{F}_i$ determined by

$$
\begin{aligned}
F_x &= 0 \\
F_y &= 0 \\
F_z &= 0.2\cdot F_{\text{drill}}
\end{aligned}
$$

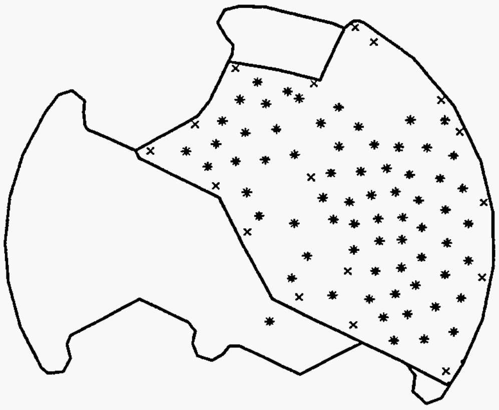


Figure 33: Our model of drilling forces. Each "×" indicates a location where a downward force is exerted, while each "+" indicates an upward force. "$\divideontimes$" indicates both. Each force also has an associated torque about z.

$$
\begin{aligned}
\tau_x &= F_z y \\
\tau_y &= -F_z x \\
\tau_z &= -\tau_{\text{drill}}
\end{aligned}
$$

We plan to drill 68 through holes and 18 blocked holes. Thus the expected machining forces may be modeled with $68\cdot 2 + 18 = 154$ separate applied forces. These forces are illustrated in Figure 33. One subtlety is that the large 0.625in holes are formed by expanding 0.5in holes already in the casting. We modeled the force during this operation as the difference between the forces required to drill a 0.625in hole and a 0.5in hole: $F_{\text{drill}} = 540\text{lb} - 434\text{lb} = 106\text{lb}$ and $\tau_{\text{drill}} = 115\text{in}\cdot\text{lb} - 76\text{in}\cdot\text{lb} = 39\text{in}\cdot\text{lb}$. The next largest hole is 0.438in, so the maximum drilling force exerted on the housing is 285lb and 43in·lb.

This completes our calculation of the applied forces expected during this machining task. To sum up, we constructed a total of 221 $\mathbf{F}_i$ vectors for this problem, corresponding to lateral milling forces of 24lb and drilling forces of up to −285lb and +57lb vertically and torques of up to −43in·lb about z.

In our implementation, these forces are accompanied by the workpiece center of mass and weight (2.821b), in the manner explained at the end of Section 6.2. The quality metric automatically combines the applied machining force with the gravitational force to determine the total force exerted on $\mathcal{W}$. This allows the user to specify the changes in the workpiece mass properties that occur as material is removed during machining; in this example, we left these values constant.

After specifying the expected applied forces, the user must also specify the maximum allowable contact force for each type of fixture element. Because the side clamp and top clamps are tightened and then locked into place, these clamps should be modeled as kinematic constraints, leaving the optional clamp force parameters F_{plunger} and F_{top} blank. Thus the user must choose values for $F_{\text{limit}_{\text{locator}}}$, $F_{\text{limit}_{\text{support}}}$, $F_{\text{limit}_{\text{side}}}$, and $F_{\text{limit}_{\text{top}}}$.

How should these values be chosen? One rational approach is to set each value to the minimum force that will cause permanent deformation of either the fixture or workpiece. Fixtures that induce contact forces exceeding these limits should be viewed as unacceptable.

Let's first consider the force limits due to the workpiece. Permanent workpiece deformation will occur when the material yield strength is exceeded. This can occur either globally when the workpiece bends, or locally when a single contact causes a dimple on the workpiece surface. Global workpiece deformation is clearly important, and appears to require a finite element analysis. We leave this problem for future work, and focus only on local deformation effects here.

Dimpling of the workpiece surface will occur whenever the contact force produces a contact pressure that exceeds the material yield strength. For aluminum B443.0, the Machinery's Handbook lists a yield strength of 6000 psi. The self-aligning support pads and the top clamp contact pad both have a contact radius of 0.125in, and a resulting area of 0.05in^2. Thus the maximum allowable support and top clamp force is $6000\frac{\text{lb}}{\text{in}^2}\cdot 0.05\text{in}^2 = 300\text{lb}$. The load capacities of the support pad and top clamp fixture elements are much larger than this, so $F_{\text{limit}_{\text{support}}}$ and $F_{\text{limit}_{\text{top}}}$ are set to 300lb. This is the same method used by Kim to limit contact force [35].

Determining a force limit for the side locators is more subtle. Because the locators ideally make a line contact with the workpiece, there is zero contact area and therefore infinite pressure. Consequently, any positive contact force will exceed the material yield strength. We resolve this problem by bounding the permanent deformation that is allowable. Englert used a similar method to analyze the effects of locator contacts [21].

In this example, we chose $\delta_{\text{max}} = 0.0001\text{in}$ as the maximum allowable depth of a dimple formed by a side locator or side clamp contact; this value is more than an order of magnitude less than the workpiece shape uncertainty ϵ_{shape}. We can use the material Brinell hardness to determine a maximum contact force that assures that this limit is not exceeded.

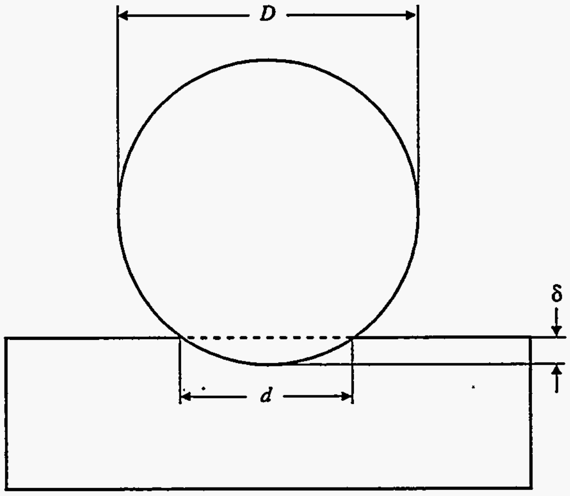


Figure 34: The relationship between penetration depth δ and indentation width d for a circular contact pressing on a flat surface.

The Brinell hardness of a material is measured by pressing a hardened steel ball into the material with a known applied force, and then measuring the diameter of the resulting dimple. If the applied force is F, the ball has diameter D, and the dimple has diameter d, the Brinell hardness H is defined to be [54]

$$H \equiv \frac{F}{\frac{\pi D}{2}(D - \sqrt{D^2 - d^2})}$$

As shown in Figure 34, $\delta = \frac{1}{2}(D - \sqrt{D^2 - d^2})$. Thus the above may be rewritten as

$$H = \frac{F}{\pi D \delta}$$

Because the circle bounding the dimple has an area $A_{\text{dimple}} = \pi D \delta$, this is simply the ratio of the applied force and the dimple area.

Suppose the fixels were spheres rather than cylinders. The maximum allowable force would then be

$$F_{\text{sphere}} = H \pi D \delta_{\text{max}}$$

The tables in the Machinery's Handbook list the minimum Brinell hardness of Aluminum B433.0 as $25\frac{\text{kgF}}{\text{mm}^2} = 35,500\frac{\text{lb}}{\text{in}^2}$. This gives

$$\begin{aligned} F_{\text{sphere}} &= 35,500\tfrac{\text{lb}}{\text{in}^2}\cdot \pi \cdot 0.6875\text{in} \cdot 0.0001\text{in} \\ &\approx 7.7\text{lb} \end{aligned}$$

Thus a spherical fixel must apply less than 7.7lb contact force to avoid exceeding the maximum deformation limit. Because our lateral milling forces are 17lb, this immediately shows that spherical fixels are inadequate for this problem.

For a cylindrical fixel, the dimple is rectangular rather than circular. For a given dimple depth δ, the width of the rectangular dimple is the same as the circular dimple diameter d. The height of the rectangular dimple is the height of the fixel contact overlap; the fixture design algorithm assures that this is always at least h_{required}. Thus the dimple area for a cylindrical fixel contact is always at least dh_{required}.

If we choose h_{required} to equal the full fixel contact height of 0.25in for our fixture kit, then

$$\begin{aligned} d &= \sqrt{D^2 - (D - 2\delta_{\text{max}})^2} \\ &= \sqrt{(0.6875\text{in})^2 - (0.6875\text{in} - 2 \cdot 0.0001\text{in})^2} \\ &\approx 0.017\text{in} \end{aligned}$$

and

$$\begin{aligned} F_{\text{cylinder}} &= H d h_{\text{required}} \\ F_{\text{cylinder}} &= 35,500 \tfrac{\text{lb}}{\text{in}^2} \cdot 0.017\text{in} \cdot 0.25\text{in} \\ &\approx 150\text{lb} \end{aligned}$$

Note that here we're using the formula $F = HA_{\text{dimple}}$, which is not exact for non-spherical contacts. However, this formula is conservative for the case of a cylinder. Because the cylinder has an infinite radius of curvature in one direction, it will produce a smaller penetration depth δ for a given applied force. Thus if we set $F_{\text{limit}_{\text{locator}}} = F_{\text{limit}_{\text{side}}} = 150\text{lb}$, we will be assured that no fixel will produce a dimple deeper than δ_{max}.

We have now shown how to determine values for the task geometric access constraints, expected forces, and force limits. As mentioned above, we envision a future system that would perform these calculations automatically. To finish specifying the fixture design problem, the user must specify critical points and tolerances, and set the algorithm control flags.

Due to variations in the casting process, the workpiece shape uncertainty is $\epsilon_{\text{shape}} = \pm 0.004\text{in}$. Meanwhile, a few key workpiece features must be located with precisions of $\pm 0.02\text{in}$ relative to the casting boundary; these form the datums from which other features are referenced. These features define the critical point set $\{\mathbf{P}_i\}$.

In the final step of setting up the problem, the user sets the algorithm control flags as follows:

use-non-silhouette-walls?	=	`false`
d_{trim}	=	1mm
side-contact-height-preference	=	`coplanar`
max-clamp/normal-angle	=	45°
z-constraint-on-grid?	=	`true`
top-clamps?	=	`true`
d_{xy}	=	1mm
d_z	=	1mm
easy-loading-required?	=	`true`
CG-support-required?	=	`true`
d_{load}	=	0.5mm
w_F	=	0.5
w_P	=	0.5
w_L	=	0
w_{xy}	=	0.5
w_z	=	0.5
w_{3d}	=	0
p	=	1.0

Most of these values are defaults that typically apply in a prototype machining scenario. The parameters that control loading analysis are set to require easy loading with minimal loading clearance; this constrains the algorithm to look for fixtures that may be loaded without removing and replacing top clamps, but does not require a substantial loading clearance to compensate for manipulator uncertainty. The user may elect to relax this requirement if the returned fixtures are not satisfactory. Similarly, the parameter *use-non-silhouette-walls?* is set to false to focus on fixtures with widely distributed fixels that are easy to see when loading the fixture manually. This restriction may also be relaxed to improve quality values.

Given this input, the algorithm returns ... nothing. Our implementation includes a progress window that informs the user of the computation status. This window initially indicates a pre-processing phase, which is over in about 15 seconds. Then the progress window indicates the total number of edge triples (in this case 26,182), the percentage of these triples that have been analyzed, and the total number of fixtures that have been generated thus far. The percentage and number of fixtures generated increment as the computation proceeds, giving the user a sense of the computation progress.

Usually the algorithm returns initial fixture designs within the first minute or two, so when this time passes without a displayed fixture design, the user halts execution to investigate. Our implementation accumulates statistics summarizing the reason that candidate fixtures are discarded. After roughly two minutes of computation, these statistics are:

Number of xy-constraints	41
after loading analysis	22
after force analysis	22
after position analysis	22
Number of z-constraints	150
after loading analysis	1
after force/tip analysis	0
Number of 3d-fixtures	0
after loading analysis	0

These numbers indicate that there are plenty of candidate *xy*-constraints that pass the loading, force, and position thresholds, and plenty of candidate *z*-constraints. However, almost all *z*-constraints are rejected in the loading analysis. The reason for this is that a large portion of the workpiece undersurface cannot be used for support, due to the holes that will be drilled. This makes it difficult to find *z*-constraint triangles that support the workpiece center of mass. This characteristic is not required to load the fixture manually, because the user can hold the workpiece in place while closing the clamps. Thus the user sets *CG-support-required?* to false, and starts the computation again.

After two minutes, the algorithm still doesn't produce any solutions! Once again, the user halts the program, and checks the computation statistics:

Number of xy-constraints	42
after loading analysis	23
after force analysis	23
after position analysis	23
Number of z-constraints	154
after loading analysis	154
after force/tip analysis	0
Number of 3d-fixtures	0
after loading analysis	0

Now all *z*-constraints pass the loading analysis, but then fail the *z*-constraint force analysis. This is because the required contact reaction forces exceed the yield strength of the material at the supports and top clamps. Even though these contacts allow 300lb of force and the maximum *z*-force is only 285lb, the workpiece geometry and applied forces combine to produce required reaction forces well in excess of 300lb. In fact, it turns out that after several hours, the computation would verify that no fixture design can solve this problem under the current settings.

However, the user doesn't wait to confirm that outcome. Instead, the user selects support pads with flat tops instead of the default self-aligning tops. These supports do not have the self-aligning feature, but consequently have a contact pad with twice the diameter of the self-aligning pad. This increases the contact area by a factor of four, and thus also the maximum allowable force. The user also selects clamp arms with a larger contact pad, and starts the algorithm again.

Now a fixture design appears after 18 seconds of computation, immediately followed by a few closely-related designs. Figure 1 shows one of these early designs. This fixture satisfies all of the task requirements, and has a reasonably-good quality score of 0.52. In practice, quality scores rarely exceed 0.8; for this example the global optimum quality turns out to be 0.75, and appears in 10.9 minutes.

Our implementation includes a plot that shows the best quality seen so far as a function of time; after watching this plot remain flat for a while, the user decides to accept the current best design and prints out instructions on how to assemble the fixture. It takes roughly five minutes to assemble and load the fixture, so the total time required for computation and fixture assembly is approximately ten minutes.

This completes our explanation of how the fixture design algorithm could be applied to an example prototype fabrication problem. In the next section, we will examine the same problem from the perspective of automated mass production.

Mass Production

Prototype production is characterized by small production volumes and labor-intensive manufacturing methods. Thus, reducing fixture design and fabrication time are significant concerns. In the mass production phase, automated manufacturing methods are often employed to reduce costs, improve productivity, and improve process consistency. Thus in mass production, the part shown in Figure 28 is likely to be produced by automated casting and machining methods.

The fixture design algorithm may be applied to this problem in the same way as in the prototype fabrication problem, with a few modifications. The most obvious of these modifications is that the manual clamps employed in the prototype fabrication fixture should be replaced with automatic power clamps. In addition, because the fixture will be fabricated from plain tooling plate to reduce cost, fixture elements may be placed at arbitrary positions. Another difference is that easy loading is a strict requirement, not just an option. This includes mandatory CG support and a loading clearance larger than the manipulator position uncertainty. Finally, because more time is available to analyze the fixture, the computation may be allowed to run to completion to produce the global optimum fixture design.

Replacing the manual clamps with power clamps requires consideration of the applied clamp force. Thus the algorithm inputs F_{plunger} and F_{top} should be set to the forces exerted by the corresponding fixture elements; how should these forces be determined?

In both cases the possible choices are limited by the clamp hardware that is available. Power clamping components are available from a variety of companies, and usually can be configured for either pneumatic or hydraulic actuation. The pressures supplied by typical hydraulic power units range between 1000psi and 5000psi; air pressure is typically supplied at 100psi.

The side clamp shown in Figure 10(a) has a piston area of 0.78in^2. This results in clamp forces ranging between 780lb and 3900lb for hydraulic actuation, and 78lb for pneumatic actuation. Because $F_{\text{limit}_{\text{side}}} = 150\text{lb}$, this clamp obviously must be actuated pneumatically.

The top clamp shown in Figure 15(a) has a piston area of 0.098in^2, and thus provides clamping forces ranging between 98lb and 490lb with hydraulics, and 9.8lb with pneumatics. Because the drilling operations produce as much as 57lb of upward force, these clamps obviously require hydraulic actuation.

What hydraulic pressure should be chosen for the top clamps? The answer depends on the force required to prevent workpiece motion, the maximum contact force that avoids local workpiece deformation, and the load bearing capacity of the clamp. This capacity varies with the clamp arm length, as shown in Figure 15(d). This plot indicates the reduction in available clamping force that occurs as the arm length increases; this reduction is due to internal clamp friction, and limits imposed to prevent premature wear of the clamp seals and shaft bearings.

This hardware constraint complicates the fixture design problem by coupling the choice of clamp force and clamp arm length; shorter arm lengths produce more force, but also require the clamp body to be placed closer to the workpiece, increasing the chance of interference. Below we will show how the fixture design algorithm may be used to resolve this trade-off.

To setup the fixture design algorithm for this problem, the user sets the same inputs as in the prototype fabrication scenario, with the following exceptions:

$$\begin{aligned}
\mathcal{C}_{\text{only-loading}} &= \mathcal{V}_{\text{gripper}} \\
\textit{max-clamp/normal-angle} &= 30^\circ \\
\textit{z-constraint-on-grid?} &= \texttt{false} \\
d_{xy} &= 2.5\text{mm} \\
\textit{easy-loading-required?} &= \texttt{true} \\
\textit{CG-support-required?} &= \texttt{true} \\
d_{\text{load}} &= 1.0\text{mm}
\end{aligned}$$

where $\mathcal{V}_{\text{gripper}}$ denotes the volume occupied by the manipulator's gripper when loading the fixture, including the region swept as the fingers move from the closed to open position. This volume may or may not be known, depending on whether the user chooses to design the gripper before the fixture, or vice versa. If the user plans to design the gripper later to avoid the fixture, then $\mathcal{C}_{\text{only-loading}} = \emptyset$. The *max-clamp/normal-angle* is reduced to avoid side loads that may cause premature wear of the side clamp seals and shaft bearing. We repeat the terms *easy-loading-required?* and *CG-support-required?* to emphasize that these must be set to true, and are not optional. The value of d_{load} is chosen larger than the manipulator uncertainty to compensate for orientation errors, and d_{xy} is set to 2.5 times this value according to a rule of thumb intended to avoid interference when the workpiece is in the loading pose.

After setting these inputs, the user sets the side clamp force F_{plunger} to 78lb, allowing standard air pressure to be used without a step-down pressure regulator. For the top clamp, the minimum practical arm length is 1.25in, given the clamp body and pad diameters. For this length, the corresponding clamp force is 360lb from Figure 15(d). The user sets l_{arm} and F_{top} accordingly, and starts program execution.

It turns out that no z-constraint is possible for this arm length, due to clamp body interference. Because *z-constraint-on-grid?* is false, the algorithm first constructs the set of all possible valid z-constraints, finds none, and returns in 40 seconds. This gives the user immediate feedback that the chosen clamp arm length is unacceptable.

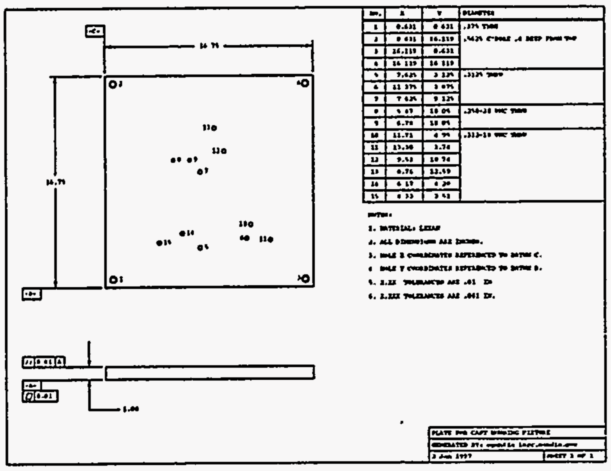


Figure 35: A mechanical drawing of the fixture plate shown in Figure 2. This drawing was automatically generated by the program.

Next the user tries a longer arm length $l_{\text{arm}} = 2.0$in, with a corresponding force of $F_{\text{top}} = 185$lb from Figure 15(d). This time a valid z-constraint is found, and the algorithm returns an initial fixture design within 66 seconds. This informs the user that feasible fixture designs are possible, so they let the program run to completion. After 61 minutes of computation, the program produced the global optimum fixture design for this problem, shown in Figure 2. This fixture should be used with 100psi air pressure to the side clamp, and 2650psi hydraulic pressure to the top clamps. Figure 35 shows a mechanical drawing of the fixture plate that was automatically generated by the program.

After pruning, the algorithm returned 4 fixture designs, including the global optimum. With $p = 0.97$, the algorithm produced 34 designs, providing more choices for the user to consider. With $p = 0.0$, the algorithm produced 170 total designs.

Figure 2 also shows the loading pose for the global optimum fixture, with the associated arm retraction directions. Successful loading of this fixture was verified with a robot manipulator over a series of repeated trials (Figure 36). This experiment included electrical continuity testing to verify that each locator contact was properly made.

7.2 Light Mechanical Assembly

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

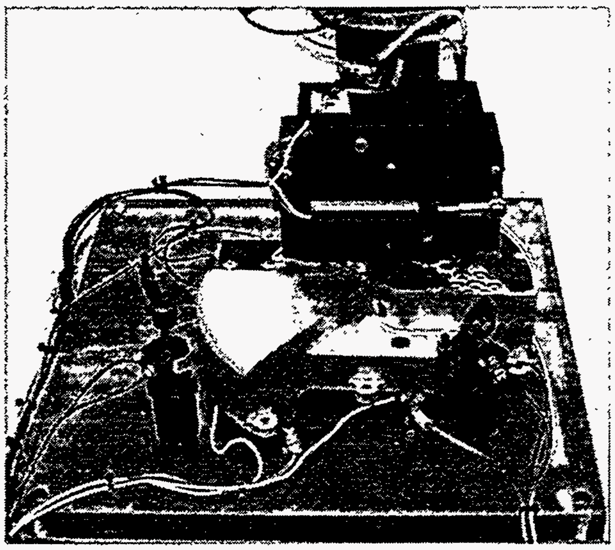


Figure 36: A robot loading the fixture shown in Figure 2.

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 most or all of the part insertions are vertical. This in turn allows the application of any of a number of vertical insertion manipulators, varying in complexity from pneumatic pick-and-place devices to SCARA manipulators. For these tasks, it is desirable that the assembly fixtures also be loaded and unloaded by vertical motions.

Assembly Pallet Design

The cost of fabricating assembly fixtures is a primary concern, because assembly lines often require many copies of these fixtures to carry the assembly from station to station. Thus assembly fixtures 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 act as supports, while the pins with conical tips provide xy-constraint and guide the part into place during vertical part loading.

These pallets are very inexpensive to produce. In cases where product changes are infrequent, they may be fabricated by drilling the required holes in plain tooling plate, and pressing the pins into place. If product changes occur frequently, then the pins may be screwed into plates with a grid of pre-drilled holes, with a special adjustable pin used for the pin that acts as a side clamp.

Our algorithm may be used to design assembly pallets, by defining an appropriate fixture kit. The side locators of this kit are specified with $h_{\text{contact}} = 0$, which indicates that the fixel is a continuous rod which allows arbitrary contact heights. The side clamp is defined with a null body and plunger, and travel range of $[0, d_{\text{grid}}]$. Choosing $y_{\text{offset}} = \frac{d_{\text{grid}}}{2}$ causes the "side-clamp-fixel" to vary along lines halfway between grid points; if $2r_{\text{locator}} + d_{\text{separation}} < \frac{d_{\text{grid}}}{2}$, then interference will never occur between the side clamp fixel and the side locator fixels. The support pad is defined with $r_{\text{support-pad}} = r_{\text{support-body}}$ and $h_{\text{support}} = 0$; this allows support pads of arbitrary height. No top clamps are included in the fixture kit.

Given this fixture kit, the user may setup the fixture design algorithm to design a pallet for the cassette chassis assembly shown in Figure 3(a). As in the machining case, the user must provide a CAD model of the workpiece and specify geometric access constraints, expected forces, force limits and position tolerances, and control flag settings. The following paragraphs will describe these inputs for this assembly example.

If all insertions are vertical and there are no top clamps, the geometric access constraints may be specified simply by forming the volumetric union of the parts that are added during assembly. Otherwise, the geometric access constraints are produced by forming the swept volume of the parts as they move along their insertion paths. If the grippers that hold the parts are defined before the fixture is designed, then the volume swept by these grippers may also be included; otherwise the grippers may be designed later to avoid the fixture, as shown in [13]. The resulting ensemble of volumes then forms the input set $\mathcal{C}_{\text{always}}$; the input sets $\mathcal{C}_{\text{only-loaded}}$ and $\mathcal{C}_{\text{only-loading}}$ are set to $\emptyset$.

As explained in the introduction, the electric motor in this assembly must be supported from below before it is fastened to the chassis with two screws. Because our algorithm only analyzes the constraint of a single base part, this implies that either the pallet will require a manually-designed support for the electric motor, or the motor must be assembled to the chassis before it is loaded into the pallet. We can ensure that the algorithm will leave room for a manually-designed support by including an additional constraint volume in $\mathcal{C}_{\text{always}}$, delineating the space beneath the motor. The resulting set of constraint volumes is shown in Figure 37.

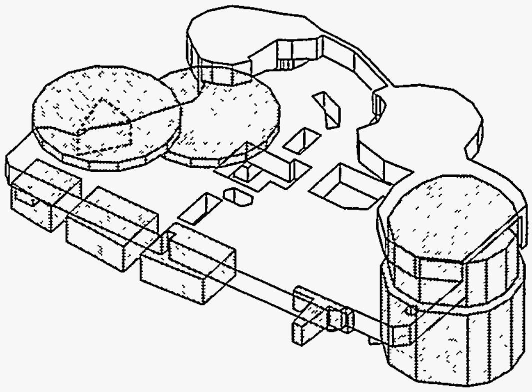


Figure 37: Assembly constraint volumes.

After defining the task geometric access constraints, the user must define the expected applied forces. As in the machining case, we envision a future system that will calculate these forces automatically, based on an input description of the planned assembly operations.

In the meantime, we can define these forces manually. There are three types of forces that arise during this assembly: downward forces as parts are inserted, torques about z as the screws are driven, and lateral acceleration forces that occur during pallet transfer movements.

The downward forces are exerted at the point where each part is inserted, as shown in Figure 38. The maximum insertion force magnitude is determined by the final force applied by the manipulator to confirm proper insertion. Here this force is $F_{\text{insert}} = 4\text{lb}$. This allows us to construct the total applied force $\mathbf{F}_i$ at each assembly site (x, y):

$$\begin{aligned} F_x &= 0 \\ F_y &= 0. \\ F_z &= -F_{\text{insert}} \\ \tau_x &= F_z y \\ \tau_y &= -F_z x \\ \tau_z &= 0 \end{aligned}$$

Each of these forces is augmented by the subassembly weight w and center of mass location $(x_{\text{com}}, y_{\text{com}})$ reflecting the state of the assembly when the insertion is performed. The downward forces are omitted for the screw placement sites, because these forces will be resisted by the custom-designed motor support.

The torques about z exerted when the screws are tightened are straightforward: $\tau_z = -0.8\text{in}\cdot\text{lb}$ exerted by the automatic screwdriver.

Similarly, the pallet transfer forces are determined by the acceleration and deceleration experienced by the pallet as it exits and enters a workcell. The direction of these accelerations depends on the application scenario. If the pallet will be fabricated by screwing pins into a pallet carrier with a pre-drilled hole grid, then the part orientation is determined by the locator setup, and the acceleration directions are fixed relative to the grid. If the pallet carrier does not include a pre-drilled grid, then the pallet design may be placed on the pallet carrier in an arbitrary orientation. When this is an option, the user may specify a particular part orientation to satisfy other workcell constraints. In this case the acceleration direction is fixed relative to the part. Another possibility would be to select the part orientation that allows the most copies of the pallet to be packed onto the pallet carrier. We envision a future system that will explicitly consider these different scenarios, but for now simply adopt the conservative approach of representing the peak acceleration in all directions.

For example, if the assembly line uses a Bosch pallet-transfer system, the peak acceleration occurs when the pallet hits a motion stop after entering the workcell. We measured this acceleration and found it to be approximately 15g, opposite the direction of pallet motion. This corresponds to an applied force on the part of $F_{\text{transfer}} = m\cdot 15g$, where m is the subassembly mass. To model this force in all lateral directions, we discretize $[0, 2\pi]$ in $\frac{\pi}{8}$ increments and construct an applied force

$$\begin{aligned}
F_x &= F_{\text{transfer}}\cos(\nu)\\
F_y &= F_{\text{transfer}}\sin(\nu)\\
F_z &= 0\\
\tau_x &= -F_y z_{\text{com}}\\
\tau_y &= F_x z_{\text{com}}\\
\tau_z &= F_y x_{\text{com}} - F_x y_{\text{com}}
\end{aligned}$$

where ν is the force direction, and $(x_{\text{com}}, y_{\text{com}}, z_{\text{com}})$ is the subassembly center of mass.

This completes our description of the applied forces; we now proceed to specify force limits. When there are no top clamps, the force-based quality metric $\mathtt{Q}_{xy_F}$ applies the usual force threshold calculation, while the $\mathtt{Q}_{z_F}$ metric applies a tipping analysis based on the required excess CG weight. Thus the user must specify the force limits $F_{\text{limit}_{\text{locator}}}$ and $F_{\text{limit}_{\text{side}}}$ as before, and then specify a maximum allowable required excess CG weight w_{limit}.

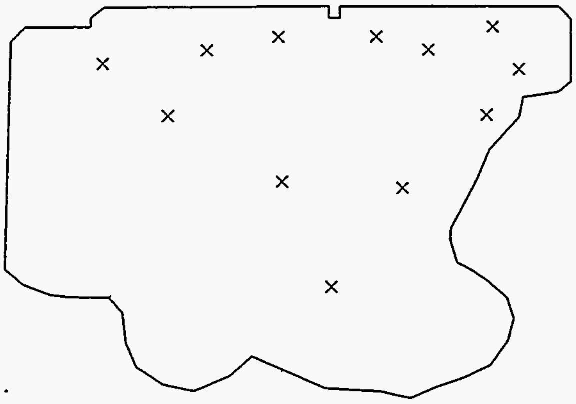


Figure 38: Sites where downward forces are exerted during assembly.

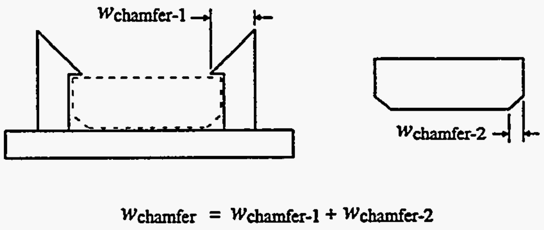

$w_{\text{chamfer-1}}$
$w_{\text{chamfer-2}}$
$w_{\text{chamfer}} = w_{\text{chamfer-1}} + w_{\text{chamfer-2}}$

Figure 39: Parameters describing the compliant insertion motion.

The xy-force limits could be determined using a deformation analysis as described above. Unfortunately, this would yield a limit much larger than any applied lateral force, thus concentrating $q_{F_{xy}}$ values near 1.0 and reducing the discriminatory power of the force-based quality metric $\mathtt{Q}_{xy_F}$. To prevent this, $F_{\text{limit}_{\text{locator}}}$ and $F_{\text{limit}_{\text{side}}}$ are set to a lower value of 5lb, which is ≈ 10 times the magnitude of the largest expected lateral force. If fixture designs fail to appear because they are discarded by $\mathtt{Q}_{xy_F}$, then these limits may be increased.

A good initial value for the maximum excess CG weight is $w_{\text{limit}} = 0$. This value causes all fixtures that result in tipping to be discarded, and assigns low q_{F_z} scores to fixtures that only marginally prevent tipping. If no fixtures are found with $w_{\text{limit}} = 0$, then runs with $w_{\text{limit}} > 0$ may be attempted to investigate fixture designs that marginally allow tipping.

The position-based quality metric plays an important role in assembly fixture design. If the locations of critical part features vary too widely, then the associated insertion operations will fail intermittently. To avoid this problem, the user constructs the critical point set $\{\mathbf{P}_i\}$ to include each of the assembly points shown in Figure 38, with associated position tolerance $\epsilon_{xy} = w_{\text{chamfer}}$, where w_{chamfer} is the effective width of the chamfers that funnel the part into position during insertion (see Figure 39). In this case w_{chamfer} varies between 0.04in and 0.05in.

In the final step of setting up the problem, the user sets the algorithm control flags as follows:

$$\begin{aligned}
\textit{use-non-silhouette-walls?} &= \texttt{false} \\
d_{\text{trim}} &= 0.25\text{mm} \\
\textit{side-contact-height-preference} &= \texttt{coplanar} \\
\textit{max-clamp/normal-angle} &= 90^\circ \\
\textit{z-constraint-on-grid?} &= \texttt{false} \\
\textit{top-clamps?} &= \texttt{false} \\
d_{xy} &= 0.25\text{mm} \\
d_z &= 1\text{mm} \\
\textit{easy-loading-required?} &= \texttt{false} \\
w_F &= 0.5 \\
w_P &= 0.5 \\
w_L &= 0 \\
w_{xy} &= 0.4 \\
w_z &= 0.6 \\
w_{3d} &= 0 \\
p &= 1.0
\end{aligned}$$

Most of these values are defaults that typically apply for assembly pallet design. Note that the loading analysis is disabled, because we assume that the conical tips on the side location pins allow the part to be inserted vertically using a compliant motion. This strategy requires the radius of the side locator pins to be larger than the position uncertainty of the manipulator.

Given this input, the algorithm solves the problem the first time, generating an initial fixture design in 1.7 minutes and the global optimum design in 9.4 minutes after generating a total of 7 fixtures. This design is shown in Figure 3(b).

Mixed-Product Assembly

The preceding example shows how the fixture design algorithm may be used to design an assembly pallet for 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. In this case, the algorithm could construct a data structure that identifies the geometric features common to all products in the family, along with geometric volumes that enclose all of the features that differ. Then assembly pallets could be designed that contact only the common features, while avoiding all of the regions where features differ.

A more severe scenario occurs when the products are dissimilar, but a single manufacturing line is still 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.

An example is shown in Figure 4(a), where two dissimilar products are shown. Suppose a company wishes to develop a cost-effective assembly system that can switch between these products with minimal overhead. Our algorithm can be applied to this problem to design an assembly pallet that satisfies the assembly task requirements of both products.

Our approach to mixed-product fixture design is inspired by two observations about the nature of the fixture design problem defined above. First, many fixture elements lie on a grid. Second, there are often a large number of solutions for a given fixture design problem, even after satisfying practical constraints. The first observation suggests that fixtures designed for different problems may share coincident fixture elements. The second observation suggests that for a set of problems which each have a large number of solutions, it is reasonably probable that there exists some combination of compatible designs that share common fixture elements. Is this the case in practice?

We investigated this question by implementing a simple procedure to synthesize mixed-product pallet designs. Given problems A and B with workpieces $\mathcal{W}_A$ and $\mathcal{W}_B$ and task constraints $\mathcal{C}_A$ and $\mathcal{C}_B$, the procedure generates the set of all pallet designs that share at least one common locator pin, can hold either $\mathcal{W}_A$ or $\mathcal{W}_B$, and do not interfere with $\mathcal{C}_A \cup \mathcal{C}_B$.

The procedure solves this problem by first generating all possible solutions to problem A, producing m pallet designs. The procedure then generates all solutions to problem B, producing n pallet designs. The procedure then considers each of the resulting mn pallet pairs, and decides whether the pair of pallet designs are compatible. If the pallets are compatible, then they are merged to produce a mixed-product pallet design; otherwise, the pair is discarded. This produces the set of all mixed-product pallet designs that are possible, given the style of pallets that are generated and the procedure's notion of compatibility.

To decide whether a pair of pallets (P_A, P_B) is compatible, the procedure seeks a relative placement of the pallets which allows the pallets to share three, two, or one common locators. We denote these compatibility conditions as 3-, 2-, and 1-compatible, respectively.

To decide whether P_A and P_B are 3-compatible, the algorithm checks the triangles formed by the three fixed locators in each pallet to see if they are congruent. This condition is satisfied when all corresponding sides of the triangle have the same length, and one triangle is not a reflection of the other. Our implementation does not check triangles involving the fourth "clamp" locator, because these are unlikely to give rise to congruent triangles. If the locator triangles are congruent, then the algorithm transforms P_A to make its locators coincident with P_B, and then checks for interference between the resulting P'_A and $\mathcal{W}_B \cup \mathcal{C}_B$, and between P_B and $\mathcal{W}'_A \cup \mathcal{C}'_A$. If no interference is found, then P_A and P_B are 3-compatible.

The checks for 2-compatibility and 1-compatibility are similar, except that the transformation mapping P_A onto P_B is no longer unique. In the 2-compatible case, such a transformation only exists when each pallet has a pair of locators separated by a common distance. When this condition is met, there are two transformations which make these locators coincident, separated by 180 degrees. In the 1-compatible case, there are nine choices of locators to make coincident, and four orientations possible for each choice. Thus as the compatibility condition weakens, the freedom in merging pallet designs increases. The quality metric in our implementation seeks to minimize the distance between the workpiece center points, thus discouraging fixtures that essentially place the workpieces side by side.

When merging fixtures it may be necessary to adjust the height of $\mathcal{W}_A$ in order to minimize support interference. This could be accomplished by intersecting the feasible height intervals for each locator. Our implementation does not include this calculation, but instead chooses the nominal height of $\mathcal{W}_A$ and $\mathcal{W}_B$ to place the primary support surface of both objects at a common height.

After generating all compatible design pairs, the procedure groups the pairs according to 1-, 2- or 3-compatibility. The procedure then scores and sorts the members of each group, using a weighted sum of the minimum single-fixture quality score and a measure that seeks to minimize the distance between workpiece center points.

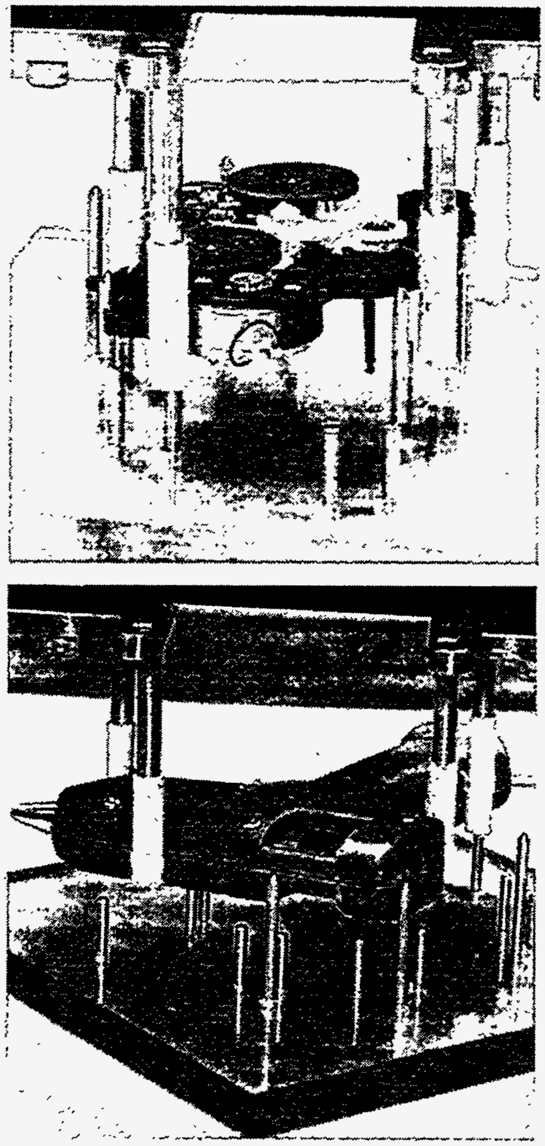


Figure 40: A robot loading a mixed-product pallet designed by the program.

Figure 4 shows the result of applying this procedure to our example problem. Part (b) shows the top-ranked 3-compatible design pair, and part (c) shows the physical pallet. This pallet may be used to assemble either product, and allows switching between products in zero changeover time. This design reduces pallet fabrication costs by requiring only 15 pins and 26.6 in^2 of pallet space, instead of the 18 pins and 37.7 in^2 required by a pair of single-part pallets. This optimal mixed-part pallet design was found in 146 minutes, including fixture generation time. Figure 40 shows an experiment verifying robust loading of this fixture using a robot manipulator.

The table below shows the density of compatible pallet pairs for this problem. In this table, pallet pairs that are 3-compatible are also counted in the 2-compatible category. The number of 1-compatible pairs was not determined, but would be very large. All pallet pairs are 0-compatible, because P_A can always be placed next to P_B while avoiding interference.

3-compatible pairs	7
2-compatible pairs	19,709
Total *mn* pairs	1,084,287

This table shows that, at least for this example, compatible pallet designs for mixed-product assembly are readily available. This result is somewhat surprising, especially given the large constraint regions present in both task specifications.

The above data reflect the results of a first experiment in mixed-product fixture design. There are two reasons why these data should be augmented by further study of this problem: First, the sample size is small, because we only considered one example. Second, the mixed-product pallets were not restricted to fit within a pre-defined pallet area, as would occur in practice. Additional experiments would help us to learn whether the above solution densities are typical or rare.

8 Discussion

The fixture design algorithm described in this paper effectively designs fixtures to solve practical problems. All fixtures designed by the algorithm rigidly constrain the workpiece, are robust in the face of workpiece shape variations, are easy to load, and satisfy the force and position tolerance requirements of the task. The fixtures designed by the algorithm may be quickly fabricated using proven hardware components that are widely available. The algorithm quickly produces good initial solutions to a given problem; and finds the global optimum solution in a reasonable amount of time.

An approach that worked exceptionally well in developing this result was that we began our investigation by selecting a class of hardware components that were currently in use in industrial applications. This selection involved simultaneous consideration of hardware and algorithmic issues, and enabled us to produce a design algorithm that mapped readily onto practical applications. We plan to continue using this strategy in our future work.

There are several ways to improve this result. One approach would be to replace some or all of the approximations made by the algorithm with exact calculations. A second approach would be to develop algorithms with faster run time. A third approach would be to extend the functionality of the algorithm. Among these, extending the algorithm's functionality is by far the most important. The following sections will explore each of these issues.

Eliminating Approximations

The algorithm makes several approximations in performing its computation. These include restricting the set of side fixel heights that are considered, discretizing the set of candidate clamp body placements when z-constraint elements are allowed at off-grid locations, and applying a simplified analysis of the fixture loading operation.

These approximations may be removed by replacing the algorithm's approximate calculations with exact methods. In some cases this will incur significant additional computation. In our judgment, this is not worth the effort in most cases. From the user's perspective, the key limitations of this software are not these approximations, but the functionality limitations discussed below.

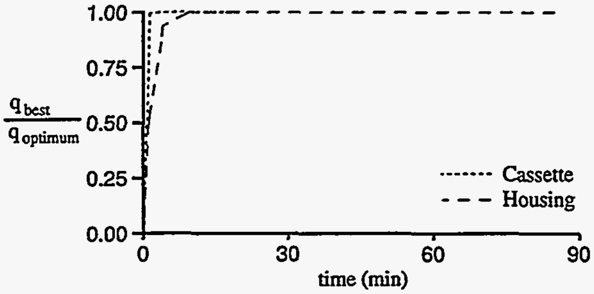


Figure 41: The run-time performance of the algorithm. q_{best} is the quality score of the best fixture generated so far; this plot shows how q_{best} improves with time as the computation proceeds. Because of the heuristics employed by the algorithm, high-quality fixture designs appear very early in the computation.

Reducing Run Time

Our algorithm is output-sensitive, with a run time that is dominated by the number of fixtures that may be output. This number may be quite large. From [10], there may be $O(n^4d^5)$ possible xy-constraints for a given problem, where n is the number of workpiece faces and d is the workpiece diameter in grid units. Similarly, there may be $O(d^6)$ possible z-constraints if elements are restricted to the grid, and $O[(n^2+d^2)^3]$ if elements are allowed off-grid. Thus the total number of possible designs is $O(n^4d^5 \cdot d^6) = O(n^4d^{11})$ for the on-grid case, and $O[n^4d^5 \cdot (n^2+d^2)^3] \approx O(n^{10}d^{11})$ for the off-grid case. For the housing example described in Section 7.1, this amounts to 10^{19} possible designs!

The various pruning techniques employed by the algorithm reduce this number dramatically, leading to practical run times of about a minute for high-quality initial fixture designs, and roughly an hour for global optimum solutions. Figure 41 shows a plot of best fixture quality as a function of time for the two examples presented in Section 7.

The philosophy behind this enumerative approach is based on the observation that real fixture design problems involve a myriad of practical considerations that must be simultaneously satisfied. This observation leads us to the enumerative approach for two reasons: First, the enumerative approach provides a generic fixture synthesis algorithm that allows us to include arbitrary collections of task constraints. Second, the enumerative algorithm returns multiple solutions, allowing the user to apply subjective evaluation criteria that are not explicitly modeled.

	Cast Housing	Cassette Chassis
No pragmatic constraints	25,910,753	26,742
With clamp body, reach, and travel limits	8,661,355	26,742
With robust form closure	847,204	3,683
With d_{xy}, d_z clearance	60,599	2,416
With constraint volumes C	60,599	185
Within force limits	13,098	48
Within position limits	12,040	48
With loading-pose and cg-support	1,244	48
With push-into-place loading	170	48
Fixtures generated with $p = 1$	4	7

Table 1: The effect of pragmatic constraints on design solution space.

Table 1 shows the effect of task constraints on the fixture solution space. The first line of the table shows the number of solutions to our two example problems before any task constraints are applied; this corresponds closely to the problem of synthesizing a grasp using ideal point contacts. The following lines show the reduction in the solution space as various task constraints are successively applied. Each line except the last reflects the number of fixtures generated by our algorithm with pruning turned off; the last line indicates the number of fixtures generated with branch-and-bound pruning enabled.

These data indicate a 10^6-fold reduction in the solution space that occurs when task constraints are considered. In this sense, the problem becomes harder in the face of these constraints. Thus alternative algorithms should be considered in the context of satisfying multiple task constraints before a meaningful run time comparison can be made. Further, if an alternative algorithm returns only the first fixture design, its run time should be compared to the time our algorithm requires to generate a single fixture design, and fixture quality should also be considered.

The table also illustrates the effectiveness of the branch-and-bound pruning method, which allows the algorithm to find the global optimum fixture design while generating only 2% of the possible feasible solutions.

With this in mind, we believe that developing faster synthesis algorithms in future work is not nearly as important as expanding the functionality of the algorithm; this is the topic of the next section.

Extending Functionality

Our current algorithm constructs global optimum fixture designs that satisfy multiple practical constraints and which can be easily fabricated using proven hardware components. In our conversations with potential users of the software, these characteristics have been met with enthusiasm.

Even so, we have identified several shortcomings of the algorithm that impede its application in practice. These shortcomings are ripe opportunities for future work.

A main area of improvement would be to extend the algorithm to employ other types of primitive fixture elements. We have encountered examples in the areas of machining, assembly, and inspection where the best fixture designs would be obtained using pin/hole or V-block location techniques, which our current algorithm does not consider. Alternative styles of clamping elements would also be helpful in some situations. Extending the algorithm to include such elements will raise an interesting question: How can we efficiently select the most appropriate components for a given problem?

In addition, several extensions would improve the algorithm's utility for machining applications. The most important of these is to interface the fixture design algorithm with a process planning facility. In many cases the fixture design problem is tightly coupled to the cutting plan, and sometimes fixture design constraints require modifications to the plan. Figure 28(b) shows an example that cannot be solved by our algorithm. If we plan to cut the underside gasket surface and mounting holes in one setup, there is no way to place top clamps to avoid interference with the machining operation. The solution to this problem is to partition the cutting operation into two or more parts, and reposition the top clamps between operations. This example motivates the coupling of fixture design and process planning; see [19, 20, 31, 66, 16] for work in this area.

Another extension that would aid machining applications would be to develop deformation analysis methods and strategies for minimizing deformation. Analysis methods might include lumped-parameter approaches based on classical beam theory, or finite element methods. See [21, 38] for examples. Possible strategies to avoid deformation include assuring that there is a continuous column of workpiece material between each support and its matching top clamp, or adding fixture elements that provide redundant support by gently contacting the workpiece and then locking in place.

The fixture design algorithm's relevance to machining problems would also be improved by extending the algorithm to analyze frictional contact when top clamps cannot be used, allowing the workpiece to rest directly on the fixture base plate, and providing an explicit analysis of datum surfaces.

In the context of assembly, a key limitation is that the algorithm only synthesizes constraint for a single part. Many assembly problems require several parts to be held in a desired relative position before fastening operations are applied. Our current algorithm can be applied to these problems by treating each part as a separate fixture design problem with the remaining parts as volumetric constraints. This will produce solutions in some cases, but the resulting designs may be expensive to fabricate because of the large number of required fixture elements. A better approach would be to exploit contacts between the parts, thus simplifying the fixture design. See [49, 5] for initial results in this area.

Finally, our mixed-product fixture design algorithm could be significantly improved. The above generate-and-test procedure is complete, but combinatorially expensive. Several alternative strategies merit future study:

1. *Exploit product families.* If the input fixture design problems $\{(\mathcal{W}_1,\mathcal{C}_1),(\mathcal{W}_2,\mathcal{C}_2),\ldots(\mathcal{W}_n,\mathcal{C}_n)\}$ are drawn from a family of closely-related products, mixed-product fixture designs may be directly constructed using the normal fixture design algorithm. This is accomplished by modifying the algorithm pre-processing step to construct the set of common contact surfaces $\mathcal{S}^\cap = \mathcal{S}_1 \cap \mathcal{S}_2 \cap \cdots \cap \mathcal{S}_n$, where $\mathcal{S}_i$ is the set of contact surfaces for workpiece $\mathcal{W}_i$. The pre-processing step should also construct the maximum material condition workpiece $\mathcal{W}^\cup = \mathcal{W}_1 \cup \mathcal{W}_2 \cup \cdots \cup \mathcal{W}_n$ and similarly the overall constraint $\mathcal{C}^\cup$. This transforms the mixed-product fixture design problem into the ordinary fixture design problem of synthesizing a fixture that constrains the workpiece by contacting surfaces in $\mathcal{S}^\cap$ while avoiding interference with $\mathcal{W}^\cup \cup \mathcal{C}^\cup$. This synthesis procedure does not consider as many design possibilities as the generate-and-test procedure described above, but has the advantage of placing every $\mathcal{W}_i$ in the same pose.

2. *Defer support synthesis until compatibility analysis.* In this approach, the normal design algorithm is used to generate xy-constraint designs, which are then combined and checked for compatibility as before. However, support synthesis is deferred until compatible xy-constraints are found, and then the relative placement of the workpieces is used to directly synthesize compatible support locations. For example, suppose workpieces $\mathcal{W}_1$ and $\mathcal{W}_2$ have support surfaces $\{\mathcal{S}_{1a},\mathcal{S}_{1b}\}$ and $\{\mathcal{S}_{2a},\mathcal{S}_{2b}\}$, where $\mathcal{S}_{1a}$ is lower than $\mathcal{S}_{1b}$, and $\mathcal{S}_{2b}$ is lower than $\mathcal{S}_{2a}$. Further, suppose that for a given mixed-product xy-constraint, the projections of $\mathcal{S}_{1a}$ and $\mathcal{S}_{2a}$ intersect. The algorithm may then adjust the workpiece heights so that $\mathcal{S}_{1a}$ and $\mathcal{S}_{2a}$ are at the same height, allowing common supports to be placed anywhere in $\mathcal{S}_{1a} \cap \mathcal{S}_{2a}$. This choice places $\mathcal{S}_{1b}$ above $\mathcal{S}_{2b}$, so supports for $\mathcal{W}_2$ may be placed anywhere in $\mathcal{S}_{2b}$, and supports for $\mathcal{W}_1$ may be placed anywhere in $\mathcal{S}_{1b} - \mathcal{S}_{2b}$. Of course, interference with the workpiece and constraint volumes must also be checked.

3. *Modify side clamp synthesis.* The procedure that generates planar fixture designs uses an analysis of planar forces to directly construct side clamp positions that provide form closure. This procedure may be revised to directly synthesize mixed-product fixture designs with up to four common side locating pins. This is accomplished by synthesizing locator setups for $\mathcal{W}_1$ and $\mathcal{W}_2$ that have two or three locators in common, and then placing each workpiece in contact with the locators and projecting its vertical faces onto the xy-plane. Intersections of the resulting projected edges indicate candidate placements of the fourth pin, as indicated in Figure 42. Each candidate clamp location must be checked to see if it provides form closure for both $\mathcal{W}_1$ and $\mathcal{W}_2$; if it does, then the locator setup and clamp pins comprise a mixed-product xy-constraint that is then subjected to support and interference analysis. This approach has the advantage that it can synthesize 4-compatible fixture designs. We thank Matt Mason for this suggestion.

Combining all three of these ideas could lead to significant improvement in both computational efficiency and in the number of required pins.

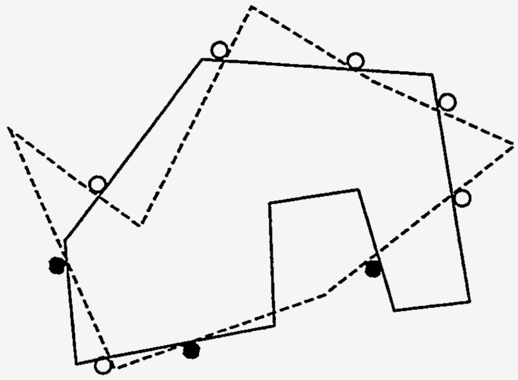


Figure 42: Direct synthesis of mixed-part xy-constraints. Solid circles indicate positions of the three common locators. The open circles indicate possible positions for the fourth locator.

Acknowledgments

We thank Ron Simon for constructing the physical demonstration fixtures and pallets, offering comments on this work, and for writing the robot control software for the fixture loading experiments. Sincere thanks go to Russell Brown, Yong Hwang, Matt Mason, Paul Plomp, John Rueping, David Strip, Aaron Wallack, and Randy Wilson for helpful discussions related to this work, and to our colleagues Gary Snavely, Jim Wells, and Steve Holland at General Motors for supporting and inspiring this work.

Appendix A: Pose Calculation

Given an edge triple and three candidate locator positions, the function

$$\mathtt{pose}(e_1, e_2, e_3, x_1, y_1, r_1, x_2, y_2, r_2, x_3, y_3, r_3)$$

calculates the set of (x, y, θ) poses where each edge is in contact with its associated locator. Here we explain our method for performing this calculation, which is an improvement over the method reported in [10]. We thank Aaron Wallack for suggesting this approach, which is similar to the method employed in [76]. Richard Wagner also used a similar method in his implementation of the 2-d fixture algorithm, based on a formulation developed by Xiaofei Huang [74]. These methods are closely related to the algebraic analysis techniques presented in [23].

We actually calculate the (x, y, θ) poses where each locator contacts the infinite line containing each edge, and then discard poses where a locator makes contact outside the edge endpoints. We will ignore this detail from here on, and simply explain the solution to the infinite line case.

The pose (x, y, θ) describes the transformation of the workpiece edges that will place them in contact with the fixed locators. Instead of solving this problem directly, we first solve the inverse problem, finding a pose (x', y', θ') which transforms the locators onto the fixed workpiece edges, and then invert (x', y', θ') to find (x, y, θ). This approach simplifies the algebra significantly.

Figure 43 shows the setup for the inverted problem. Here each edge e_i is fixed in space, described by constant $A_i x + B_i y + C_i = 0$ homogeneous line equations. These equations are oriented so that positive values of $A_i x + B_i y + C_i$ correspond to points on the outside side of e_i. The locator centers are functions of the pose variables (x', y', θ'):

$$\begin{aligned} x'_i(x', y', \theta') &= x_i c_{\theta'} - y_i s_{\theta'} + x' \qquad (6) \\ y'_i(x', y', \theta') &= x_i s_{\theta'} + y_i c_{\theta'} + y' \qquad (7) \end{aligned}$$

where $c_{\theta'}$ and $s_{\theta'}$ refer to $\cos(\theta')$ and $\sin(\theta')$, and (x_i, y_i) is the untransformed position of locator i. The locator will contact the edge exactly when

$$A_i x'_i + B_i y'_i + C_i = r_i,$$

where r_i is the locator radius. The desired poses are those which satisfy this condition for all three locators.

Thus we wish to find the values of (x', y', θ') that simultaneously solve the following equations:

$$\begin{aligned} A_1 x_1'(x', y', \theta') + B_1 y_1'(x', y', \theta') + C_1 &= r_1 \\ A_2 x_2'(x', y', \theta') + B_2 y_2'(x', y', \theta') + C_2 &= r_2 \\ A_3 x_3'(x', y', \theta') + B_3 y_3'(x', y', \theta') + C_3 &= r_3 \end{aligned}$$

Substituting equations (6) and (7) and re-arranging gives

$$\begin{aligned} D_1 c_{\theta'} + E_1 s_{\theta'} + A_1 x' + B_1 y' + F_1 &= 0 \quad (8) \\ D_2 c_{\theta'} + E_2 s_{\theta'} + A_2 x' + B_2 y' + F_2 &= 0 \quad (9) \\ D_3 c_{\theta'} + E_3 s_{\theta'} + A_3 x' + B_3 y' + F_3 &= 0 \quad (10) \end{aligned}$$

where

$$\begin{aligned} D_i &= A_i x_i + B_i y_i \\ E_i &= B_i x_i - A_i y_i \\ F_i &= C_i - r_i \end{aligned}$$

This gives us three equations in four unknowns: x', y', $c_{\theta'}$, and $s_{\theta'}$. If we add the identity $c_{\theta'}^2 + s_{\theta'}^2 = 1$, then we have four equations and four unknowns. Three of these equations are linear, and one is quadratic.

We can use the three linear equations to eliminate the x' and y' variables, producing two equations:

$$\begin{aligned} G c_{\theta'} + H s_{\theta'} + J &= 0 \quad (11) \\ c_{\theta'}^2 + s_{\theta'}^2 &= 1 \quad (12) \end{aligned}$$

where

$$\begin{aligned} G &= K(B_1 D_3 - B_3 D_1) + L(B_2 D_1 - B_1 D_2) \\ H &= K(B_1 E_3 - B_3 E_1) + L(B_2 E_1 - B_1 E_2) \\ J &= K(B_1 F_3 - B_3 F_1) + L(B_2 F_1 - B_1 F_2) \\ K &= B_1 A_2 - B_2 A_1 \\ L &= B_1 A_3 - B_3 A_1 \end{aligned}$$

Notice that the terms G, H, J, K, and L contain no denominators, so they are never ill-conditioned.

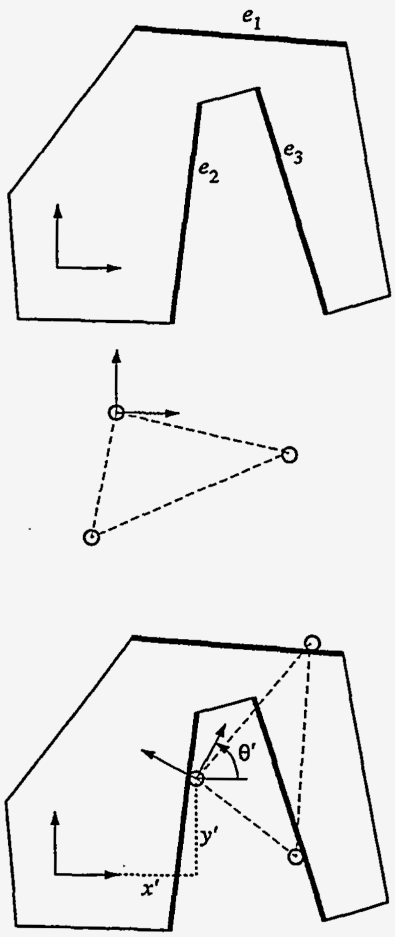


Figure 43: Pose analysis parameters. The parameters x', y', and θ' specify the position of the locator set with respect to the workpiece.

One way of solving these equations is to first solve (11) for $s_{\theta'}$, and substitute the result into (12). This yields

$$s_{\theta'} = \frac{-G c_{\theta'} - J}{H} \qquad (13)$$

$$(G^2 + H^2) c_{\theta'}^2 + 2GJ c_{\theta'} + (J^2 - H^2) = 0 \qquad (14)$$

We can then solve (14) using the quadratic formula, producing up to two resulting $c_{\theta'}$ values. Each $c_{\theta'}$ value corresponds to a valid placement of the locators on the three infinite lines. We then substitute each $c_{\theta'}$ solution into (13) to obtain $s_{\theta'}$, and then compute the x' and y' values from the linear equations (8–10).

Notice that if $H = 0$, then equation (13) becomes ill-conditioned. When this occurs, we apply a solution developed by solving (11) for $c_{\theta'}$ instead of $s_{\theta'}$. This yields an alternative pair of equations

$$c_{\theta'} = \frac{-Hs_{\theta'} - J}{G} \tag{15}$$

$$(G^2 + H^2)s_{\theta'}^2 + 2HJs_{\theta'} + (J^2 - G^2) = 0 \tag{16}$$

which are ill-conditioned when $G = 0$. To maximize the robustness of this computation, our implementation uses equations (15) and (16) when $|G| > |H|$, and equations (13) and (14) otherwise.

A similar numerical conditioning problem can occur when $G^2 + H^2 \approx 0$. The occurrence of this condition is sensitive to the order of equations (8–10), due to the variable elimination we employed to obtain (11). Our implementation avoids this problem by constructing G and H for the three orderings (8,9,10), (9,8,10), and (10,8,9), and using the ordering which produces the largest $G^2 + H^2$ value.

This calculation identifies up to two sets of values of the four unknowns x', y', $c_{\theta'}$, and $s_{\theta'}$. For each set, we then calculate $\theta' = \tan^{-1}\left(\frac{s_{\theta'}}{c_{\theta'}}\right)$ to obtain the inverse pose (x', y', θ'). This pose is then inverted to produce the desired (x, y, θ) pose by

$$\begin{aligned} x &= -x'c_{\theta'} - y's_{\theta'} \\ y &= x's_{\theta'} - y'c_{\theta'} \\ \theta &= -\theta' \end{aligned}$$

This method of solving for the (x, y, θ) workpiece pose is algebraically simpler than the method reported in [10], and avoids the need to treat parallel edges as a special case. For an extension of this approach to include circular edges, see [77].

Appendix B: Force Calculations

The quality metrics $\mathtt{Q}_{xy_F}$, $\mathtt{Q}_{z_F}$, and $\mathtt{Q}_{3d_F}$ compute quality scores based on the fixture's ability to resist applied forces without exerting large contact reaction forces on the workpiece. This analysis requires the following basic calculation: Given a fixture design and an applied force $\mathbf{F}_A = [F_{x_A}\ \ F_{y_A}\ \ F_{z_A}\ \ \tau_{x_A}\ \ \tau_{y_A}\ \ \tau_{z_A}]$, what are the contact reaction forces? This appendix explains how we perform this calculation.

With frictionless contacts, all z-constraint forces are parallel to the z-axis, and all xy-constraint forces are parallel to the xy-plane. This allows us to decouple our force analysis into separate calculations that consider only the xy or z components of the fixture design.

For the moment, let's assume that the xy-constraint contacts are points rather than vertical line segments, and that these points lie on a constant-z plane. Under these conditions, the xy-constraint contact forces exert no net moment about the x and y axes, and no force component in the z direction. This can be seen by observing that if we place the coordinate system origin in the constant-z plane, then all of the xy-constraint forces lie in the xy-plane, and thus exert zero moment about the x and y axes. Similarly, the z-constraint contact forces exert no moment about z, and have no force components in the x or y directions. Thus the xy-constraint can only affect the force components F_x, F_y, and τ_z, while the z-constraint contacts can only affect F_z, τ_x, and τ_y.

This allows us to write two independent sets of force-balance equations describing in-plane and out-of-plane equilibrium conditions. For the xy-constraint we have:

$$F_1c_1 + F_2c_2 + F_3c_3 + F_4c_4 + F_{x_A} = 0 \tag{17}$$

$$F_1s_1 + F_2s_2 + F_3s_3 + F_4s_4 + F_{y_A} = 0 \tag{18}$$

$$F_1l_1 + F_2l_2 + F_3l_3 + F_4l_4 + \tau_{z_A} = 0 \tag{19}$$

where indices 1, 2, and 3 refer to the side locator contacts, and 4 refers to the side clamp contact. The terms c_i and s_i are the cosine and sine of the angle of contact normal i, and l_i is the moment arm about z exerted by contact i. See Figure 44.

Similarly, for a z-constraint we can write:

$$F_5 + F_6 + F_7 + F_{z_A} = 0 \tag{20}$$

$$F_5y_5 + F_6y_6 + F_7y_7 + \tau_{x_A} = 0 \tag{21}$$

$$-F_5x_5 - F_6x_6 - F_7x_7 + \tau_{y_A} = 0 \tag{22}$$

where indices 5, 6, and 7 refer to the support/top-clamp contacts, and (x_i, y_i) is the location of contact i, measured in the same coordinate system as $\mathbf{F}_A$.

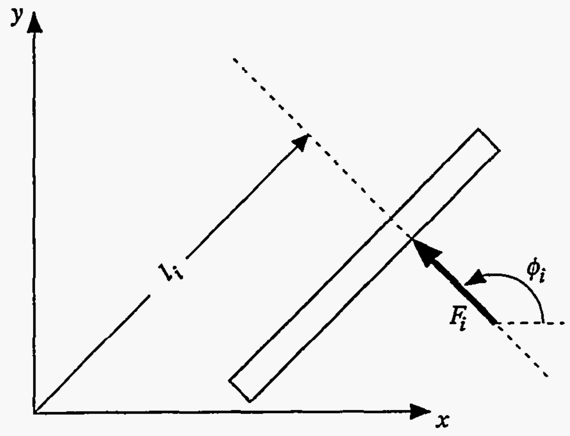
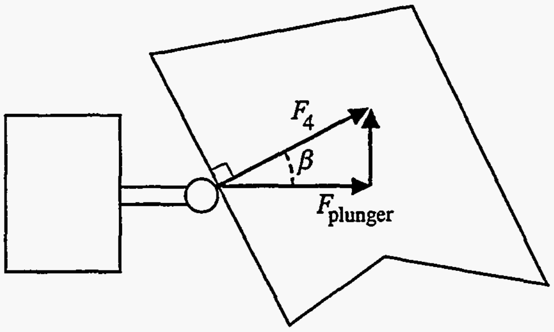


Figure 44: Force parameters describing an xy-constraint contact i. The terms c_i and s_i in equations (17–18) are the cosine and sine of ϕ_i.



$$F_4 = \frac{F_{\text{plunger}}}{\cos\beta}$$

Figure 45: Calculating the contact reaction force at the side clamp, given the side clamp thrust force F_{plunger}.

Equations (17–19) describe the in-plane equilibrium condition with three equations in four unknowns, and equations (20–22) describe the out-of-plane equilibrium condition with three equations in three unknowns. The following sections will explain our method for solving these equations. We will then proceed to explain how we handle situations where the side contacts are at different heights, and the case of fixtures without top clamps.

B.1 xy-Constraint Force Calculation

Equations (17–19) describe the in-plane equilibrium with three equations and four unknowns: F_1, F_2, F_3, and F_4. However, if the side clamp thrust force F_{plunger} is defined, then F_4 may be determined from F_{plunger} and the fixture geometry, as shown in Figure 45. Thus when F_{plunger} is defined there are only three unknowns, and the values of F_1, F_2, and F_3 may be found by using standard elimination methods.

If F_{plunger} is not defined, then the system is statically indeterminate. This is because there may be an arbitrary internal force resulting from positive contact forces that cancel to produce no net external force. To resolve this problem, we assume that this internal force is zero; that is, if no external force is applied, then all of the contact reaction forces are zero. Under this assumption, a non-zero applied force is resisted by contact reaction forces at three of the four contacts, with the fourth contact exerting zero reaction force. The contacts that generate non-zero reaction force may be identified by considering the arrangement of the corresponding vectors in the (F_x, F_y, τ_z) space, or simply by setting F_1, F_2, F_3, and F_4 individually to zero, solving equations (17–19) for the remaining contact forces, and taking the solution that produces non-negative reaction forces at each contact. The implementation in [10] used the former of these methods; our current implementation uses the latter for simplicity.

Our assumption of zero internal strain essentially treats the side contacts as ideal unilateral kinematic constraints that are perfectly placed. If the real side contact locations are slightly loose, then the workpiece will move slightly in response to the applied force, and the calculated contact reaction forces will be correct. If the real side contact locations are slightly tight, then non-zero internal strain will exist, and the calculated contact reaction forces will underestimate the true forces. Here we assume that this effect is small compared to the applied force magnitude; this assumption is realistic when the side clamp mechanism allows the clamp to be locked in position while applying only a very small force. The manual clamp shown in Figure 9(a) has this property.

In some cases it may be desirable to intentionally place the side locators to slightly compress the workpiece. For example, we experimented with assembly pallet designs where we gave the algorithm a value of r_{locator} that was slightly smaller than the true pin radius; this caused the algorithm to design pallets which held the part snugly with a positive squeezing force. To properly estimate the contact reaction forces for these pallets, our calculation should be extended to include this internal strain force, estimated from the compliance of the part material and the difference between the input r_{locator} and the true pin radius.

B.2 z-Constraint Force Calculation

Equations (20–22) describe the out-of-plane equilibrium condition with three equations in three unknowns. These equations may be solved directly to find F_5, F_6, and F_7. These force magnitudes may then be converted to the contact forces exerted by each fixture element.

If the top clamp force F_{top} is left undefined, then the supports and top clamps are treated as ideal kinematic constraints, and we assume zero internal strain. Thus for each support/top-clamp contact i, if $F_i < 0$, then we take $F_{\text{support}_i} = -F_i$ and $F_{\text{top}_i} = 0$; otherwise $F_{\text{support}_i} = 0$ and $F_{\text{top}_i} = F_i$.

If F_{top} is defined, then we replace each F_i in equations (20–22) with $(F_{\text{support}_i} - F_{\text{top}})$ and solve the resulting linear system for each F_{support_i} value. If any F_{support_i} is negative, then the top clamp is overwhelmed and the fixture is discarded.

B.3 Side Contacts at Different Heights

If the side locator contacts are at different heights, then the in-plane and out-of-plane constraint equations are no longer decoupled. However, it is still the case that the z-constraint contacts do not exert moments about z, or produce force components in the x or y directions.

If F_{plunger} is defined, then there is exactly one choice of contact reaction forces that solve equations (17–19). In this case we can identify all of the contact reaction forces by first solving equations (17–19) to obtain the contact reaction forces at all of the side contacts, and then apply the contact heights to determine the moments exerted by the side contact reaction forces

$$\begin{aligned}\tau_{x_{\text{side}}} &= -F_1 s_1 z_1 - F_2 s_2 z_2 - F_3 s_3 z_3 - F_4 s_4 z_4 \\ \tau_{y_{\text{side}}} &= F_1 c_1 z_1 + F_2 c_2 z_2 + F_3 c_3 z_3 + F_4 c_4 z_4\end{aligned}$$

where z_i is the height of contact i, which we will explain in a moment. We then form $\tau'_{x_A} = \tau_{x_A} + \tau_{x_{\text{side}}}$ and $\tau'_{y_A} = \tau_{y_A} + \tau_{y_{\text{side}}}$, replace the τ_{x_A} and τ_{y_A} terms in equations (20–22), and solve for the z-constraint forces as before. This identifies the set of contact reaction forces at all fixture contacts, including the moments exerted by the side contacts at different heights.

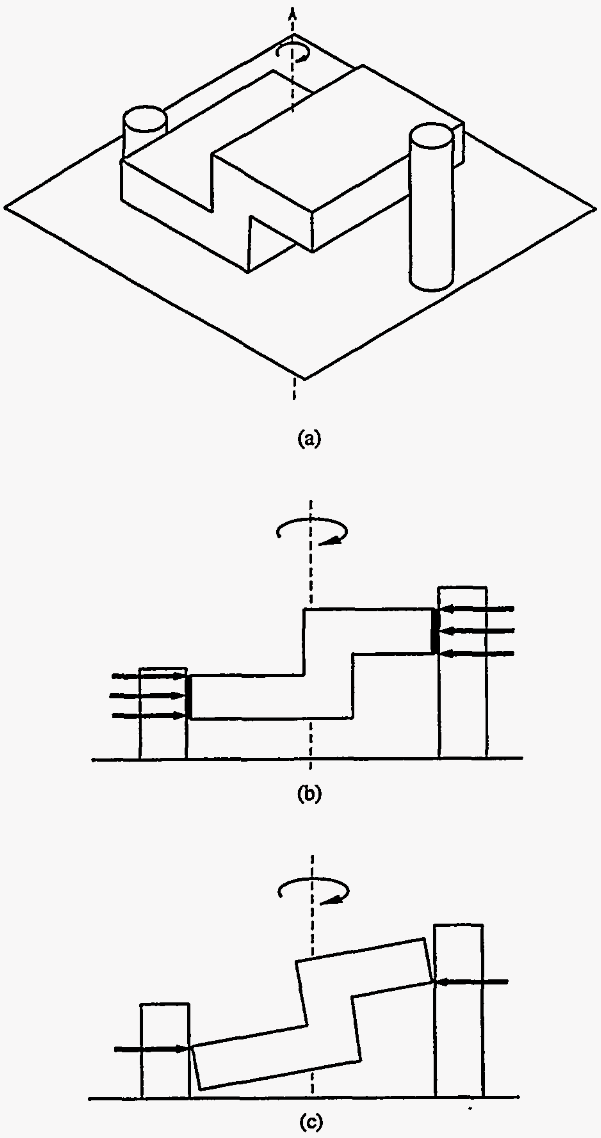
If F_{plunger} is not defined, then the underconstraint in the in-plane equations (17–19) leads to underconstraint in the out-of-plane equations as well. This is because the xy-constraint's internal strain can now be chosen to produce moments about x and y, thus altering the contact reaction forces of the z-constraint. We resolve this problem by again assuming that all contact reaction forces are zero in the absence of an external force. Under this assumption, there is only one consistent solution to the in-plane equations (17–19), which in turn determines specific out-of plane torques $\tau_{x_{\text{side}}}$ and $\tau_{y_{\text{side}}}$ which are added to the out-of-plane equations as described above.

These calculations require that we determine a particular height z_i for each side contact i. Because each fixel actually contacts the workpiece along a range of heights, the fixel makes a line contact with the workpiece, with an associated contact pressure distribution. Our single parameter z_i is a model of the lumped effect of this distribution, corresponding to the centroid of this pressure distribution. How do we determine this value?

Figure 46 shows a simple example that motivates our approach to this problem. The workpiece is subjected to a torque about z, which induces contact reaction forces in two side locators at different heights. Figure 46(b) shows a side view of this situation. The side locators make contact along line segments with associated height ranges $\mathcal{Z}_1 = [z_{\min_1}, z_{\max_1}]$ and $\mathcal{Z}_2 = [z_{\min_2}, z_{\max_2}]$. Regardless of the pressure distribution at each contact, the equivalent contact heights z_1 and z_2 must lie within these intervals. Thus if $\mathcal{Z}_1$ and $\mathcal{Z}_2$ do not overlap, then the contact reaction forces must exert a net moment, which tends to cause the workpiece to tip as shown in Figure 46(c). As the workpiece tips, it breaks contact at the most distant endpoints of $\mathcal{Z}_1$ and $\mathcal{Z}_2$, concentrating contact pressure at the near endpoints of $\mathcal{Z}_1$ and $\mathcal{Z}_2$, minimizing the induced moment.

Based on this observation, we choose to assume that the side contact pressure distribution will produce an aggregate contact height that minimizes the induced tipping moment. The idea behind this assumption is that if reaction forces induce an out-of-plane moment on the workpiece, it will tend to tip away from the contacts in response, thus transferring contact pressure to reduce the tipping moment. This assumption essentially amounts to a simplified model of object deformation; a more exact calculation appears to require the use of finite element methods.

We can apply this assumption to select contact heights for our example: If $\mathcal{Z}_1$ and $\mathcal{Z}_2$ overlap, we choose $z_1 = z_2 \in (\mathcal{Z}_1 \cap \mathcal{Z}_2)$, resulting in zero induced moment and decoupled in-plane and out-of-plane equilibrium equations. If $\mathcal{Z}_1$ and $\mathcal{Z}_2$ do not overlap, we choose the closest endpoints of $\mathcal{Z}_1$ and $\mathcal{Z}_2$ to minimize the resulting tipping moment. This method has the desirable feature of decoupling equations (17–19) and (20–22) whenever possible, while allowing a smooth transition to cases where the equations cannot be decoupled.



(a)

(b)

(c)

Figure 46: Tipping moments induced by side contacts at different heights. (a) A simple example. (b) Side view, showing possible contact pressure distributions. (c) The uneven contact heights tend to cause the object to tip, which in turn causes the contact pressure distribution to concentrate at the nearest endpoints of the contact line segments.

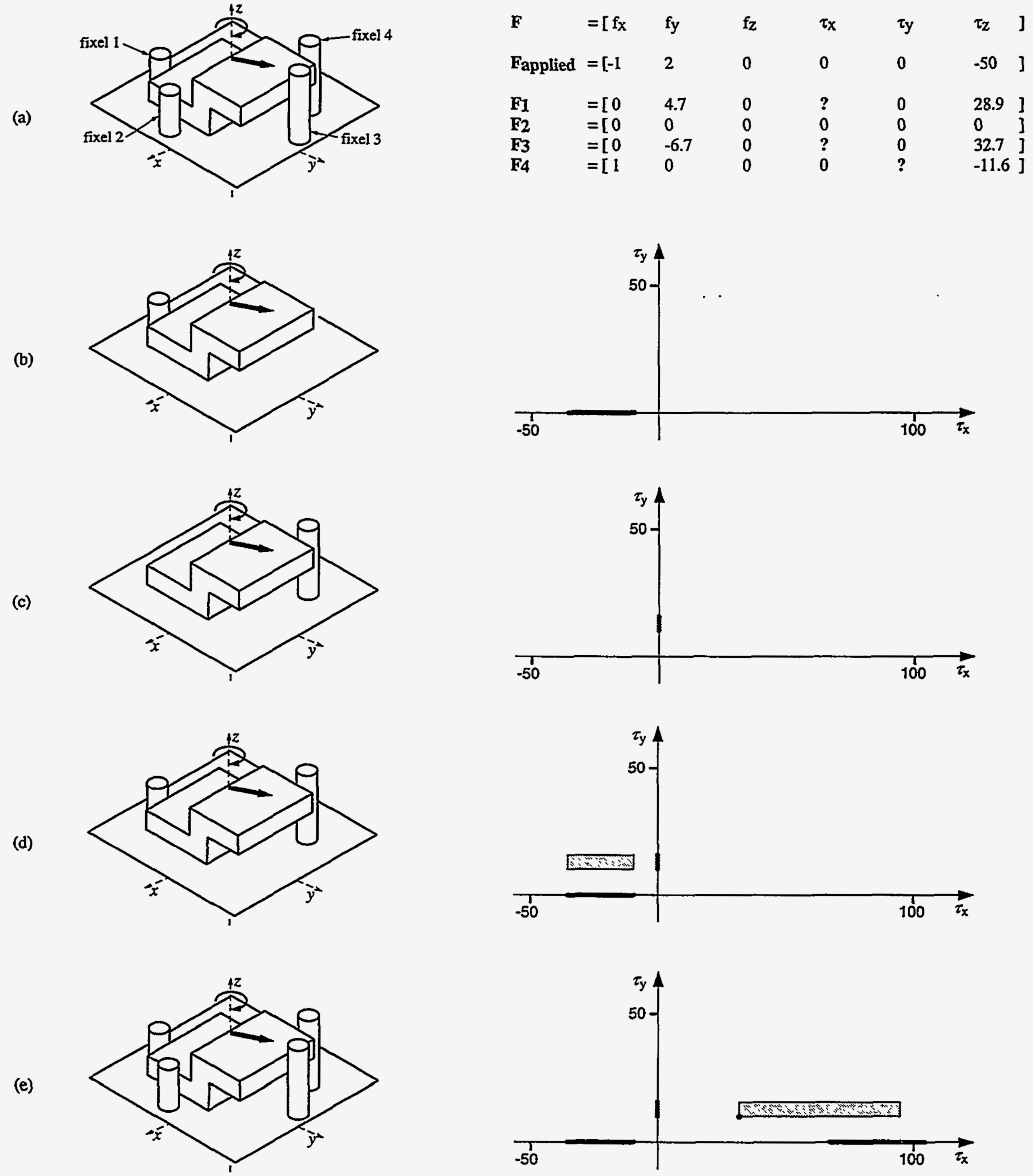
We can generalize this approach to handle the case of four fixels by plotting the locus of possible induced moments in the (τ_x, τ_y) space. Given known reaction force magnitudes, each fixel contact gives rise to a line segment of possible induced moments, corresponding to the range of moments exerted as the contact height varies from $z_{\min}$ to $z_{\max}$; see Figure 47(a)-(c). The set of possible moments for two contacts corresponds to the Minkowski sum of the two corresponding line segments, as shown in Figure 47(d). The resulting parallelogram in the (τ_x, τ_y) space delineates the set of all total moments that are possible given choices of $z_1 \in \mathcal{Z}_1$ and $z_2 \in \mathcal{Z}_2$. Continuing this process will produce the set of total moments that may be exerted by all four contacts acting in concert, for all choices of specific contact heights. If $\mathcal{L}_i$ is the line segment in (τ_x, τ_y) space corresponding to contact i, then the set of all possible total moments is $\mathcal{M} = \mathcal{L}_1 \oplus \mathcal{L}_2 \oplus \mathcal{L}_3 \oplus \mathcal{L}_4$, where $\oplus$ denotes Minkowski sum. See Figure 47(e).

The set $\mathcal{M}$ is always convex; we can compute this set by calculating all possible vertices of $\mathcal{M}$ and then forming the convex hull. These vertices may be generated by the following equations:

$$\begin{aligned} \tau_{x_j} &= -F_1 s_1 z_{\pm 1} - F_2 s_2 z_{\pm 2} - F_3 s_3 z_{\pm 3} - F_4 s_4 z_{\pm 4} \\ \tau_{y_j} &= \quad F_1 c_1 z_{\pm 1} + F_2 c_2 z_{\pm 2} + F_3 c_3 z_{\pm 3} + F_4 c_4 z_{\pm 4} \end{aligned}$$

where $z_{\pm i}$ indicates either $z_{\min_i}$ or $z_{\max_i}$. This produces a set of $2^4 = 16$ candidate points; the convex hull typically has fewer vertices.

Constructing $\mathcal{M}$ allows us to immediately determine the minimum induced tipping moment. If $\mathcal{M}$ includes the origin, then the minimum induced moment is zero and equations (17–19) and (20–22) are decoupled. Otherwise, the minimum moment corresponds to the point in $\mathcal{M}$ that is closest to the origin. It is possible to determine the corresponding z_i values through appropriate bookkeeping, but this is unnecessary, because the coordinates of the closest point directly provide the desired $\tau_{x_{\text{side}}}$ and $\tau_{y_{\text{side}}}$ values.



F	=[fx	fy	fz	τx	τy	τz]
Fapplied	=[-1	2	0	0	0	-50]
F1	=[0	4.7	0	?	0	28.9]
F2	=[0	0	0	0	0	0]
F3	=[0	-6.7	0	?	0	32.7]
F4	=[1	0	0	0	?	-11.6]

Figure 47: Constructing the set of possible total moments about the x and y axes that may be induced by all four side fixels acting in concert. (a) Workpiece in contact with four fixels. The applied force and moment and the fixel forces and moments are listed in the table. The contact force at fixel 2 is zero. (b) Moments about the x and y axes that may be induced by fixel 1. (c) Possible moments about the x and y axes for fixel 4. (d) Possible combined moments from fixel 1 and fixel 4 are indicated by the shaded box. (e) Possible moments induced by fixels 1-4 are indicated by the shaded box; the minimum value is indicated by a dot.

B.4 Force Analysis Without Top Clamps

Equations (20–22) assume that the z-constraint is a tripod of bilateral constraints formed by support/top-clamp pairs. If top clamps are absent, then there may be more than three support contact points, each of which provides only unilateral support constraint. In this case the basic calculation is: Given a z-constraint and an applied force $\mathbf{F}_A$, what downward force w_{CG} is required at the center of mass to prevent tipping? This result is then used to calculate $w_{\text{excess}} = w_{CG} - w_{\mathcal{W}}$, which is used as the basis of the quality function $\mathtt{Q}_{z_F}$ as described in Section 6.2.

To perform this calculation, we first verify that $(x_{\text{com}}, y_{\text{com}})$ is contained within the support convex hull. If this is not the case, then the part will tip in the absence of any applied force, and the fixture design is discarded. Likewise, we check to see whether $\mathbf{F}_A$ is a pure vertical force passing through $(x_{\text{com}}, y_{\text{com}})$; if so, then $w_{CG} = F_{z_A}$.

If neither of these conditions applies, we proceed to calculate the required CG force. To understand this computation, consider the case where $\mathbf{F}_A$ is a pure downward force exerted at some point $\mathbf{p}_A$. In this case F_{z_A} is negative. As shown in Figure 48, this point must lie in a sector formed by the center of mass and some pair of support vertices; the line connecting these vertices forms the fulcrum of a lever that balances F_{z_A} and w_{CG}. To produce force balance, w_{CG} must be positive if $\mathbf{p}_A$ is outside the support region, and negative if $\mathbf{p}_A$ is inside the support region. Regardless of whether $\mathbf{p}_A$ is inside or outside the support region, the contact forces at the support vertices are both positive.

We exploit this property to identify the CG force required to prevent tipping. This is accomplished by forming the convex hull of the support points, and representing the resulting edges as $(\mathbf{p}_i, \mathbf{p}_j)$ point pairs. Then for each point pair, we apply equations (20–22) to the pseudo-tripod formed by $\mathbf{p}_i$, $\mathbf{p}_j$, and $(x_{\text{com}}, y_{\text{com}})$. If the resulting contact forces at $\mathbf{p}_i$ and $\mathbf{p}_j$ are both positive, then $(\mathbf{p}_i, \mathbf{p}_j)$ is the critical support edge and the force at $(x_{\text{com}}, y_{\text{com}})$ is $-w_{CG}$; otherwise we proceed to the next point pair. Note that this method will recover w_{CG} even if $\mathbf{F}_A$ is not a pure downward force.

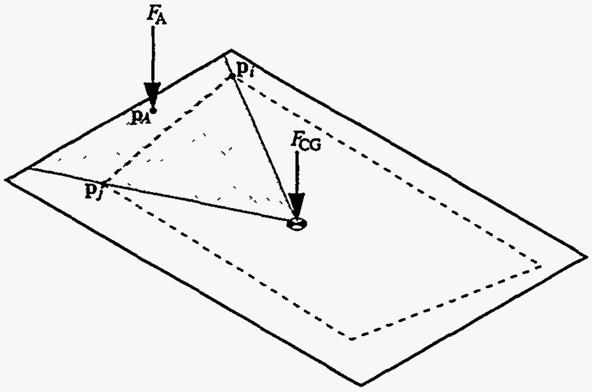


Figure 48: Determining the CG force required to prevent tipping. The support region is indicated by the dashed line.

Appendix C: Form Closure Numerical Issues

Form closure is a condition that results when the constraints imposed by a collection of contact normals combine to span the space of all possible motions. This test for total constraint of a rigid body has been widely used in the robotics literature [37, 52, 42, 43].

Unfortunately, computational tests for form closure are vulnerable to numerical brittleness problems. This discussion will focus on form closure in the xy-plane where four properly-placed contact normals are sufficient to hold an object in 2-d form closure.

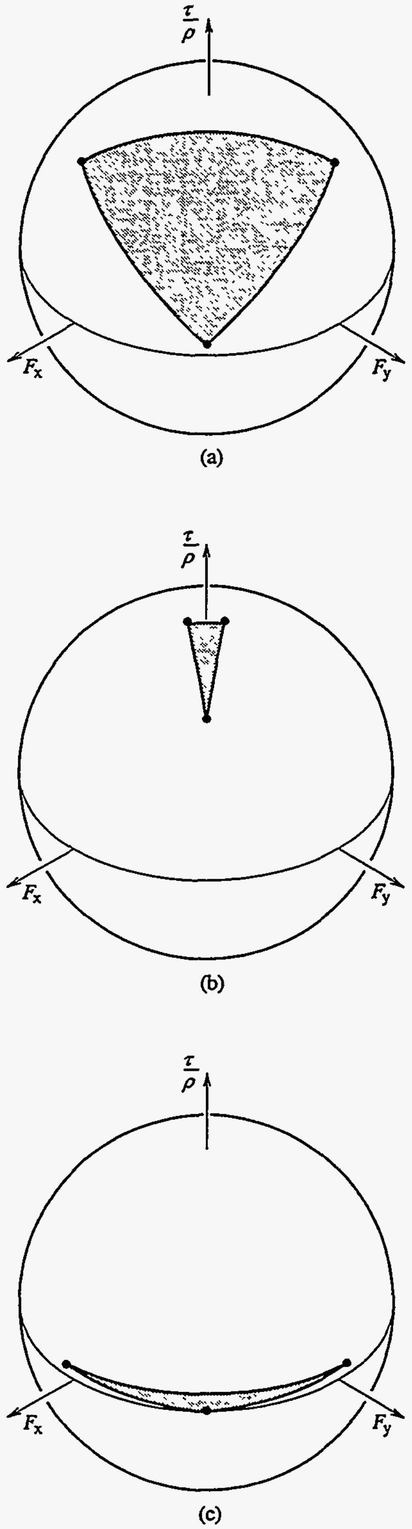
Part (a) of Figure 49 shows a plot of forces and torques of three contact normals on the surface of a unit sphere. These three contact normals plus a fourth contact normal on the opposite side of the sphere span the sphere and provide form closure for an object. For this discussion, these three points are grouped into a triangle and the fourth point on the opposite side of the sphere is omitted for clarity.

Part (b) of Figure 49 shows a plot of the same forces and torques shown in (a), except the origin for calculating the torques has been moved a large distance from its previous position, increasing the magnitude of the torques by a factor of ten. The object is still in form closure, because form closure is invariant with respect to the placement of the origin.

But calculating the test for form closure may be affected by this change in origin, due to numerical round-off errors. For example, prior to determining whether a set of four points span the force sphere, the algorithm will delete redundant points. It may decide that the two points near the upper pole are so close together that they are really only one point. Thus the slender triangle collapses into a line segment on the sphere, and only three total points remain. Three points can never span the force sphere, and so the test fails.

Part (c) of Figure 49 shows a plot of same forces and torques plotted in (a), except the value of ρ has been increased by a factor of ten, reducing the value of $\frac{\tau}{\rho}$. In this case the middle point near the equator is likely to be deleted because it is near the line connecting the other two points; deleting it reduces the triangle to a line segment, and causes the form closure test to fail.

Our implementation includes a form closure test that is designed to mitigate these problems. First, the algorithm selects a value of ρ that avoids the problem of Figure 49(c). Second, the algorithm moves the origin to the center of the bounding box formed by the contact points, thus avoiding the problem of Figure 49(b).



(a)

(b)

(c)

Figure 49: Locator forces projected onto the force sphere. (a) Base case. (b) Torques increased by a factor of 10. (c) ρ increased by a factor of 10.

Appendix D: Push-Into-Place Loading Analysis

This appendix addresses the problem of determining whether the workpiece will slide into place when pushed by the side clamp during a fixture loading operation. As mentioned above, this operation may fail for several reasons, which we repeat here:

- *Jamming.* The workpiece jams due to friction before reaching the desired final position. This condition persists even if the side-clamp force is increased.
- *Clamp stall.* The workpiece stops moving before reaching the desired final position, because the side clamp is not powerful enough to overcome friction. This condition may be eliminated by increasing the side clamp force.
- *Tipping.* The workpiece tips in an out-of-plane rotation before reaching the desired final position.
- *Escape.* The workpiece slips away from the desired final position. The clamp successfully closes, but the workpiece has "slipped out" of the desired grasp.

The development of an exact test to detect these failures seems possible. One reasonable approach would be to extend the results presented in [8]. The basic approach is to construct a "capture region" of initial workpiece positions that are guaranteed to reliably slip into the desired final position when the side clamp closes. If the initial workpiece position $(x, y, \theta)_L$ lies within this region, then the loading operation will reliably succeed; otherwise it might fail.

[8] describes two methods for constructing this capture region, both of which are performed in the task configuration space. The workpiece is a rigid object in three dimensions and the fixture side clamp has one degree of freedom, so a seven-dimensional configuration space $(x, y, z, \phi, \psi, \theta, x_{\text{travel}})$ may be used to describe this system. If we treat all out-of-plane rotations as illegal situations that should be avoided, then a four-dimensional configuration space $(x, y, \theta, x_{\text{travel}})$ may be used for the analysis. Both methods presented in [8] construct a *strong backprojection*, which is a region of initial configurations that are guaranteed to reach the task goal — in this case to reach the desired final loaded position. The desired capture region is then constructed by taking the slice of the strong backprojection where $x_{\text{travel}} = x_{\text{retracted}}$, which produces a volume in (x, y, θ) space.

The first step in constructing the strong backprojection is to construct the configuration-space obstacle of the workpiece/fixture combination. [8] presents an implemented algorithm for performing this construction in the three-dimensional (x, y, θ) space, and explains how to extend this algorithm to the required four-dimensional $(x, y, \theta, x_{\text{travel}})$ space. Other approaches to constructing configuration-space obstacles may be found in [2, 65, 64].

The configuration-space obstacle represents the set of all configurations where the workpiece and fixture are in contact. For some of these configurations the workpiece may tip because the support triangle does not contain the workpiece CG. These configurations may be marked as illegal, using extensions of the method developed by Caine [14]. Caine's algorithm delineates regions on the c-space obstacle surface where tipping occurs because the support is inadequate.

Once the set of legal contact configurations has been constructed, [8] presents two methods for constructing the strong backprojection. Both methods explicitly consider task uncertainty and the numerous motion ambiguities that characterize the behavior of real physical systems that are modeled imperfectly.

The first method uses energy arguments to identify a region in the configuration space for which *(i)* an initial configuration cannot escape the region, and *(ii)* jamming is never possible, except for goal configurations. If the system is dissipative, then these conditions ensure that an initial configuration in the region will reach the goal in finite time. This analysis may only be applied when there exists some configuration variable that is bounded by the initial state; the plunger position x_{travel} satisfies this requirement, since the plunger closes monotonically under suitable assumptions. Thus this method may be applied to construct a capture region for a fixture loading task. The resulting capture region will avoid the jamming and escape failure modes listed above, and also avoid tipping due to inadequate CG support. However, failure may still result from clamp stall or tipping due to out-of-plane contact forces. Thus, the energy-based method is only adequate in cases where these failures are known to be impossible.

The second method uses trajectory integration to identify a region in configuration space where every possible task trajectory is guaranteed to reach the goal. This method begins by constructing a set of configurations S delineating the stable subset of the goal, which in this case is a single configuration. S is then progressively expanded to include adjacent configurations whose cone of possible infinitesimal trajectories strictly enters $\mathcal{S}$. Thus at all times $\mathcal{S}$ contains a set of initial configurations known to reliably achieve the goal. Expansion of $\mathcal{S}$ continues as long as desired or until no further expansion is possible. The resulting $\mathcal{S}$ is a strong backprojection of the goal, from which the desired capture region may be extracted. If the infinitesimal trajectory analysis considers the effect of workpiece tipping and limited plunger force, then the resulting capture region will delineate initial positions that avoid all four of the above failure modes. The required infinitesimal trajectory analysis is beyond the current state of the art in analyzing the motion of sliding bodies experiencing multiple contacts; see [26, 27, 73, 41, 8] for relevant papers. Thus while the integration-based approach would produce a correct solution to the problem, its development would be a significant research result in its own right.

Unfortunately, we did not have time available to pursue these exact approaches in our current project. Nonetheless, some test was needed — our experiments showed that many fixtures returned by the algorithm failed to load due to jamming or clamp-stall conditions. Initially we conjectured that fixtures with high quality scores might not be susceptible to these problems, but this conjecture turned out to be false: In fact, we observed several instances of globally optimal designs that failed to load due to jamming. We did not encounter tipping or escape failures in the examples that we studied, although it is easy to construct cases that suffer from these problems.

Thus we decided to implement an approximate test to discard fixtures that clearly suffer from jamming or clamp stall problems during loading. Fixtures are only discarded if a witness configuration is found that exhibits a jamming or clamp stall condition. This approximate test appears to be quite effective — loading failures were frequently observed before implementing the test, but have not been observed since. The remainder of this section describes this approximate test.

D.1 Generating Sample Workpiece Positions

The basic structure of our test is as follows: We begin by identifying the side clamp plunger position x_{travel_F} when the workpiece is in the final loaded pose and clamped in place. We also find the plunger position x_{travel_L} where the plunger first contacts the workpiece in the loading pose. As the side clamp closes, the plunger position varies from x_{travel_L} to x_{travel_F}. We thus discretize the interval $[x_{\text{travel}_L}, x_{\text{travel}_F}]$ and generate a series of intermediate workpiece configurations at each of the resulting discrete points. Each configuration is checked for jamming or clamp stall problems.

Given the interval $[x_{\text{travel}_L}, x_{\text{travel}_F}]$, we produce ten discrete plunger positions

$$\{x_{\text{travel}_L}, x_{\text{travel}_2}, x_{\text{travel}_3}, \ldots, x_{\text{travel}_9}, x_{\text{travel}_F}\}.$$

For each position x_{travel_i}, the algorithm uses the pose function to generate positions where the workpiece contacts the side clamp tip and two of the three side locators. Completing this calculation for all two-combinations of the three locators produces a total of 30 configurations, each of which is characterized by three simultaneous contacts.

Before passing these configurations on to the jamming and clamp stall analysis procedures, the algorithm first checks each configuration to see if the side clamps and locators still contact their corresponding workpiece features. In some cases, one or more fixture elements has "fallen off" its corresponding contact feature. If this occurs, we discard the fixture because the analysis cannot be completed. A more sophisticated test might attempt to determine whether the workpiece will move in a manner that re-establishes the desired contacts; for example, the above strong backprojection methods include this.

D.2 Testing for Jamming

Each of the 30 configurations is then tested for jamming. Jamming occurs when the frictional contact forces can resist clamp plunger forces of arbitrary magnitude. Thus jamming cannot be overcome by simply selecting a more powerful side clamp.

In this test we treat the jamming condition as a purely planar problem. Under this view, jamming occurs if and only if the six forces bounding the three contact friction cones positively span the $(F_x, F_y, \frac{\tau}{\rho})$ space of planar forces. The "if" part can be seen by observing that if the friction cones positively span the force/torque space, then there exists a selection of contact forces on the strict interior of each friction cone that sum to zero; this set of forces may be multiplied by an arbitrary scalar while still maintaining the equilibrium condition. Thus for any clamp plunger force, some scalar can be found that yields a corresponding set of contact forces which are in static equilibrium. The "only if" part can be seen by observing that if an arbitrary plunger force can be resisted, then there must exist some force in the plunger tip friction cone whose negation lies within the convex combination of the locator friction cones, expressed in the $(F_x, F_y, \frac{\tau}{\rho})$ space. Geometric analysis shows that this implies that the locator and plunger friction cones combine to positively span the space. See [11, 8, 9] for supporting analysis.

Thus we can determine whether jamming is possible by determining whether the rays bounding the contact friction cones positively span the force/torque space. This is accomplished using the same methods that we use to verify form closure, but with six input forces instead of four.

Note that this test determines whether jamming is *possible;* it does not assure that jamming *will* occur. In some cases it is also possible for motion to occur; these ambiguous cases are of no concern to us, because we seek to discard all fixture designs where jamming is a possibility. After discarding all such fixtures, the fixtures that remain are known to avoid the jamming failure condition, at least for the sample configurations tested. See [22, 11, 73] for further discussion of motion ambiguities under Coulomb friction.

D.3 Testing for Clamp Stall

If all 30 sample configurations do not jam, then each configuration is tested for clamp stall failures. The side clamp will stall when the force required to move the workpiece exceeds the clamp's available closing force. Because internal clamp friction reduces this closing force when the plunger is moving, our implementation accepts a separate value F_{close} for the available closing force, and uses this value for clamp stall analysis. Typically $F_{\text{close}} < F_{\text{plunger}}$.

To determine whether clamp stall will occur, the test first estimates the motion that results from successfully pushing the workpiece toward the final configuration, and then calculates the frictional resistance to that motion. This resistance is then converted to a required side clamp closing force; if this force exceeds F_{close}, then clamp stall can occur.

Estimating the workpiece motion is complicated by the fact that several motion contact modes are possible. One possible mode would be for all three contacts to persist, with sliding occurring at each contact. Another possible mode would be for sticking to occur at one contact, sliding to occur at another, with a loss of contact at the third. A third possiblity would be for two contacts to persist, both sliding. We resolve this problem by hypothesizing all possible motion modes, calculating the resulting contact reaction forces for each, and discarding those that are inconsistent with the hypothesized mode. All hypothesized modes that pass this filter represent possible workpiece motions; if any of these require a closing force greater than F_{close}, then the fixture is discarded because clamp stall is possible. If no consistent mode is found, then we discard the fixture because we are unable to complete the analysis. See [45, 4, 22] for a discussion of motion inconsistencies in the presence of Coulomb friction.

For each hypothesized contact mode, we construct the resulting workpiece motion, calculate the contact forces required to produce that motion, and the check for consistency. If the forces are consistent with the hypothesized mode, we then convert the side clamp contact force into a required plunger closing force, and compare this result to F_{close}. If the required force exceeds F_{close}, then the fixture is discarded; otherwise, we proceed to the next hypothesized mode. The remainder of this section will explain this calculation in detail.

To begin, we make two assumptions. First, we treat all contacts as if they are at the same height, neglecting the effects of out-of-plane torques. Second, we assume that the support positions are known; if the z-constraint support positions have not yet been determined, then a canonical support placement is assumed that will generate substantial resistance to rotational motion.

Given a workpiece configuration contacting two locators and the side clamp plunger, there are several possible motion modes that may occur as the side clamp closes. These modes are characterized by the motion mode at each contact: sticking, sliding, or breaking. Thus there are at most $3^3 = 27$ possible modes to consider. Because we are interested in detecting clamp stall conditions, we do not need to consider modes where the side clamp breaks contact. Further, we do not need to consider modes where the side clamp is the only remaining contact; if $F_{\text{close}} > \mu_{\text{support}} \cdot w_{\mathcal{W}}$, then these modes will never cause the clamp to stall. This condition may be checked during pre-processing.

Thus we only need to consider modes that maintain at least two contacts, one of which is the side clamp. These modes may be characterized by the following classes:

- All contacts persist.
  - (i) In general, this is only possible if all three contacts slide.
  - (ii) In certain degenerate cases, it is possible for one contact to stick. This situation is best handled as a special version of case (iii) below.
- One contact breaks.
  - (iii) The remaining two contacts slide.
  - (iv) One remaining contact slides, the other sticks.

The following sections will explain each of these cases in detail. It is easiest to explain the solutions to these cases in the following order: (i), (iv), (ii), (iii).

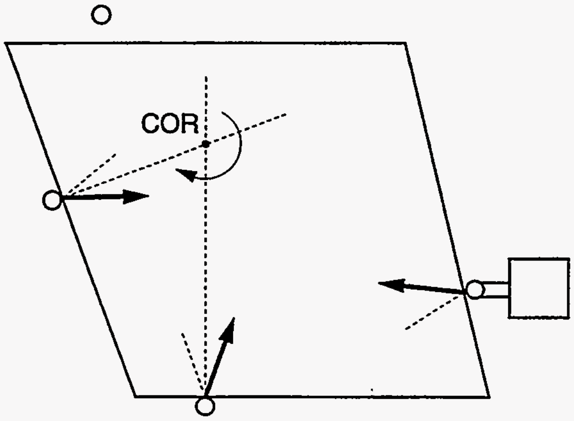


Figure 50: Constructing the instantaneous center of rotation (COR) for the case where all three contacts slide. The active edge of each friction cone is shown. If the contact normals are parallel, then the COR is a point at infinity.

Case (i): All Three Contacts Slide

Let's first consider the case where all three contacts slide. Since the workpiece remains in contact with both locators, its motion can be constructed directly using the method developed by Reuleaux [63]. This motion is a rotation about a point determined by intersecting the locator contact normals, as shown in Figure 50. The rotation direction may be determined from the motion direction of the side clamp plunger.

Identifying the rotation center and the rotation direction allows us to construct the frictional reaction force at each support pad, and also the force direction at each contact. The force at each support is found by identifying the workpiece motion direction at each support point, and then constructing a force at each point in the opposite direction of magnitude $\mu_{\text{support}} w_i$, where w_i is the portion of the workpiece weight $w_{\mathcal{W}}$ born by support i. This construction is shown in Figure 51. The force direction at the location and clamp contacts is found by determining the workpiece sliding direction at each contact, and constructing the opposite ray of the contact friction cone.

We now know the force at each support point, and the force direction at each side contact. What is unknown is the force magnitude at each side contact. The minimum clamp force consistent with workpiece motion will exactly balance the support friction's resistance to motion; thus we determine the sliding contact force magnitudes by formulating three equations expressing force balance in x, y, and θ, and solving for the three missing contact force magnitudes. If the resulting scalars are all non-negative, then the motion mode is consistent; otherwise the mode is impossible. If the motion mode is consistent, then the side-clamp plunger force is determined from the side-clamp contact force, and the result is compared to F_{close}.

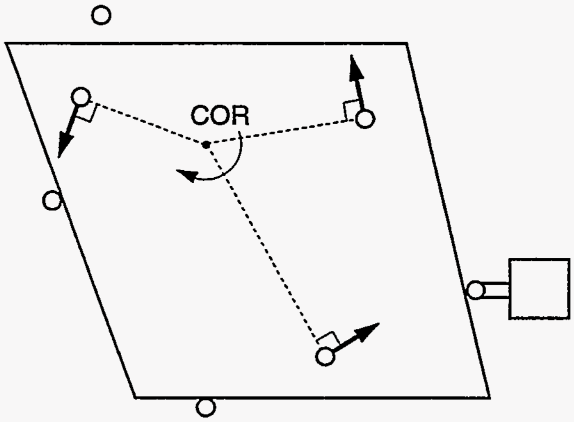


Figure 51: Constructing the force at each support point. The support friction force opposes the direction of local motion.

Case (iv): One Contact Breaks, One Slides, One Sticks

Next we consider the case where one contact breaks, once contact slides, and one contact sticks. There are four possibilities for this mode, because the side clamp can only stick or slide, and for each of these choices there are two choices of the remaining locator.

Our approach to each of these cases is fundamentally the same: First we calculate the workpiece motion, then calculate the forces corresponding to that motion, and then check for consistency. We again construct a rotation center and direction, as shown in Figure 52. Unlike the case with three sliding contacts, we must check for kinematic motion consistency by determining whether the resulting motion violates the third contact instead of breaking it. If so, then the motion mode is rejected as impossible. If the motion causes the third contact to persist without breaking or violating it, then we have case (ii), explained below.

Once the center of rotation and rotation direction are known, we calculate the forces at the support points and the force direction at the sliding contact. Now the remaining unknowns are the magnitude of the force at the sliding contact, and the two components of force at the sticking contact. We again find these values by solving three equations in three unknowns. This result

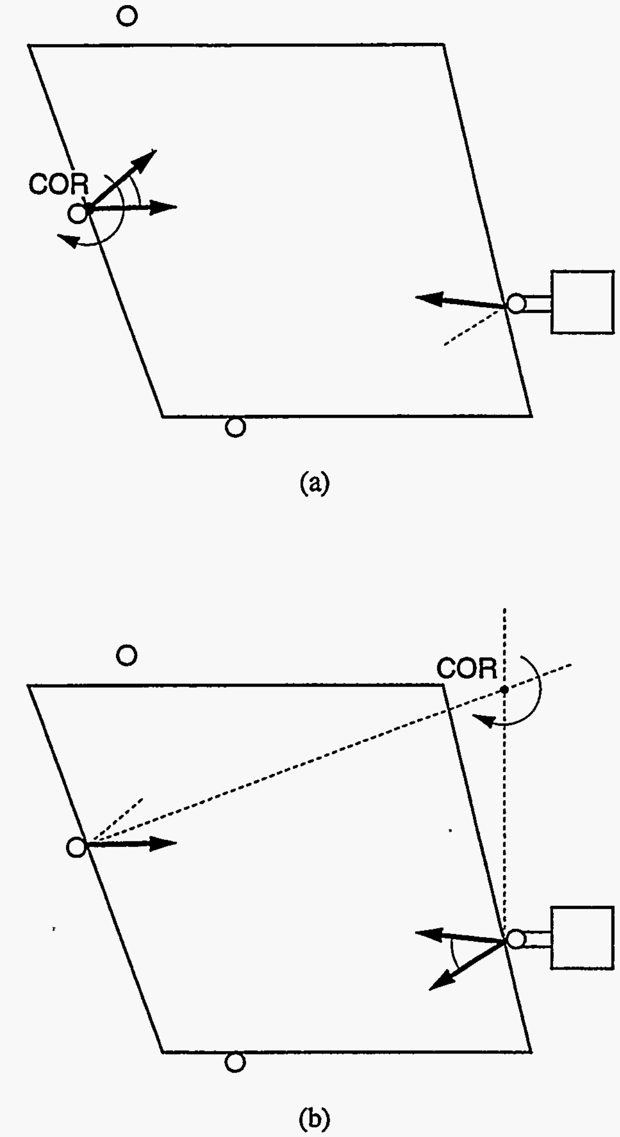


Figure 52: Constructing the instantaneous center of rotation for the case where one contact sticks, one slides, and one breaks. (a) If the sticking contact is a locator, then the rotation center coincides with the contact point, and the rotation direction can be inferred from the clamp closing direction. In this case this motion is kinematically infeasible because it violates the second locator. (b) If the sticking contact is the clamp, then the rotation center is constructed by intersecting the sliding contact normal and the normal to the plunger travel direction; the rotation direction is again inferred from the clamp closing direction. Both of these constructions neglect the locator radius, treating the sticking contact as an ideal point rather than a rolling contact on a circle. This approximation may be corrected by suitable geometric analysis which is beyond the scope of this paper.

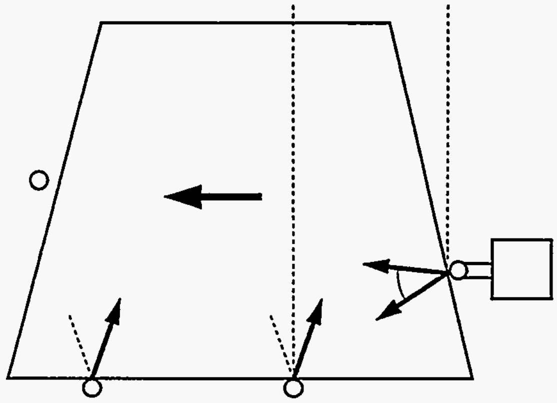


Figure 53: A degenerate case where one contact sticks, but both remaining contacts persist. The lower-right locator and clamp yield a rotation center at infinity, corresponding to a motion where the workpiece translates left. Under this motion, the lower-left locator contact persists.

is then checked for consistency: The sliding-force magnitude must be non-negative, and the sticking force must lie within the contact friction cone. If both of these conditions are met, then the motion is consistent and the side-clamp plunger force is calculated and compared to F_{close}.

Case (ii): One Contact Sticks, Two Slide

In certain degenerate geometric situations, it is possible for one contact to stick while the other two slide. Figure 53 shows one example. These situations are identified when the kinematic feasibility check of case (iv) determines that the third contact persists rather than breaks.

When this occurs, the center of rotation and rotation direction are already known from the calculation of case (iv). The only modification required is to include the contact force of the third sliding contact before checking for consistency. This produces an underdetermined problem, in that we now have three equations in four unknowns. However, consistency is indicated by the existence of any valid solution to these equations, which may be found by either linear programming or the geometric analysis methods presented in [11].

If the motion mode is found to be consistent, then we can determine if clamp stall can occur by finding the maximum plunger force consistent with the motion, and comparing this to F_{close}. Linear programming methods may be used to find this maximum plunger force; manual direct solution methods may also be formulated for this simple linear system.

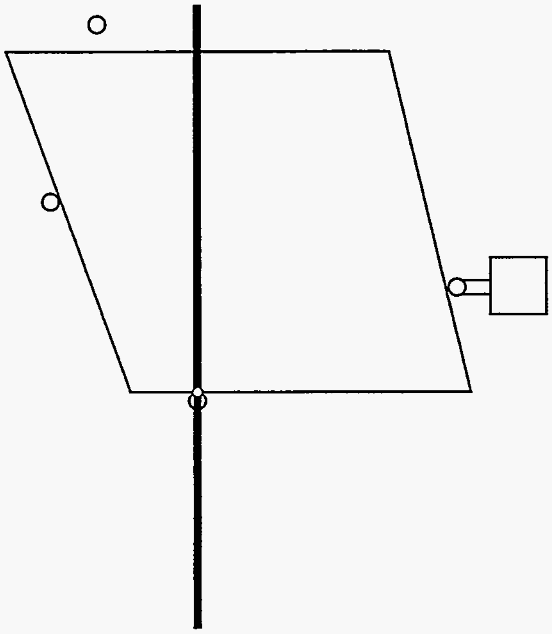


Figure 54: The locus of rotation centers consistent with sliding at one locator and the side clamp. If the plunger moves to follow the workpiece, all rotation centers on the locator contact normal are consistent with sliding at both contacts. The single exception is the locator contact point; this rotation center is consistent with sticking at the locator and sliding at the side clamp.

Case (iii): One Contact Breaks, Two Slide

This case is the most difficult to solve, because the workpiece motion is not uniquely determined by kinematic constraints. Since the plunger moves, the simple construction of Figure 50 cannot be used. Instead, any rotation center on the locator contact normal is kinematically consistent with some plunger motion, as shown in Figure 54. Various rotation centers chosen along this line correspond to different relative rates of rotation vs. plunger motion, all of which are kinematically consistent.

To determine whether or not any of these motions are possible, we must consider the contact and support forces. As in the previous cases, a motion is possible if the appropriate contact forces can exactly match the corresponding support forces. Unlike the previous cases, we do not have a single specific motion which we can use to evaluate this question. Instead, we have a range of possible motions; this motion mode is consistent if there exists at least one specific motion in the range that obeys the consistency condition. Thus our problem is to determine whether or not such a motion exists, and if so, its identity.

We can solve this problem by observing that both the motions and forces can be described as one-dimensional sets embedded in a two-dimensional space, and that a consistent motion will correspond to an intersection between these sets. The following paragraphs will make this abstract observation concrete. We will draw heavily on the force and motion representations of [63], [11] and [26, 27].

We begin by observing that there are actually four subcases: Each of the two sliding contacts may slide left or right. For the sake of brevity, we will only discuss the case where both contacts slide left; the other cases are similar.

Figure 55 shows four constraints on the set of possible rotation centers that are consistent with left sliding at both contacts, the clamp plunger closing, and breaking the second locator contact. Intersecting these constraints yields the set of rotation centers consistent with all four conditions (Figure 56).

The left/left sliding mode also constrains the set of possible contact forces. This is shown in Figure 57. The active friction cone edges are shown, as well as the point A where these force rays intersect. Regardless of the magnitude of each contact force, the total force resulting from the combined effect of both contacts must pass through point A, in a direction inside the cone shown by the arc.

For the motion mode to be possible at the critical clamp closing force, there must exist some rotation center within the range shown in Figure 56 which yields a total support friction force which passes through A in a direction opposing the cone of possible force directions. If such a rotation center exists, then the motion, support force, and contact forces are all consistent.

We can search for such a motion by performing binary search along the rotation center locus of Figure 56, seeking a rotation center which yields a support friction force that has zero moment about point A. A study of the convexity and dual force/motion properties of the support friction limit surface developed in [26, 27] leads us to conjecture that this moment will vary monotonically over the locus. If so, then binary search would find a suitable rotation center, if one exists. Unfortunately, developing a rigorous proof of this conjecture is beyond the scope of this paper, so we will proceed under the assumption that it is true.

If so, then binary search will yield the rotation center that produces a total support friction force that passes through A, if one exists. If a valid rotation center is found, then we check to see if the direction of the total support friction force lies within the negation of the

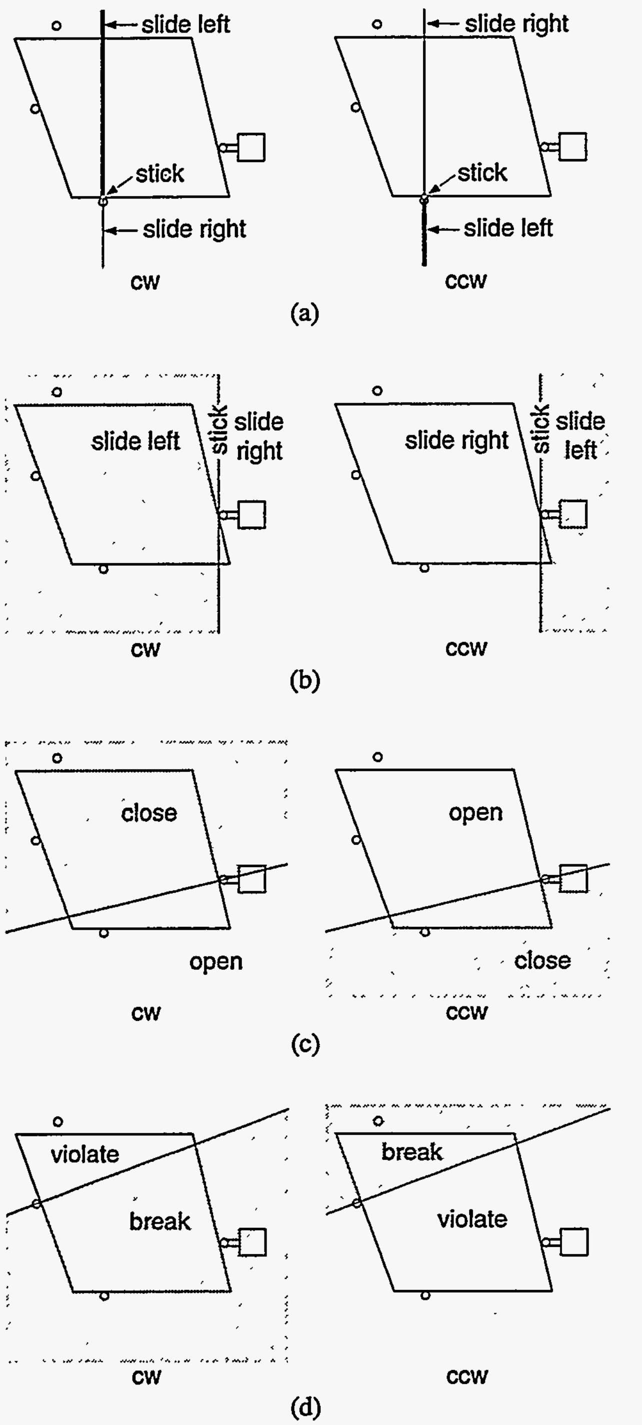


Figure 55: Constraints on the set of rotation centers consistent with left sliding at both the side clamp and the locator. (a) Rotation centers consistent with left sliding at the locator. (b) Rotation centers consistent with left sliding at the clamp. (c) Rotation centers consistent with the clamp closing. (d) Rotation centers consistent with breaking the second locator contact.

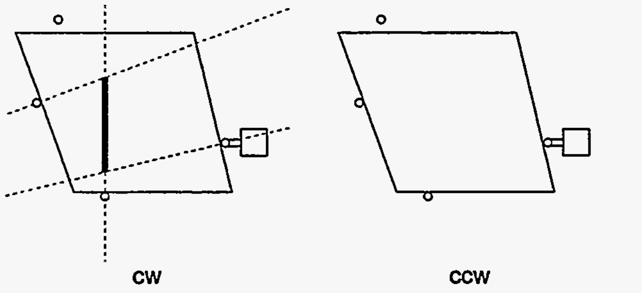


Figure 56: Rotation centers consistent with all four conditions shown in Figure 55.

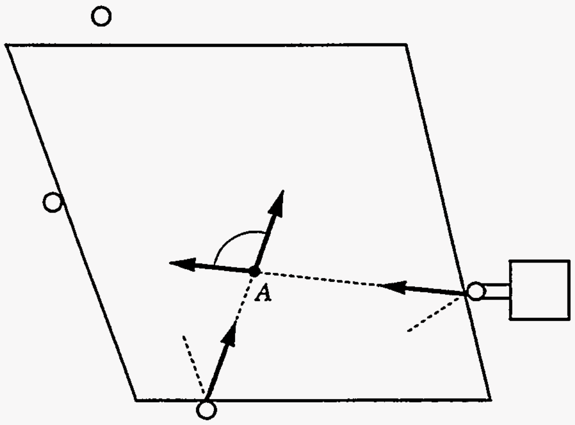


Figure 57: Possible contact forces for the left/left sliding mode.

cone of possible total contact force directions. If so, then the motion is consistent.

If a consistent motion is found, then the required closing force is calculated as before, and compared to F_{close}. If the contact force exceeds F_{close}, then clamp stall is possible.

This completes our description of the clamp stall test. Unfortunately, project time limits prevented us from fully implementing this test. As of this writing, our code does not employ the rotation center construction methods described above, but instead uses a simpler approximation based on nearby feasible workpiece positions. Further, our implementation does not address case (iii) at all. Nonetheless, our implementation was able to detect and reject fixtures that were shown to fail due to clamp stall in physical experiments. Similarly, fixtures accepted by our code did not exhibit clamp stall problems in physical tests.

References

[1] H. Asada and A. B. By. Kinematic analysis of workpart fixturing for flexible assembly with automatically reconfigurable fixtures. *IEEE Journal of Robotics and Automation*, RA-1(2):86–94, June 1985.

[2] F. Avnaim and J. D. Boissonnat. Polygon placement under translation and rotation. *Informatique Théorique et Applications*, 23(1):5–28, 1989.

[3] B. S. Baker, S. Fortune, and S. Grosse. Stable prehension with a multi-fingered hand. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 570–575, March 1985.

[4] D. Baraff. Determining frictional inconsistency for rigid bodies is np-complete. Technical Report TR 90-1112, Cornell University Department of Computer Science, April 1990.

[5] D. Baraff, R. Mattikalli, and P. Khosla. Minimal fixturing of frictionless assemblies: Complexity and algorithms. Technical Report CMU-RI-TR-94-08, Carnegie Mellon University Robotics Institute, April 1994.

[6] J. J. Bausch and K. Youcef-Toumi. Kinematic methods for automated fixture reconfiguration planning. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 1396–1401, May 1990.

[7] W. Boyes, editor. *Handbook of Jig and Fixture Design, 2nd Edition.* Society of Manufacturing Engineers, 1989.

[8] R. C. Brost. *Analysis and Planning of Planar Manipulation Tasks.* PhD thesis, Carnegie Mellon University School of Computer Science, January 1991. Available as Technical Report CMU-CS-91-149.

[9] R. C. Brost. Computing the possible rest configurations of two interacting polygons. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 686–693, April 1991.

[10] R. C. Brost and K. Y. Goldberg. A complete algorithm for designing planar fixtures using modular components. *IEEE Transactions on Robotics and Automation*, 12(1):31–46, February 1996.

[11] R. C. Brost and M. T. Mason. Graphical analysis of planar rigid-body dynamics with multiple frictional contacts. In H. Miura and S. Arimoto, editors, *Robotics Research: The Fifth International Symposium*, pages 293–300, Cambridge, Massachusetts, 1990. MIT Press.

[12] R. C. Brost and R. R. Peters. Automatic design of 3-d fixtures and assembly pallets. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 495–502, April 1996.

[13] R. G. Brown and R. C. Brost. A 3-d modular gripper design tool. Technical Report SAND 97-0063, Sandia National Laboratories, January 1997.

[14] M. E. Caine. The design of shape from motion constraints. Technical Report 1425, MIT Artificial Intelligence Laboratory, September 1993.

[15] V. Chandrasekharan, S. G. Kapoor, and R. E. DeVor. A mechanistic approach to predicting the cutting forces in drilling: With application to fiber-reinforced composite materials. *Machining of Advanced Composites*, ASME MD-Volume 45/PED-Volume 66:33–51, 1993.

[16] C. Chang. Computer-assisted fixture planning for machining processes. *Manufacturing Review*, 5(1):15–28, March 1992.

[17] Y-C. Chou, V. Chandru, and M. M. Barash. A mathematical approach to automatic configuration of machining fixtures: Analysis and synthesis. *ASME Journal of Engineering for Industry*, 111:299–306, November 1989.

[18] Paul H. Cohen. Automated fixture design. In *National Science Foundation Design and Manufacturing Systems Conference*, pages 405–413, January 1991. (Published by the Society of Manufacturing Engineers).

[19] Y. Descotte and J. C. Latombe. Gari: A problem solver that plans how to machine mechanical parts. In *Proceedings, 7th International Joint Conference on Artificial Intelligence*, pages 766–772, August 1981.

[20] Y. Descotte and J. C. Latombe. Making compromises among antagonist constraints in a planner. *Artificial Intelligence*, 27(2):183–217, November 1985.

[21] P. J. Englert. *Principles for Part Setup and Workholding in Automated Manufacturing.* PhD thesis, Carnegie Mellon University Department of Mechanical Engineering, December 1987.

[22] M. Erdmann. On a representation of friction in configuration space. *International Journal of Robotics Research*, 13(3):240–271, June 1994.

[23] A. O. Farahat, P. F. Stiller, and J. C. Trinkle. On the geometry of contact formation cells for systems of polygons. *IEEE Transactions on Robotics and Automation*, 11(4):522–536, August 1995.

[24] C. Ferrari and J. Canny. Planning optimal grasps. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 2290–2295, May 1992.

[25] M. V. Gandhi and B. S. Thompson. Automated design of modular fixtures for flexible manufacturing systems. *Journal of Manufacturing Systems*, 5(4):243–252, 1986.

[26] S. Goyal, A. Ruina, and J. Papadopolous. Planar sliding with dry friction, part 1: Limit surface and moment function. *Wear*, (143):307–330, 1991.

[27] S. Goyal, A. Ruina, and J. Papadopolous. Planar sliding with dry friction, part 2: Dynamics of motion. *Wear*, (143):331–352, 1991.

[28] R. A. Grupen, T. C. Henderson, and I. D. McCammon. A survey of general purpose manipulation. *International Journal of Robotics Research*, 8(1):38–62, February 1989.

[29] F. Gu. *Prediction of Cutting Forces and Surface Errors in Face Milling with Generalized Cutter and Workpiece Geometry.* PhD thesis, University of Illinois at Urbana-Champaign, 1993.

[30] C. C. Hayes. *Machining Planning: A Model of an Expert Level Planning Process.* PhD thesis, Carnegie Mellon University Robotics Institute, December 1990.

[31] C. C. Hayes and P. K. Wright. Automating process planning: Using feature interactions to guide search. *Journal of Manufacturing Systems*, 8(1):1–15, 1989.

[32] F. B. Hazen and P. Wright. Workholding automation: Innovations in analysis, design, and planning. *Manufacturing Review*, 3(4), December 1990.

[33] M. J. Hockenberger and E. C. De Meter. Preliminary experimentation and modeling of workpiece displacement during machining. *Transactions of the NAMRI/SME*, 21:351–357, 1993.

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

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

[36] L. S. King, M. Hart, and D. Wong. Geometry and physics of automatic fixture planning for assembly. In *Proceedings of the International Conference on Computer Integrated Manufacturing*, pages 525–529, 1991.

[37] K. Lakshminarayana. The mechanics of form closure. Technical Report 78-DET-32, ASME, 1978.

[38] J. D. Lee and L. S. Haynes. Finite-element analysis of flexible fixturing systems. In *Proceedings ASME Japan-USA Symposium on Flexible Automation*, pages 579–584, 1986.

[39] S. H. Lee and M. R. Cutkosky. Fixture planning with friction. *Transactions of the Journal of Engineering for Industry*, 113:320–327, August 1991.

[40] T. Lozano-Pérez, J. L. Jones, E. Mazer, P. A. O'Donnell, W. E. L. Grimson, P. Tournassoud, and A. Lanusse. Handey: A task-level robot system. In R. Bolles and B. Roth, editors, *Robotics Research: The Fourth International Symposium*, pages 29–36, Cambridge, Massachusetts, 1988. MIT Press.

[41] K. M. Lynch and M. T. Mason. Stable pushing: Mechanics, controllability, and planning. *International Journal of Robotics Research*, 15(6):533–556, December 1996.

[42] M. Mani. *Automatic Design of Workholding Fixtures Using Kinematic Constraint Synthesis.* PhD thesis, Northwestern University Department of Mechanical Engineering, August 1988.

[43] X. Markenscoff, L. Ni, and C. H. Papadimitriou. The geometry of grasping. *International Journal of Robotics Research*, 9(1):61–74, February 1990.

[44] X. Markenscoff and C. H. Papadimitriou. Optimum grip of a polygon. *International Journal of Robotics Research*, 8(2):17–29, April 1989.

[45] M. T. Mason and Y. Wang. On the inconsistency of rigid-body frictional planar mechanics. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 524–528, April 1988.

[46] R. J. Menassa and W. R. DeVries. Locating point synthesis in fixture design. *Annals of the CIRP*, 38(1):165–169, 1989.

[47] R. J. Menassa and W. R. DeVries. Optimization methods applied to selecting support positions in fixture design. *Transactions of the ASME: Journal of Engineering for Industry*, 113(4), 1991.

[48] E. C. De Meter. Restraint analysis of machining fixtures. *Transactions of the NAMRI/SME*, 20:339–346, 1992.

[49] E. C. De Meter. Restraint analysis of assembly work carriers. *Robotics and Computer Integrated Manufacturing*, 10(4):257–265, 1993.

[50] E. C. De Meter. Restraint analysis of fixtures which rely on surface contact. *ASME Journal of Engineering for Industry*, 116(2):207–215, May 1994.

[51] B. Mishra. Workholding: Analysis and planning. In *Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 53–57, November 1991.

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

[53] V. Nguyen. Constructing force-closure grasps. *International Journal of Robotics Research*, 7(3):3–16, June 1988.

[54] E. Oberg, F. D. Jones, and H. L. Horton. *Machinery's Handbook, 23rd Edition.* Industrial Press, New York, NY, 1988.

[55] E. N. Ohwovoriole. Kinematics and friction in grasping by robotic hands. *Transactions of the ASME*, 109, 1987.

[56] M. Overmars, A. Rao, O. Schwarzkopf, and C. Wentnik. Immobilizing polygons against a wall. In *ACM Symposium on Computational Geometry*, pages 29–38, 1995.

[57] J. Pertin-Troccaz. Grasping: A state of the art. In O. Khatib, J. J. Craig, and T. Lozano-Pérez, editors, *The Robotics Review I*, pages 71–98. MIT Press, Cambridge, Massachusetts, 1989.

[58] J. Ponce. On planning immobilizing fixtures for three dimensional parts. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 509–514, April 1996.

[59] J. Ponce and B. Faverjon. On computing three-finger force-closure grasps of polygonal objects. Technical Report UIUC-BI-AI-RCV-93-05, Beckman Institute, University of Illinois, 1993.

[60] J. Ponce, S. Sullivan, J. Boissonnat, and J. Merlet. On characterizing and computing three- and four-finger force-closure grasps of polyhedral objects. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 821–827, May 1993.

[61] K. Quinter. Qu-co modular fixturing system. Qu-Co, Inc., Union, Ohio, 1993.

[62] M. R. Rearick, J. S. Hu, and S. M. Wu. Optimal fixture design for deformable sheet metal workpieces. *Transactions of the NAMRI/SME*, 21:407–412, 1993.

[63] F. Reuleaux. *The Kinematics of Machinery*. Macmillan, 1876. Republished by Dover, New York, 1963.

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

[65] E. Sacks and C. Bajaj. Sliced configuration spaces for curved planar bodies. Technical Report 95-078, Purdue University, 1995.

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

[67] H. Sakurai. Automatic setup planning and fixture design for machining. *Journal of Manufacturing Systems*, 11(1):30–37, 1992.

[68] J. K. Salisbury and J. J. Craig. Articulated hands: Force control and kinematic issues. *International Journal of Robotics Research*, 1(1):4–17, Spring 1982.

[69] B. Shirinzadeh. Issues in the design of the reconfigurable fixture modules for robotic assembly. *Journal of Manufacturing Systems*, 12(1):1–14, 1993.

[70] Society of Manufacturing Engineering. *Tool and Manufacturing Engineering Handbook*, 1988.

[71] P. Somoff. Uber gebiete von schraubengeschwindigkeiten eines festen korpers bei verschiedener zal von stutzflachen. *Zeitschrift fur Mathematik und Physik*, 45:245–306, 1900.

[72] J. C. Trinkle. A quantitative test for form closure grasps. In *Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1670–1677, July 1992.

[73] J. C. Trinkle and D. C. Zeng. Prediction of the quasistatic planar motion of a contacted rigid body. *IEEE Transactions on Robotics and Automation*, 11(2):229–246, April 1995.

[74] R. Wagner, March 1996. Personal Communication.

[75] R. Wagner, Y. Zhuang, and K. Goldberg. Fixturing faceted parts with seven modular struts. In *Proceedings, IEEE International Symposium on Assembly and Task Planning*, pages 133–139, August 1995.

[76] A. S. Wallack and J. F. Canny. Planning for modular and hybrid fixtures. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 520–527, May 1994.

[77] A. S. Wallack and J. F. Canny. Modular fixture design for generalized polyhedra. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 830–837, April 1996.

[78] C. Wentink, A. F. van der Stappen, and M. Overmars. Fixture planning. In Jean-Paul Laumond and Mark Overmars, editors, *The Algorithmic Foundations of Robotics.* A. K. Peters, Boston, MA. In preparation.

[79] J. D. Wolter and J. C. Trinkle. Automatic selection of fixture points for frictionless assemblies. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 528–534, May 1994.

[80] K. Yu and K. Goldberg. Fixture loading with sensor-based motion plans. In *Proceedings, IEEE International Symposium on Assembly and Task Planning*, pages 362–367, August 1995.

[81] Y. Zhuang, K. Y. Goldberg, and Y. Wong. On the existence of modular fixtures. In *Proceedings, IEEE International Conference on Robotics and Automation*, May 1994. Also available as USC Techreport IRIS-93-314.

Distribution:

MS 9018	Central Technical FIles, 8940-2
0899	Technical Library, 4414
0619	Review & Approval Desk, 12630 For DOE/OSTI