Accurate Insertion Strategies Using Simple Optical Sensors

Eric Paulos*
paulos@cs.berkeley.edu

John Canny*
jfc@cs.berkeley.edu

Department of Electrical Engineering and Computer Science
University of California
Berkeley, CA 94720

Abstract

Peg-in-hole insertion is not only a longstanding problem in robotics but the most common automated mechanical assembly task [10]. In this paper we present a high precision, self-calibrating peg-in-hole insertion strategy using several very simple, inexpensive, and accurate optical sensors. The self-calibrating feature allows us to achieve successful dead-reckoning insertions with tolerances of 25 microns without any accurate initial position information for the robot, pegs, or holes. The program we implemented works for any cylindrical peg, and the sensing steps do not depend on the peg diameter, which the program does not know. The key to the strategy is the use of a fixed sensor to localize both a mobile sensor and the peg, while the mobile sensor localizes the hole. Our strategy is extremely fast, localizing pegs as they are en route to their insertion location without pausing. The result is that insertion times are dominated by the transport time between pick and place operations.

1 Introduction

We describe a method for performing accurate insertion operations using simple optical sensors. A key to the method is the use of one sensor to compute the position of the other, which eliminates the need for prior set-up and calibration. The sensors are simple optical beam sensors, which respond to the presence or absence of an object along the beam line. These sensors require relative motion between object and sensor, and we claim that this is a feature. During assembly, parts are always being transported between various positions in the workspace. As a part passes over the cross beam sensor, the part is accurately localized. All of this is done without the need to pause over the sensor. Other applications of these sensors in the broader setting of RISC robotics are presented in work by Canny and Goldberg [4]. In addition we have implemented this method and demonstrated it to be highly reliable for any locally accurate (≈ 5 cm) robot.

Typical calibration methods in use today involve special robot actions and costly calculations, resulting in a reduction of the throughput for the entire assembly system. We correct position errors in the assembly workcell dynamically during normal assembly operations. The advantage of this technique is that the varying position uncertainties for the manipulators and other parts attached or grasped by the robots do not prevent us from performing successful insertions. This makes our approach well suited for an automated manufacturing system (AMS) where it is common to find part differences and changing robot position errors over successive assemblies.

1.1 Previous Insertion Work

The inherent uncertainty in robots has led several others to propose methods for performing robust assembly operations. Most notable is the work by Lozano-Pérez, Mason, and Taylor [9]. Their approach, referred to as LMT, explicitly models sensor and control uncertainty and defines guaranteed strategies under these models. Erdmann [6] expanded LMT and made several of the major steps computational. Later work by Donald [5] added error recovery to LMT. A general algorithm for full LMT was presented by Canny [3]. But planning with this kind of uncertainty is very expensive in the worst case [2], and LMT has not been applied beyond two dimensions.

Another active area of research uses mechanical compliance to perform insertions under uncertainty.

*Financial support provided by National Science Foundation Presidential Young Investigator Award #IRI-8958577 and National Science Foundation Grand #IRI-9114446.

Compliance based on the Remote Center of Compliance (RCC) was introduced by Whitney [17] and later developed by Peshkin [12]. RCCs are somewhat specialized though, and active compliance has been studied as a way to deal with a variety of assembly scenarios, in particular by Khatib [8].

Our work complements this previous work by extending the number of assembly tasks that can be performed without the need for compliance. Our dead-reckoning strategy is as fast as RCC insertion, and as flexible as active strategies. In addition our approach avoids unnecessary contact between parts, eliminating the possibility of binding or scratching while using existing inexpensive optical sensors already found in industry.

1.2 Previous Optical Sensor Work

For several decades, optical beam sensors have been in use throughout industry for performing quality checks, part alignments, and other tasks. However, the construction of powerful general purpose assembly tools using only a small number of simple optical sensors coupled with intelligent algorithms has not been exploited.

Optical sensing has been used for calibration since 1988 by Everett and Ives [7]. Their approach uses optical sensors and special precision spheres to perform calibrations. They mount these spheres onto a special calibration block, calibrate the robot's position, and then remove the block. Their off-line calibration scheme is reported to be repeatable to 58 microns, while our online insertion system is repeatable to 25 microns.

Work in 1993 by Prenninger *et al.* [13] uses a laser optical system for calibration. They use a laser/mirror system to triangulate the position of the end-effector of a robot. Their system is quite complex, involving a laser, a mirror mounted on the end-effector, a high precision universal joint, and a CCD vision system. They report an accuracy of approximately 50 microns.

Beam sensors can be used to do recognition as well as localization, as described by Wallack, Canny, and Manocha [16]. In the future we plan to combine that work with the present, to produce a general precision insertion algorithm for arbitrary parts.

2 RISC Robotics

RISC robotics [4] (Reduced Intricacy in Sensing and Control) is an attempt to fuse automation and robotics technologies. The RISC acronym, borrowed from computer architecture, suggests the parallels between the two technologies. RISC robotics performs complex manufacturing operations by composing simple elements.

RISC robotics can be applied to many areas of manufacturing. For example, RISC grasping uses simple two and three fingered grippers with traditional fixturing devices such as clamps and vices [15]. RISC sensing employs simple but precise sensor elements that can be combined to form complete systems for localizing and recognizing arbitrary objects from a library [16, 14].

RISC robotics systems inherently consist of few degrees of freedom and low-dimensional sensor spaces. This results in algorithms for manipulation and sensing that are simple, highly accurate, and very fast.

3 Self-Calibrating Strategy

Our strategy localizes all of the elements necessary for the assembly operation using a fixed optical cross beam sensor. The cross beam optical sensor, shown in Figure 1, consists of three oppositely mounted transmitters-receivers sensor pairs. These optical sensors respond to beam breakage, hence object presence or absence along each of the beam lines. This cross beam sensor is mounted on a fixed platform in the workspace and used to localize both the pegs and the optical hole sensor.

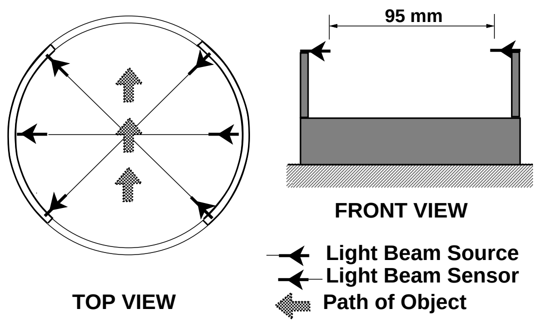

Figure 1: Optical cross beam sensor

The optical hole sensor is shown in Figure 2. This sensor differs from the cross beam sensor by having both its transmitting and receiving sensor elements in the same lens. By mounting this sensor vertically at the end of one of the robot modules we can detect position information for various holes in the workspace by detecting the presence or absence of a reflected optical signal.

Although we discuss our system using two separately mounted elements, a gripper end-effector and a

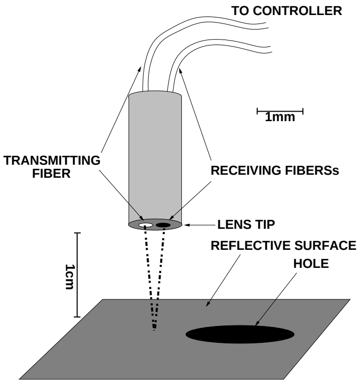

Figure 2: Optical hole sensor

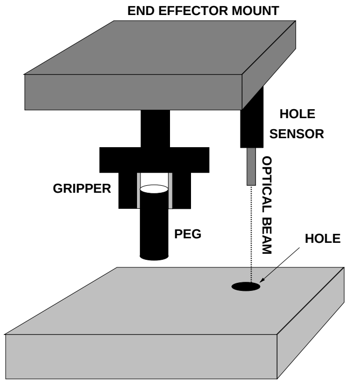

Figure 3: An instrumented sensor/gripper unit

hole sensor end-effector, we can implement the same system on a single end-effector by mounting the hole sensor on the gripper end-effector as shown in Figure 3. Using this setup, we simply measure the relative offset of the hole sensor from the peg in the gripper using the cross beam sensor. Using that relative information, we simply remove the offset after we localize the hole using the attached offset hole sensor and then perform the insertion. This means that any robot with locally accurate position control, such as an ADEPT, Fuji, or Panasonic, can perform our insertion strategy. Although we use the RobotWorld robot in our implementation, no special purpose robot is necessary.

We will first outline our peg-in-hole insertion strategy followed by detailed descriptions of each localization strategy in Sections 3.2 and 3.3.

3.1 Strategy Outline

The first part of our strategy is to localize the hole sensor, used for hole localization, with the cross beam sensor. As the tip of the hole sensor is passed through the cross beam sensor, beam breakages are sensed and position information recorded. With the now localized hole sensor, we move to the assumed hole location and, using the hole sensor, localize the hole using the pattern from Figure 6. After this operation, we know the relative position of the hole with respect to the cross beam sensor. In the final step we pass the peg through the cross beam sensor, localizing it. Then, using the relative position information of the hole determined by the hole sensor, we perform a dead reckoning peg-in-hole insertion.

If we use an instrumented sensor/gripper end-effector as shown in Figure 3, we simply pass the whole unit through the cross beam sensor, recording the position information of the peg and hole sensor as well as the relative distance between them. After we localize the hole using the attached offset hole sensor, we simply remove the measured relative offset and then perform the insertion.

In a typical assembly workcell we would place the cross beam sensor between the parts bin and assembly station. As parts were picked from the bin and moved to be assembled, they would pass through the sensor and be localized instantaneously for free.

All of the tools necessary to perform the actions of our algorithm are quick, simple, and highly accurate. In the following sub-sections we present solutions to the above localization problems using these tools.

Since the pegs and hole sensor in our environment are cylinders, and the holes circular, we develop efficient localization strategies for these cases. Our approach can be expanded to include non-cylindrical shapes using only a few small additions. In the following sub-sections we develop algorithms for localizing both holes and peg-like (cylindrical) objects.

3.2 Localize Cylinders

In this section we want to determine the position and shape of an opaque cylindrical object relative to some reference frame.

3.2.1 Using an Optical Sensor

As we pass an object through a single optical cross beam sensor perpendicular to its motion, we can record the position of the robot when the beam is broken and later reconnected. Figure 4 depicts this process. We record the y-position of the object when the beam is first broken ($T = 0$) and again when it reconnects ($T = 1$). The diameter of the object can be determined directly from the difference in the posi-

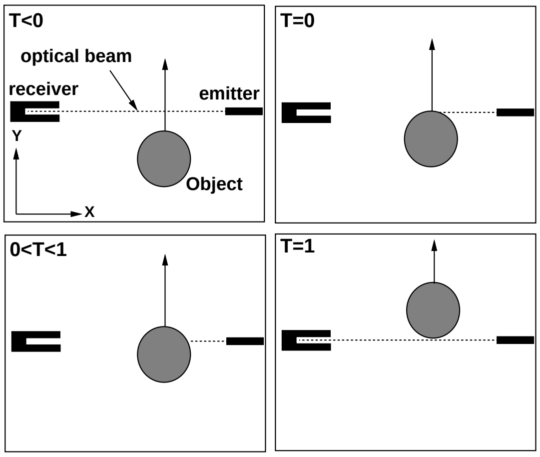

Figure 4: Use of a single optical sensor

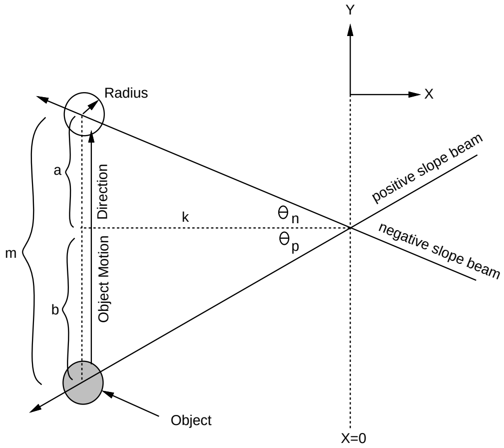

Figure 5: Optical cross beam sensor arrangement

tions of the object at the two points where the horizontal beam changed states. Assuming that the function $y_{pos}(t)$ returns the y-position of the robot at time t we get the following.

$$P_{diam} = \|y_{pos}(1) - y_{pos}(0)\| \tag{1}$$

Using the same data we can also extract the y-positional center of the object relative to the optical beam using Equation 2.

$$P_y = y_{pos}(0) + \frac{P_{diam}}{2} \tag{2}$$

We are left to extract the x-positional information for the center of the object. We could use a similar approach using another optical sensor mounted perpendicular the x-axis while moving the object in the x-direction. However, we want to extract all of the position information in a single pass and avoid special robot motions dedicated specifically to sensing and calibration. We solve this by using two noncollinear beams with known angles to obtain complete positional information and shape.

3.2.2 Using a Cross Beam Sensor

In the general case we have the situation shown in Figure 5. The beam geometry is fixed meaning that initial measurements give us θ_n and θ_p as knowns. The measurement of these beam angles is the only calibration step necessary in our approach and is performed by accurately measuring a few points along each beam with a calibration tool. This calibration is necessary only once after the sensor array has been installed in the workspace.

We place the origin of an xy frame at the intersection of the two beams. The origin of this frame with respect to the center of the object is given by Equation 3. We make a measurement of m as the object passes through the beams by recording position information at beam breakages. Since the x-position remains constant throughout the motion, the value for $x_{pos}(t)$ can be taken for any t. We only require this straight-line motion over a length of a few centimeters, allowing our technique to remain applicable on a wide variety of different robots.

$$\begin{aligned} P_x &= x_{pos}(t) + k \\ &= x_{pos}(t) + \frac{m}{\tan(\theta_n) + \tan(\theta_p)} \end{aligned} \tag{3}$$

Since the entire denominator of Equation 3 is known, it can be computed off-line. Therefore, we only need to perform a single addition and division operation to extract the x-position information.

With respect to the same frame we can also obtain y-positional information using Equation 4, where $y_{pos}(\alpha)$ is the y-position of the end-effector when the object is in the shaded position as shown in Figure 5.

$$\begin{aligned} P_y &= y_{pos}(\alpha) + b \\ &= y_{pos}(\alpha) + k\tan(\theta_p) \\ &= y_{pos}(\alpha) + \frac{m\tan(\theta_p)}{\tan(\theta_n) + \tan(\theta_p)} \end{aligned} \tag{4}$$

An important results from this approach is that none of the optical beams are required to be aligned coincident to an axis of the robot. We are now ready to describe the algorithm for localizing a cylinder.

Algorithm 1 (Localize Cylinder) *Assume that we are given a set of optical beams arranged as depicted in Figure 1 and a method to read position information for the robot end-effector as we move the object to be localized through such an arrangement.*

1. *If the object is not mounted on the robot end-effector, grasp the part.*
2. *Pass the portion of the object to be localize through the cross beam sensor arrangement. For insertion tasks the tip of the object is localized.*
3. *Record end-effector position information when any beam changes state along with the current beam state.*
4. *Use Equation 1, 3, and 4 to calculate* P_{diam}, P_x, *and* P_y *for the object.*

In our implementation we use three cross beam sensors. The middle beam is used to determine the object's diameter and y-position directly and the other diagonal beams for the x-position.

Errors in z and θ (about z) caused by robot end-effector inaccuracies and/or misaligned parts in grippers, can be corrected by using two sets of cross beams. Arranged at different heights, these two sets of cross beams provide the necessary information to detect such errors and correct them.

3.3 Localize Holes

Starting with a rough initial guess of the location of a hole, we will use an optical hole sensor to localize a hole to within 25 microns. As in the previous section, we will first look at the geometry of the the hole localization problem and describe an efficient algorithm for solving it.

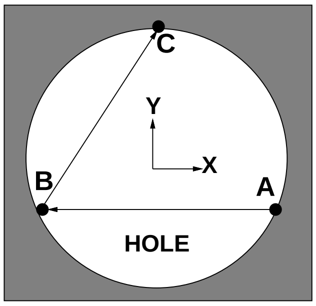

Figure 6: Strategy for collection of hole edge points

Our approach centers around recording the location of three points on the edge of the hole using a reflective optical sensor. We must be able to initially locate some point along the edge of the hole. If our initial guess does not place us in the hole, we perform a spiral or grid based search strategy until an edge of the hole is detected. From this first point we use the simple movement strategy depicted in Figure 6 to collect two other edge points which we use to calculate the diameter and center of the hole.

$$H_{diam} = \frac{a\,b\,c}{2\sqrt{s(s-a)(s-b)(s-c)}} \tag{5}$$

$$H_x = B_x + \frac{a}{2} \tag{6}$$

$$H_y = B_y + \sqrt{b^2 - \frac{a}{2}} \tag{7}$$

$$a = ||A-B|| \qquad b = ||B-C||$$
$$c = ||C-A|| \qquad s = \tfrac{1}{2}(a+b+c)$$

Earlier we claimed that an advantage of our system was that it did not involve costly motions that were specific to sensing and calibration. However, hole localization requires several specialized motions. These motions do not violate our claim since their cost is negligible. For a hole with diameter d, we will move through a distance less than $2d$. Typical high precision assembly tasks contain holes with diameters of a centimeter or less. This means that the entire hole sniffing operation can be completed on typical robots in well under a second.

Algorithm 2 (Localize Hole) *Assume that we are given a reflective optical hole sensor as depicted in Figure 2 and a method to read end-effector position information for the robot as we move this sensor over a hole.*

1. *Move the robot with the sensor mounted on its end-effector to the assumed hole location.*
2. *If the sensor is located over the hole goto step 3. Otherwise the sensor does not point into the hole (i.e. the initial guess for the location of the hole is worse than expected). Therefore, perform a grid or spiral search with the reflective sensor to locate an edge of the hole.*
3. *Perform the robot motions as shown in Figure 6 and record the position of the end-effector at points A, B, and C.*
4. *Use Equation 5, 6, and 7 to compute* H_{diam}, H_x, *and* H_y *for the object.*

Simple checks can be performed on the hole edge points that we collect. One such check would be to determine if all of the points actually lie along the edge of a circle. This check would allow us to rule out erroneous edges values that may have been detected from a feature of the assembly which is not a hole edge.

4 Experimental Setup

4.1 Sensor Hardware

Several optical-fiber photo-electric sensors were used for all of the sensing described in this paper. The basic sensor consists of a simple 23 gram controller that generates an optical light beam and, using a photo-electric diode, outputs a digital detection signal with a response time of 50μs. Attached to this controller is a fiber-optic cable with one of several different lens types at its end. These sensors can detect objects with diameters as small as 15–30 microns.

Our implementation uses both an optical cross beam sensor as shown in Figure 1 and an optical reflective hole sensor as shown in Figure 2. The digital outputs from the various optical sensors are connected to a VME bus by a digital I/O board. Once on the VME bus, the robot controller can then read the sensor values.

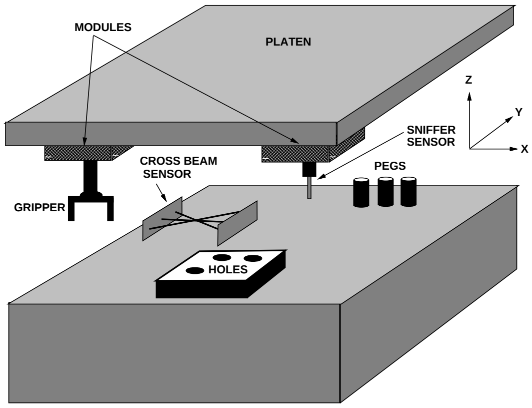

Figure 7: The RobotWorld system

4.2 Robot Hardware and Control

We implemented our peg-in-hole system at U.C. Berkeley on the RobotWorld robot [1] depicted in Figure 7. This robot consists of several Sawyer motor modules magnetically attached to an 0.8×1.3 meter rectangular steel platen and separated from the platen by an air bearing, thus allowing planar motion with low friction. This Sawyer drive system provides xy-planar positional accuracy on the order of 25–50 microns. Standard DC servo motors attached to the Sawyer drive base of each module provide motion in the z and θ (about z) directions. Although there are four degrees of freedom, we are only concerned with the motion directions using the xy-planar steppers and the z DC motors. These robot systems, originally developed by Yaskawa and well suited for vertical assembly tasks, are in use throughout industry.

The basic RobotWorld configuration consists of two independent modules. A three-fingered gripper is attached to one of the modules for the pick and place operations while the other module has the optical hole sensor mounted on it for hole localization. In this system, compliance, a highly desired property of RCC peg-in-hole systems, is traded for high positional accuracy. All robot control and sensing are performed by Talisker [11], a multi-threaded, real-time robot control system developed by Ed Nicolson *et al.* at U.C. Berkeley. Talisker currently executes on a 68040 microprocessor with a VME backplane.

5 Results

We performed repeated insertion tests of our peg-in-hole strategy using pegs and holes of various diameters both with and without chamfers. It is also important to note that both the peg and hole locations were displaced by arbitrary distances up to approximately one centimeter over successive executions to demonstrate the dynamic self-calibration feature. The results from these tests are shown in Table 1. We conclude that our system is highly robust to varying position errors over multiple executions.

Insertion Tolerance	Chamfered Hole or Peg?	Success Rate
25μm	No	99%
26μm	Yes	99%
26μm	No	99%

Table 1: Insertion test results

In addition, our peg-in-hole system is fast. Insertion times are always dominated by pick, place, and transport operations. The hole localization can be completed in under a second while the object localization and actual insertion are instantaneous. Subsequent insertions are performed even quicker since we do not require re-localization of the hole sensor before those holes are localized. We also exploited the parallelism of the system by localizing holes in parallel with localizing pegs.

6 Conclusion

In this paper we successfully demonstrated a use of RISC sensors for performing high accuracy peg-in-hole insertions. More importantly our strategy is dynamically self-calibrating, making it robust to position errors in the pegs, holes, and robot end-effector over successive assembly operations. We have eliminated the need for any costly off-line calibrations, instead localizing objects to within 25 microns during normal assembly operations such as part transport. Finally, our system is cost-effective, using inexpensive sensors already found in manufacturing applications.

7 Acknowledgments

The authors would like to thank Ed Nicolson for designing the controller for RobotWorld and for all of his assistance with the real-time interface to the RobotWorld environment, Aaron Wallack for calibrating and upgrading the cross-beam, optical sensor, and Dr. Richard Murray for constructing the original prototype interface to RobotWorld. In addition we would like to thank all of the people mentioned above and the anonymous referees for their numerous thoughtful criticisms, suggestions, and comments.

References

[1] Description of a Robot Workspace Based on a Linear Stepper Motor. *AT&T Technical Journal*, 67(2):6–11, 1967.

[2] J. Canny and J. Reif. New lower bound techniques for robot motion planning problems. In *IEEE Conference on Foundations of Computer Science*, pages 39–48, 1987.

[3] John Canny. On computability of fine motion plans. In *IEEE Conference on Robotics and Automation*, pages 177–183, 1989.

[4] John Canny and Kenneth Goldberg. RISC for industrial robotics: Recent results and open problems. In *IEEE International Conference on Robotics and Automation*, May 1994.

[5] B. Donald. A geometric approach to error detection and recovery for robot motion planning with uncertainty. *Artificial Intelligence*, 37:223–271, 1988.

[6] Michael Erdmann. Using backprojections for fine motion planning with uncertainty. *International Journal of Robotics Research*, pages 19–45, 1986.

[7] Louis Everett and Thomas Ives. A sensor for measurments in the calibration of production robots. In *IEEE International Conference on Robotics and Automation*, pages 174–179, 1993.

[8] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, RA-3(1):43–53, February 1987.

[9] T. Lozano-Pérez, M.T. Mason, and R.H. Taylor. Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research*, pages 3–24, 1984.

[10] James Nevins and Daniel Whitney. Computer-controlled assembly. *Scientific America*, 238(2):62–74, February 1978.

[11] Ed Nicolson. Standardizing I/O for mechatronic systems (SIOMS) using real-time unix device drivers. In *IEEE International Conference on Robotics and Automation*, May 1994.

[12] Michael A. Peshkin. Programmed compliance for error corrective assembly. In *IEEE Transations on Robotics and Automation*, pages 473–482, 1990.

[13] J.P. Prenninger, M. Vincze, and H. Gander. Contactless position and orientation measurments of robot end-effectors. In *IEEE International Conference on Robotics and Automation*, pages 180–185, 1993.

[14] Aaron Wallack and John Canny. A geometric matching algorithm for beam scanning. In *SPIE International Society for Optical Engineering*, volume 2060 Vision Geometry II, pages 143–160, October 1993.

[15] Aaron Wallack and John Canny. Planning for modular and hybrid fixtures. In *IEEE International Conference on Robotics and Automation*, May 1994.

[16] Aaron Wallack, John Canny, and Dinesh Manocha. Object localization using crossbeam sensing. In *IEEE International Conference on Robotics and Automation*, pages 692–699, 1993.

[17] D.E. Whitney. Quasi-static assembly of compliantly supported rigid parts. *ASME Journal of Dynamic Systems Measurement and Control*, pages 65–77, 1982.