INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Basis Functions for Use in Direct Calibration Techniques to Determine Part-in-Hand Location

by

Ulix Goettsch

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

University of Washington

2001

Program Authorized to Offer Degree: Department of Mechanical Engineering

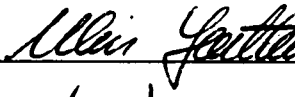
UMI Number: 3014074

UMI Microform 3014074

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Doctoral Dissertation

In presenting this thesis in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of the dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to UMI Dissertation Services, 300 North Zeeb Road, P.O. Box 1346, Ann Arbor, MI 48106-1346, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature 

Date 3/15/2001

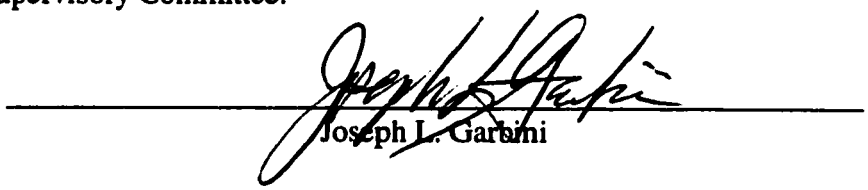
University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Ulix Goettsch

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

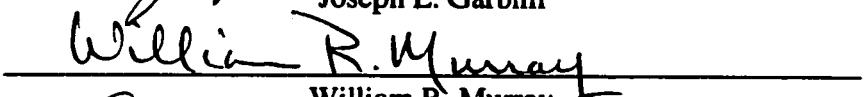
Chair of Supervisory Committee:



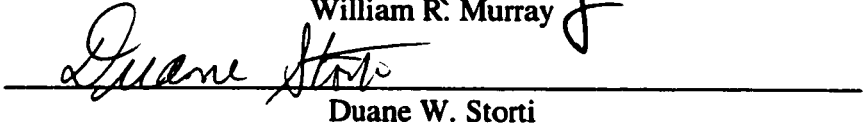
Joseph L. Garbini

Reading Committee:

Joseph L. Garbini



William R. Murray



Duane W. Storti

Date: 3/14/01

University of Washington

Abstract

Basis Functions for Use in Direct Calibration Techniques to Determine Part-in-Hand Location

by Ulix Goettsch

Chairperson of the Supervisory Committee: Prof. Joseph L. Garbini
Department of Mechanical Engineering

A common difficulty in implementing flexible automation is the complexity and lack of robustness of the intercalibration of all the components of the workcell and the part itself.

A novel approach to the calibration problem, called Direct Calibration, calibrates the entire assembly system – the robot, the sensors, and the parts to be assembled – in a single procedure. The relationship between feature information in sensor coordinates and the part location in robot coordinates is determined in three steps: calibration data are generated by using the robot to move the part to be assembled under the view of the sensors by known amounts; the best-fit mapping representing this assembly process is calculated; and this mapping is used in production to estimate the part locations from the current sensor data.

While this approach has been demonstrated to work very well in its current application, this research establishes which basis functions should be used in the Direct Calibration mapping for optimal process performance. Furthermore, this work generates a better understanding of the significance of the basis functions for the process performance.

"Perfect" basis functions are developed for the most common classes of parts. These perfect basis functions are broken down into their elementary terms and

compared to the basis function set that was found to be suited best for a general application of Direct Calibration.

The limits to performance improvement of the Direct Calibration technique due to changes in basis functions for certain classes of parts are established in simulation.

A set of basis functions for the general application of Direct Calibration is recommended. This basis function set is shown to have wide applicability and high accuracy.

The performance potential of higher-order basis functions is evaluated in simulation.

An empirical technique for finding the optimal part-specific basis functions, without the need for a model of the part, is presented. This technique uses methods from experimental design and statistics and maximizes the performance of Direct Calibration for a specific part.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This dissertation is dedicated to my parents Heide and Dr. Manfred Göttsch. Without their inspiration and support, I would not have made it this far.

I wish to thank my advisor Prof. William R. Murray for his mentoring and guidance throughout the years. The friendly work atmosphere was much appreciated.

Thanks to Prof. William R. Murray and Christopher M. Pohlhammer for developing the Direct Calibration technique that set the foundation for my work.

Thanks to Chris Rasmussen and Randy Krauter at Hewlett-Packard for funding and real-world expertise.

CHAPTER 1: INTRODUCTION

The average market life of consumer products has decreased drastically over the last two decades. To remain competitive in this changing market place, manufacturers must minimize their time to market – the time it takes to get a product through design, up the ramp to full production, and to the market in volume. Manufacturing today is also characterized by high production volume which often results in the use of automated assembly [97, 108, 110]. Reducing time to market necessitates a reduction in the time to design and implement the assembly automation equipment for a new product. Traditional dedicated automation equipment – that is designed to handle one specific assembly task efficiently and precisely [15]– has a long lead time because of its complex mechanical makeup and often inexact development methods. The design of bowl feeders, for example, is often described as a "black art" [38]. Furthermore, product design and production tooling must be finalized before dedicated automation equipment can be implemented.

Flexible assembly can solve these problems by "going up the ramp", or evolving, with the changing part design [17]. Flexible assembly systems use generic hardware such as two-finger, parallel-jaw grippers, and employ computer vision and sophisticated programming to allow a high degree of flexibility in part shape and assembly task [24]. Since the hardware used in a flexible assembly cell is

generic and modular, it is easy to adapt the software to a changing part design so that the assembly system can evolve along with the part design.

One problem with flexible automation is the complexity and lack of robustness of the intercalibration of all the components of the workcell and the part itself. Whenever sensors are used in a robotic assembly cell to compensate for variations in part location, part-to-part variations, and robot inaccuracies, the spatial relationships between the robot, the sensors, and the parts to be assembled must be accurately known. Traditionally, this calibration has been treated as related only to the assembly system itself, independent of the part and the assembly task [53]. In such an approach, the transformation of sensor data to estimates of part location then depends on a CAD model-based algorithm, which can be highly sensitive to the accuracy to which the model represents the real production parts.

A novel approach to the calibration problem, called Direct Calibration, calibrates the entire assembly system – the robot, the sensors, and the parts to be assembled – in a single procedure [87]. The relationship between feature information in sensor coordinates and the part location in robot coordinates is determined in three steps: calibration data are generated by using the robot to move the part to be assembled under the view of the sensors by known amounts; the best-fit mapping representing this assembly process is calculated; and this mapping is used in production to estimate the part locations from the current sensor data.

While this approach has proven to work very well in its current application [75], this research establishes which basis functions should be used in the Direct Calibration mapping for optimal process performance. Furthermore, this work generates a better understanding of the significance of the basis functions for the process performance.

The research described herein answers the following questions.

- Can the Direct Calibration performance be improved by a better choice of basis functions?
- How can the Perfect Basis Functions for certain classes of parts be found in simulation?
- How do these Perfect Basis Functions differ from the basis functions in use in the prototype Direct Calibration workcell?
- Which set of basis functions should be used for a general application of the Direct Calibration technique?
- If a general applicability of the Direct Calibration technique is not important, how can the performance for a specific part be optimized without requiring a CAD model of the part?

The current research application of Direct Calibration is a tray loading cell used for flexible feeding. Chapter 2 provides an overview over the field of flexible

feeding. Illustrated are the importance of part-in-hand sensing and the benefits of Direct Calibration in real-world applications. Chapter 3 introduces the tray loader cell that was used in this research and explains the Direct Calibration method in detail. The significance of the choice of basis functions for the Direct Calibration technique is outlined in Chapter 4. Perfect Basis Functions are developed for different classes of parts and their performance, as well as their expanded form, is compared to that of the basis functions used in the actual cell. Chapter 5 explores variations of the set of basis functions that was found to work best for a general application of Direct Calibration. In Chapter 6, future research topics are recommended and a method for finding the Optimal Basis Functions for one particular part is outlined. Chapter 7 summarizes the findings of this research.

CHAPTER 2: A REVIEW OF FLEXIBLE FEEDING

The traditional approach for high-volume assembly has been to use custom-designed, dedicated automated assembly systems that rely on hard tooling; however, the long lead time necessary to design and implement these complex assembly systems has become prohibitive. This shortcoming of traditional automation has given rise to a growing demand for modular, flexible automation for assembly [97, 108, 110]. Here, flexible refers to a system that can handle a wide variety of parts through reconfiguration involving changes primarily in the software, with changes in the hardware limited to nothing more involved than simple mechanical adjustments. With flexible assembly systems, a significant reduction in time to market is achievable because a considerable portion of the assembly cell hardware can be specified much earlier in the product design cycle than is possible with dedicated automation. In this regard, flexible automation for assembly is important to achieving the full benefits of concurrent engineering. Furthermore, since the capital equipment involved can be depreciated over multiple generations of products, the ability to reuse these flexible assembly modules and rapidly reconfigure them for new or redesigned products is economically attractive.

Although great strides have been made in flexible automation systems for the assembly of discrete parts [92, 101], the technology for parts feeding has not kept pace. Feeding parts to flexible assembly systems frequently is the bottleneck for

high-volume assembly [32, 35, 97, 108]. Here, feeding refers to locating (positioning and orienting) the parts to be assembled in a manner suitable for grasping by a robot in the flexible assembly system. Over the last two decades, flexible parts feeders have been evolving, primarily through the addition of sensors to conventional parts feeders, such as vibratory bowl feeders. More recently, the trend toward less expensive but more powerful computer systems, coupled with advances in sensor technology, especially improvements in the hardware and software for machine vision, has accelerated this evolution. In the past several years, vision-based, flexible parts feeders have become commercially-available products [13, 97, 108].

In this review of flexible parts feeding, the operational aspects of parts feeding are presented and methods of achieving flexibility in parts feeding are examined. Two state-of-the-art flexible parts feeders are described in some detail. And finally, emerging technology related to flexible parts feeding is reviewed.

2.1 Parts Feeding

For a variety of economic and logistic reasons, it is common for component parts to be available only in bulk, that is, in hoppers, bags or other containers with each container holding many identical parts.

The objective of parts feeding is to deliver a sequence of parts in which each part has a specified location (position and orientation) so that it can be grasped by a robot or directly enter some other downstream process. To achieve this

objective, the two basic tasks of a parts feeder are to singulate parts from bulk storage and to orient and position individual parts.

A wide variety of mechanical approaches have been applied to part singulation. Once singulated, parts will be in one of the small number of stable poses possible for that part. As pose is used herein, each of the different poses of a part is distinguished by a different set of points of contact between the part and the surface on which it rests. For example, a cubic part has six possible stable poses on a flat surface, which correspond to the six sides on which the cubic part can rest. Part pose is important in part feeding because which part features are hidden and which are accessible are uniquely specified by part pose.

In general, singulated parts can be oriented and/or positioned in one of three ways. The part orientation and/or part position can be *preserved*, such as by transporting the parts in formed trays; mechanically *enforced*, for instance by the orienting devices in a bowl feeder, or *sensed*, as in most vision-based feeders. Although sensorless approaches to flexible parts feeding are beginning to receive considerable research attention, sensing is the predominant practicable approach to achieving flexibility in systems for feeding discrete parts.

Conventional feeders, such as vibratory bowl feeders, are widely used and perform well once designed and configured [2, 15]. However, conventional approaches to parts feeding are not flexible. In the context of parts feeders, flexible refers to a system that can handle a wide variety of parts through reconfiguration involving changes primarily in the software, with changes in the

hardware limited to nothing more involved than simple mechanical adjustments. This flexibility is required when a minimal time-to-market of the product is essential.

2.2 Flexible Parts Feeding

In this section, the operational aspects of parts feeding are presented and the various methods of achieving flexibility in parts feeding are examined. Conventional parts feeders that have evolved by incorporating sensors to provide flexibility are discussed, as are the more inherently flexible vision-based parts feeders.

2.2.1 Operational Characteristics of Parts Feeders

The major operational characteristics of a parts feeder are shown in Figure 2-1. The ovals represent the states of the parts, while rectangles denote feeder operations. Parts are usually delivered in bulk; however, the goal of feeding is for parts being fed to end up in the state having a prescribed orientation and position that is suitable for subsequent grasping by a robot or subsequent entry directly into some downstream process. The three basic functions of a parts feeder are singulating, orienting, and positioning. These three functions can each be achieved in either of two ways: the desired state can be *enforced* mechanically by the feeder, or the existing state of the parts can be *sensed* and subsequently accommodated. For example, a part can be manipulated into a prescribed orientation or its present orientation can be sensed and accommodated by the

robot. The common combinations of enforcing and sensing are the five feeder paths labeled in Figure 2-1.

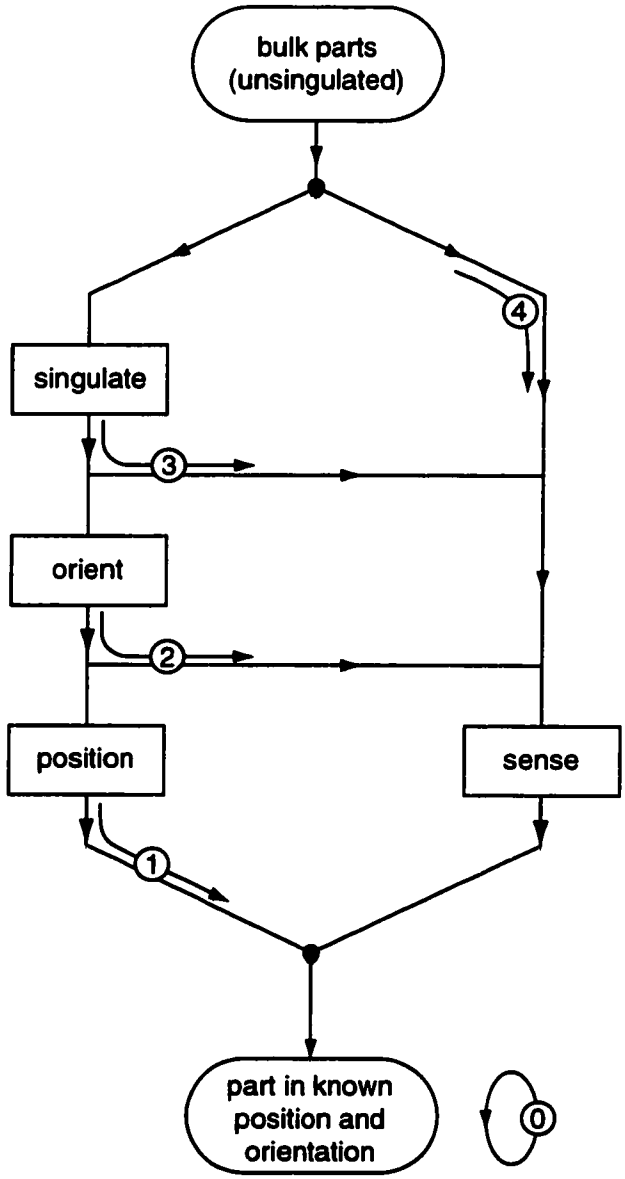


Figure 2-1: Flow diagram showing several operational paths through which bulk parts can be fed from an unsingulated state in a hopper to a state in which the position and orientation of each individual part are known at the output stage of the parts feeder.

Path 0 – Transport the parts to the assembly process in a manner that presents them at the desired position while maintaining the desired orientation. This is possible only if the parts have the desired orientation initially.

Path 1 – Singulate the parts, then mechanically enforce a specific predefined orientation and position.

Path 2 – Singulate the parts mechanically, enforce a predefined orientation, and then sense the position of the oriented parts.

Path 3 – Singulate the parts mechanically, then sense both orientation and position.

Path 4 – Sense the orientation and position of parts without singulating them beforehand.

Preserve Orientation and Position (Path 0)

Delivering the parts to the assembly robot in a predefined orientation and position makes possible the use of blind assembly robots, which reduces the complexity and cost of the assembly system [33]. Tray-based systems, such as the Sony SMART system [92, 101], deliver a tray of identical parts to the assembly cell. The tray is fixtured in position and the robot grasps parts blindly from an array of accurately located cavities in the tray. Although this approach

does simplify parts handling in the immediate neighborhood of the assembly cell, the new question of how to get the parts in the trays in the first place must be answered.

Other common approaches for preserving both orientation and position in parts feeding are the use of tubes for feeding cylindrical components [2], the use of tapes for electrical components and small subassemblies [1, 3, 6, 63, 101], and the use of various other types of magazines as described in [109] and [2].

Orient and Position without Sensors (Path 1)

Bowl feeders, either vibratory or centrifugal, are the traditional mechanism for feeding discrete parts [15]. These feeders do not require sensing, but the design of the bowl and its orienting devices has been called "a black art" [38]. The long lead times for bowl feeder design and construction, coupled with the fact that even slight changes in part design frequently necessitate the retooling of a bowl, make bowl feeders fairly inflexible. The simplification of the bowl feeder design process by simulating the feeding process is an active area of research [9, 10, 21, 28, 56, 57, 68, 80, 91, 94, 105].

Sensorless feeding is very robust in those cases where significant advantage can be taken of special part geometry. A feeding system for headed parts such as screws and bolts is described in [66]. This system is easily reconfigured for other shapes and sizes of headed parts.

The problem of loading parts into trays for the Sony SMART system described in the previous section has been answered for various classes of small parts by APOS [48, 101], a tray-loading system in which parts are tumbled over an inclined tray that is vibrated. The tray contains an array of accurately located identical cavities, with each cavity designed to capture a part in the proper orientation and reject parts in all other orientations. Once a part is captured in one of the cavities, its orientation and position relative to the tray are enforced.

Mechanically Preorient with Subsequent Sensing (Path 2)

Rather than trying to completely orient a part by moving it through a series of passive fences with a conveyor belt, such fences can be used to reduce the number of possible orientations of the part [43, 44, 49, 50, 86, 104, 108]. This approach is applicable to a wide range of parts and simplifies the subsequent vision processing required for completely determining the orientation of the part.

Sense Orientation and Position after Singulation (Path 3)

Most modern parts feeders use machine vision of some form to determine part orientation and part position after the parts have been singulated. These feeders can be grouped into two classes: conventional feeders with sensing, which consists mainly of bowl feeders to which sensing has been added to increase flexibility or decrease design complexity; and vision-based parts feeders, which typically use conveyor belts to move parts into the field of view of a machine vision system that is used to sense orientation and position of parts. These two

classes of flexible parts feeders are treated in more detail in the subsections that follow.

A number of parts feeding systems sense part orientation but have no provision for enforcing or sensing the position of the part [22, 23, 29, 47, 67]. These systems are generally used for part qualification [18, 37] or other applications in which only orientation is required [23]. Otherwise, to complete the part feeding process, these systems must rely on the addition of an output track or magazine that mechanically locates the part.

Sense Orientation and Position without Singulation (Path 4)

The process of sequentially extracting parts from a hopper containing unsingulated parts is called bin picking. Even when the hopper is known to contain only one type of part, identifying individual parts in such a bin, selecting an accessible part near the top of the heap, and computing the part location and path clearance information necessary for a robot to grasp the part are demanding tasks for any sensor system [12, 14, 36, 58, 78]. Clearly, bin picking provides the ultimate flexibility for parts feeding; however, due to the inherent complexity of the task, reliable bin picking is generally unachievable for industrial parts, at least, unachievable at a cost competitive with other options for parts feeding.

2.2.2 Conventional Feeders with Sensing

Sensors have been added to conventional parts feeders, primarily vibratory bowl feeders and centrifugal bowl feeders, for two main reasons: the nondestructive

qualification of parts and the determination of part pose. In either case, stationary instrumentation is added to sense parts as they move by on the output track of a bowl feeder or on a conveyor that has been fed by a bowl feeder. A variety of acoustic, inductive, capacitive and microwave sensors have been used for nondestructive qualification of parts [18, 37], and machine vision systems have been used for determination of part pose [41, 47, 55, 71, 77, 104] or simultaneous pose determination and nondestructive qualification [29]. If special features can be designed into the part, the vision processing can be greatly simplified [54, 72].

These conventional feeder systems with sensing have some means, typically an air jet [41, 47, 67] or pneumatically-activated mechanical gate [23, 29], for eliminating from the output stream of parts those parts that have been deemed unacceptable. In nondestructive parts qualification systems, the failed parts are diverted to a rejection bin; whereas in the pose determination systems, parts in an incorrect pose are diverted to a chute that returns them to the bowl of the vibratory feeder. In some systems, parts that are upside down are turned over into a graspable pose resulting in no rejection of parts [17, 104]. While it is possible to stop the feeder to grasp a correctly oriented part directly from the feeder track [71], this is usually avoided to maximize feed rate. Because the parts are sensed on a moving conveyor, part position typically is not measured and the parts cannot be grasped by a robot without additional sensing or additional tooling to mechanically locate the parts. In an exception, the robot "pursues" and grasps the moving part on the feeder track [55]. Since the bowl is used solely for

singulation in these systems, several different part types can be fed with one feeder [71]. However, cycle time may suffer if the robot has to wait for the desired part in the desired orientation to arrive. For certain classes of parts, the parts can be fixtured mechanically before they are sensed [104]. Although such fixturing increases the mechanical complexity of the system, it does reduce the complexity of the vision algorithm and facilitates position determination at the same time.

2.2.3 Vision-Based Parts Feeders

Most vision-based, flexible parts feeders use some form of vibrating element to singulate incoming bulk parts [13, 46, 51, 62, 108]; however, some systems use inclined belts [26, 49]. Once singulated, parts will be in one of the small number of stable poses possible for that part. At this point, the parts are sensed to determine part pose, as indicated schematically in Figure 2-2. Identification of part style typically is not required because only one type of part is usually fed at a time, though some research has been done on feeding multiple part styles at once [16]. Parts in unacceptable poses are either recycled back into the bulk parts hopper [13, 19, 26, 49, 62, 108] or are reoriented into a new pose and sensed again [97, 110]. As shown in Figure 2-2, after a part has been deemed to be in an acceptable pose, the orientation and position of the part are determined by further processing of the image used to determine part pose, or by acquiring and processing a new image of the part. The vision systems used are mixed in

regards to lighting: some use back lighting [13, 19, 26, 46, 62, 108], whereas others use top lighting [7, 49, 97, 103].

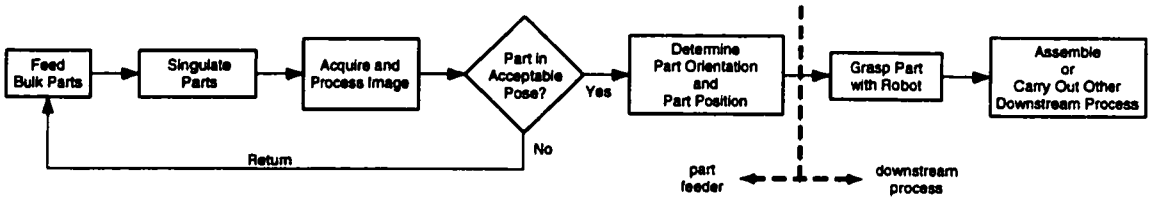


Figure 2-2: Diagram of the major steps in a common automation cell containing a vision-based, flexible parts feeder.

Vision-based, flexible parts feeders have been evolving over almost two decades. A forerunner of this type of feeder, which used discrete photocells rather than CCD cameras, required extensive mechanical preorientation in order to reliably determine part orientation for a restricted class of parts [86]. Early vision-based parts feeders were generally limited in the variety of parts they could successfully feed. The limited resolution of early machine vision systems often necessitated two-stage sensing, where a fixed overhead camera was used to locate parts in the field of view, and an arm-mounted camera was used to sense orientation and position of the part more accurately [7, 103]. More recently, vision-based, flexible parts feeders have been a topic of active research in university research laboratories [16, 19, 26, 27, 49, 74, 75]. In the past several years, vision-based, flexible feeders have become available commercially [13, 97, 108], and major industrial applications have shown that this type of flexible feeder fares well on the factory floor [64, 92].

2.3 State-of-the-Art Flexible Parts Feeders

Recent technological trends, particularly the availability of less expensive but more powerful computer systems and the advances in the hardware and software for machine vision, have led to a new generation of flexible parts feeders [13, 19, 20, 26, 43, 44, 49, 50, 52, 64, 69, 79, 88, 95-97, 108]. The state-of-the-art in flexible parts feeding is well represented by two of these modern vision-based parts feeders: the FPF2000 Flexible Feeder for Small Parts [43, 44, 108] by Intelligent Automation Systems, Inc. (IAS) and the Adept FlexFeeder 250™ [13, 25, 52, 64, 79] by Adept Technology, Inc. Details of these two feeders are presented in the remainder of this section.

2.3.1 Flexible Feeder Operation

In these vision-based, flexible parts feeding systems, a vibrating conveyor system or vibrating plate is used to singulate the incoming bulk parts. Singulated parts then are presented on a back-lit conveyor to a downward-looking CCD camera, as indicated schematically in Figure 2-2. In the Adept FlexFeeder 250™, multiple parts are presented to the vision system for each image; whereas for the IAS FPF2000, programmable mechanical gauges align singulated parts in single file as they move through the system, and a single part is presented to the vision system for each image. Each part presented to the vision system will be on the back-lit conveyor in one of the small number of stable poses possible for that part. Based on the part geometry and the desired operation for the part once it has been grasped by the robot, there are a limited number of possible points at

which the part can be grasped or picked. Stable poses are classified as pickable or unpickable, depending on the accessibility of a pick point.

A major task of the vision system in a flexible parts feeder is to determine the pose of the part being examined. Identification of part style is not required since only one type of part is fed at a time. Once a presented pose is identified as pickable, the position and orientation of the part must be determined. Because the part is at rest on a flat surface, orientation of the part is restricted and can be described simply by a rotation about an axis perpendicular to the back-lit conveyor. In the IAS FPF2000, the programmable mechanical gauges are used to configure the part path such that parts in undesirable poses fall off of the edge of the conveyor, where they fall onto a separate conveyor and are returned to the bulk parts hopper. Vision processing is based on a template matching approach in which a silhouette of the part is extracted from the image and compared to an array of templates generated in the calibration or training procedure. A mirror at the edge of the conveyor is positioned at 45° to the camera axis to allow both a top view and a side view of the part to be captured in a single image. This feature adds additional information that is useful when a top view is not sufficient to distinguish part pose [44]. In the Adept FlexFeeder 250™, because multiple parts are in the field of view of the camera, the image must first be processed to determine all candidate parts. Afterwards, a suite of vision processing tools is applied in turn to each candidate part in the image to determine part pose as well as part position and orientation for parts in pickable

poses. These vision processing tools range from connectivity analysis to advanced tools for disambiguation of similar but distinct stable poses [76].

After the part location information for a part in a pickable pose has been transmitted to the robot, the part is grasped by the robot and the desired manipulation or assembly task is carried out. In operation, an unpickable pose is rejected and the next part is considered. For the Adept FlexFeeder 250™, parts that are found to be in unpickable poses remain on the back-lit conveyor until all candidate parts in the image have been examined, at which time the back-lit conveyor is advanced to bring new parts into the field of view. Unselected parts eventually travel to the end of the conveyor, where they fall onto a separate conveyor that returns them to the bulk parts hopper. Because the vision system in the IAS FPF2000 accepts only one part at a time, any part found to be in an unpickable pose is immediately indexed off the back-lit conveyor onto the return conveyor. The preorientation carried out by the programmable mechanical gauges reduces the likelihood of an unpickable pose at this stage.

The changeover from feeding one type of part to feeding a different part involves purging the feeder of parts by closing the supply hopper and allowing the remaining parts to circulate through the system and back to the bulk parts hopper. The feeder then is reconfigured for the new part type. On the IAS FPF 2000 all adjustments necessary for reconfiguration are programmable, so that only software changes and no manual intervention are required for part changeover. For the Adept FlexFeeder 250™, the typical adjustments required for

part changeover are programmable, while some less frequent adjustments must be made manually if required. For both systems, the time required for part changeover is brief.

Although these two flexible feeders differ significantly in the details of their approach to parts feeding, these systems each perform the four basic functions of a vision-based, flexible parts feeder: isolate single parts on the flat, back-lit output conveyor of the feeder; determine part pose; calculate the part location for parts that are in a pickable pose; and recirculate parts that were presented in an unpickable pose.

2.3.2 Flexible Feeder Capabilities

These flexible feeders currently can handle parts with a minor dimension as small as 0.5 mm (0.02 in) and a major dimension of up to 65 mm (2.5 in). Cycle times depend upon the robot used and on the details of the parts being fed, but generally range from one to four seconds per part, with 2-3 seconds per part (20-30 parts per minute) being typical. Based on pose frequencies estimated from CAD data for the parts being fed, feeder throughput can be predicted for the Adept FlexFeeder 250™ [39, 70], and the optimal belt speed to maximize feeder throughput can be estimated [45].

2.3.3 Flexible Feeder Limitations

Other than the limitations on part size and cycle time mentioned above, and the obvious limitations that parts must have at least one suitable stable state when resting on the flat, back-lit conveyor and must be durable enough to withstand multiple passes through the feeder, the major limitation of flexible part feeders is that the location of the part is sensed before the part is grasped by the robot.

Due to errors in the vision processing, errors in the robot and gripper, and part-to-part variations in the parts being fed, the grasping error, which is the total distance between the actual pick point and the ideal pick point, may be as large as ±2 mm (±0.08 in) [75, 76]. For a given resolution image system, the vision processing errors tend to increase with the size of the field of view and the size of the part. This effect is more significant with the Adept FlexFeeder 250™, where the larger field of view does potentially increase feeder throughput, but also reduces resolution and increases distortion in the vision system. The current approach to overcoming grasping errors, which can be large with respect to the overall process tolerance, is to use part-specific grippers on the robot to mechanically ensure that the part is correctly registered relative to the gripper. For those cases in which grasp error is important, such as assembly, this type of flexible parts feeder typically is used with robots having part-specific grippers.

2.4 Emerging Technology for Flexible Parts Feeding

2.4.1 Recent Research in Part Manipulation

Recent research in the area of parts manipulation may lead to a generation of novel flexible part feeders. Research in this field strives towards orienting parts with little or no sensor information.

The following are some of the part manipulation strategies that have been proposed:

- moving a table on which the part rests against fixed but adjustable fences [65]
- manipulation a part through contact with the walls of a tilting tray [34]
- a 3-DOF, horizontally-vibrating plate [93]
- moving the part through a series of curved fences using a conveyor belt [11, 84, 107]
- manipulating the part with two single-degree-of-freedom rotating fences [112]
- manipulating the part with a single-degree-of-freedom rotating fence while the part is being transported by a conveyor belt [4, 99, 100]
- toppling (knocking the part over) [61]
- tumbling the part down a series of steps [111]
- pushing [5, 98, 102]
- grasping the part with a parallel-jaw gripper having a linear bearing on one jaw [38, 90]

- grasping the part with a parallel-jaw gripper with pivoting fingers [89]
- manipulating the part with a force field generated by an array of manipulators [30, 60]

Working towards an analytical design of a parts feeder, some authors provide rigorous mathematical algorithms for planning these motion strategies [11, 89, 107].

Very relevant to part feeding are methods that can orient parts by a predefined sequence of blind manipulations. For certain classes of parts, a specific sequence of manipulations results in the same final orientation of each part, irrespective of the initial orientation [11, 34, 40, 84, 112]. This research may lead to a new generation of sensorless parts feeders in the future.

Sensorless manipulation methods which cannot manipulate parts to the desired orientation from any possible initial orientation can be used to reduce the uncertainty in the orientation of the part, as described above in Path 2.

2.4.2 Accomodating Part-in-Hand Error

Just as the singulation, orientation and position of a part can be enforced mechanically by a parts feeder or the existing state of the part can be sensed and accommodated, the orientation and position of a part relative to the gripper can be enforced or sensed and accommodated after the part is grasped. Part-specific

grippers overcome grasp error by enforcing a particular orientation and position of the part relative to the gripper once the part has been grasped. As shown in Figure 2-3, an alternative approach to overcoming grasping errors is to sense the location of the part relative to the robot gripper after the part has been grasped.

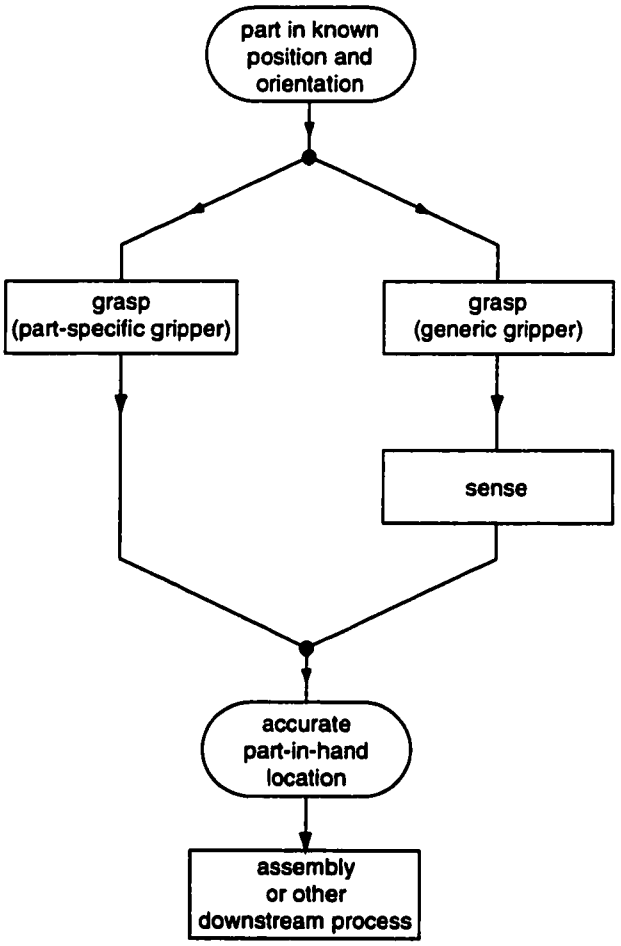


Figure 2-3: Flow diagram showing two approaches for overcoming significant grasp error: enforce the required orientation and position with a part-specific gripper; or sense the location (position and orientation) of the grasped part relative to the generic gripper and alter the robot path for the subsequent process to accommodate the grasp error that was sensed.

Several ways of directly measuring the part-in-hand location for a part in the grasp of a robot have been reported. One method, which has been used successfully for years in component placement on circuit boards, is to use an upward-looking CCD camera to measure features of a part held by the robot [53]. Also used for component placement on circuit boards are robots with a camera mounted to the robot arm, which are called camera-in-hand or eye-in-hand systems [8]. These methods, however, are typically restricted to planar parts and allow correction in only the planar degrees of freedom. In order to handle non-planar parts or to correct for part misalignment in more than three degrees of freedom, both of which are common with discrete mechanical parts, more sophisticated sensing is required. Conventional approaches for measuring the location of non-planar parts range from using two CCD cameras for stereo vision to using even more sophisticated, real-time, three-dimensional sensing systems [42]. Two novel, straightforward approaches to directly measuring the part-in-hand location of a part in the grasp of a robot are described here [24, 75, 83, 106]. These approaches are inexpensive in that they do not require sophisticated sensors; simple in that they are self-calibrating and in that they do not require a CAD model of the part; and robust in that recalibration is relatively quick and easy.

The first of these approaches uses a nest of three simple optical beam sensors, each of which responds to the presence or absence of an object along the beam line [24, 83, 106]. The robot passes the part through this sensor nest at a constant velocity. The position of the robot is recorded when each beam is first broken by

the part and again when each beam is reconnected after the part passes. From this information, the location of the part relative to the robot gripper can be calculated in real time quickly enough for adjustment of the paths required in the subsequent assembly operation. In an evaluation using cylindrical parts, this system has been shown to be accurate to 0.025 mm (0.001 in).

The second of these approaches uses a CCD camera to observe how a grid of laser lines breaks across the part in the grasp of the robot [75]. After the robot has picked up a part from the output conveyor of the flexible feeder, the part is positioned in the active area of a stationary sensor pod, where the CCD camera takes an image of the intersection of a tic-tac-toe pattern of laser lines with the part. From the information available in this image, the location of the part relative to the robot gripper can be calculated in real time quickly enough for adjustment of the paths required in the subsequent assembly operation. In a tray-loading application involving a variety of inkjet printer parts having moderately complex geometry, this approach has been shown to compensate for more than 90% of the error in part-in-hand location that was present after the part was grasped from the output conveyor of the flexible feeder. This tray-loading station was used to generate the data for the research presented later in this document.

Although systems using these approaches for measuring part-in-hand location for discrete mechanical parts are not commercially available at present, these approaches provide significant improvements in overall accuracy and are

sufficiently simple, inexpensive and easy to use that they may be commercially viable in the future.

CHAPTER 3: THE TRAY LOADER CELL AND DIRECT CALIBRATION

The research described herein focuses on an important aspect of the Direct Calibration technique, the choice of basis functions. So that the significance of the choice of basis functions can be appreciated, a review of Direct Calibration is warranted. In the Direct Calibration approach to robotic workcell calibration the entire manipulation system – the robot, the sensors, and the parts to be handled – is calibrated for the required task in a single procedure to directly determine the relationship between the part feature information in sensor coordinates and the part location in robot coordinates. The prototype application of Direct Calibration is a tray loading cell which is presented with parts in unknown poses and orientations and sorts these parts into trays. This tray loading cell is used to illustrate the Direct Calibration concept. The following two sections outline the tray loading cell and explain Direct Calibration in detail. The last section illustrates how the performance of the system is evaluated.

3.1 Description of the Tray Loader System

The purpose of the tray loading cell is to sort inkjet printer parts that are received in bulk into part-specific trays so that a downstream assembly process can use blind robots to grasp the parts from known locations in the tray. Six different inkjet printer parts are considered as shown in Figure 3-1.

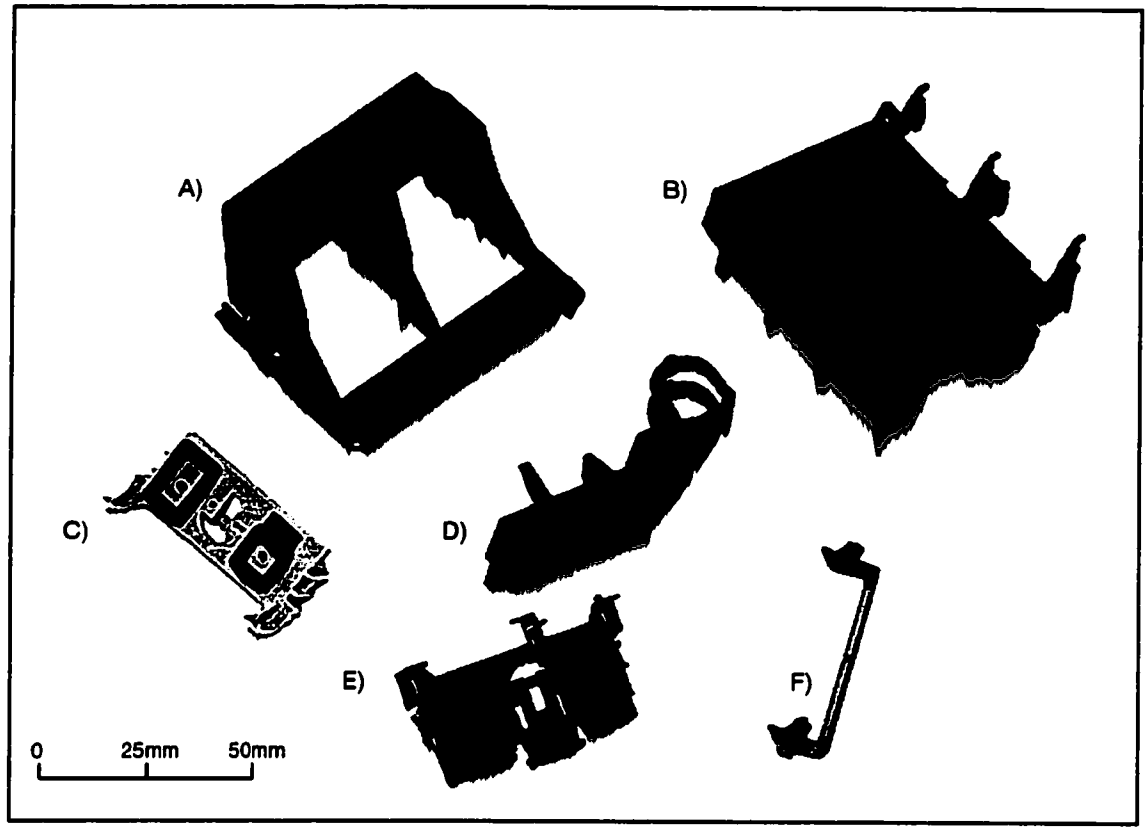


Figure 3-1: The six printer parts handled by the tray loader

The tray loader system shown in Figure 3-2 centers around a 6-DOF serial link robot with a robot controller that is slaved to a Macintosh computer. The Macintosh also contains a video frame grabber that can digitize raw image data from the initial sensing and final sensing cameras in the workcell. Image processing software running on the Macintosh is used to find and identify features in the acquired images.

Figure 3-2: The tray loader cell.

In production, a part in unknown pose and orientation is dropped onto a backlit platform that represents a commercially available feeder [64, 108] and presented to the first stage of sensing. A downward looking camera takes a picture of the part silhouette and transfers the image to the computer via the frame grabber. Connectivity analysis and edge detection are used to identify the position and orientation of the part or to reject the part if it is in an ungraspable pose. If the

part is graspable, a location frame is attached to the part. Figure 3-3 shows an initial sensing image with video processing performed.

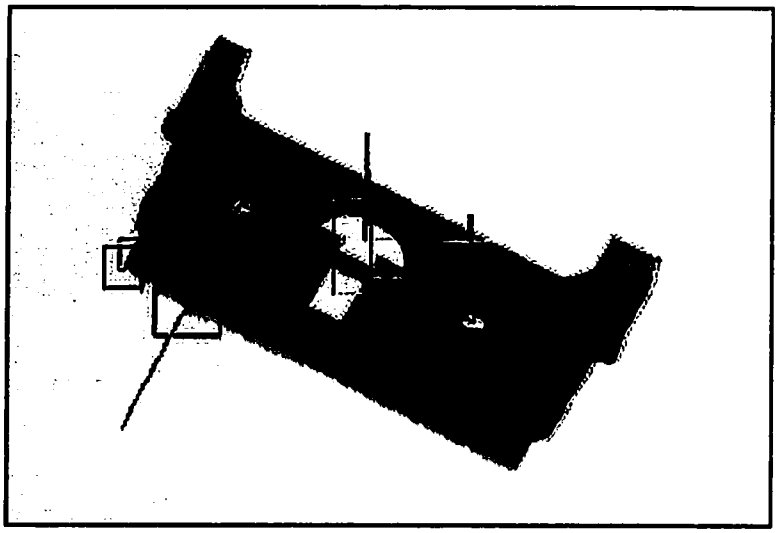


Figure 3-3: Initial sensing image with vision processing

The robot gripper is then commanded to a grip point predefined in the part location frame and picks up the part. While the location of the part is known with sufficient accuracy for pick-up after initial sensing, the grasping process with the parallel-jaw gripper reorients the part sufficiently so that the knowledge of the exact part location with respect to the gripper is lost. The part-in-hand error is measured as an offset from the part position at calibration. Since the part is constrained by the parallel jaw gripper, the part-in-hand error is constrained to three degrees of freedom, two translations and one rotation. To determine this part-in-hand error, the robot moves the part into a grid of laser lines where a final sensing camera, fitted with a laser light specific optical band-pass filter, acquires an image of the laser lines falling onto the part. The vision system then

accurately locates the ends of the laser lines as they fall off the part edges as shown in Figure 3-4.

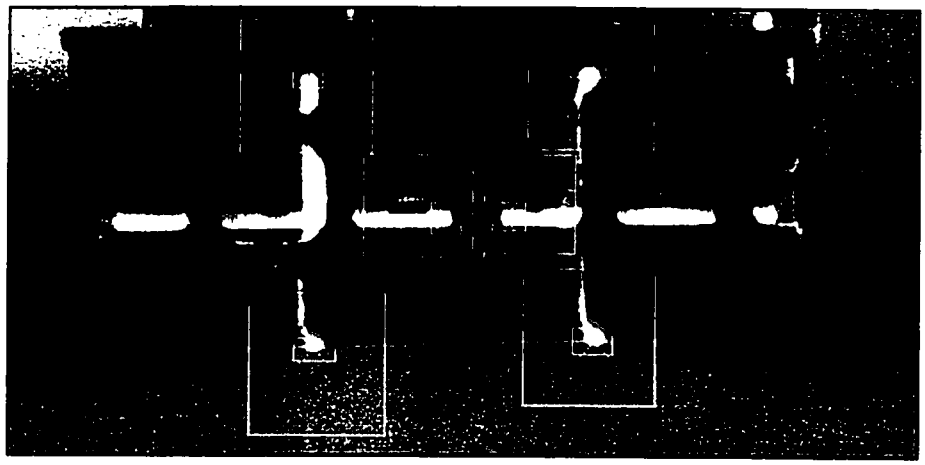


Figure 3-4: Final sensing image: Part with laser lines and vision processing performed.

These sensor data are used to calculate the part-in-hand location via a mapping determined using the Direct Calibration approach described in the next section. The nominal insertion path, which was taught at calibration, is then offset by the part-in-hand error and the part inserted into the tray.

3.2 Direct Calibration

The Direct Calibration process is divided into three steps. The first two steps comprise the calibration process whereas the third step takes place during on-line operation.

Acquisition of Calibration Data: The robot moves the part by known increments in the view of the camera. The ends of the laser lines are sensed as the calibration data.

Determination of Process Mapping: An optimization process is used to find the best-fit mapping between part location in robot coordinates and sensor data.

Determination of Part Location: The mapping is used to determine the part location relative to the location of the part at calibration.

The three steps are discussed in more detail in the following subsections.

3.2.1 Acquisition of Calibration Data

Calibration data are gathered by moving a part in the grasp of the robot gripper around a nominal position and recording the location of all features at every perturbation. The calibration range for this application is $\pm 2mm$ for the translations and $\pm 2^{\circ}$ for the rotation, based on expected part-in-hand errors. A uniform grid of calibration points over the calibration range is used.

For a set of N calibration patterns, the set of calibration vectors is

$$\mathbf{L}_{CAL} = \begin{bmatrix} \mathbf{L}_1 \\ \vdots \\ \mathbf{L}_N \end{bmatrix}_{CAL} \quad \text{and} \quad \mathbf{S}_{CAL} = \begin{bmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_N \end{bmatrix}_{CAL},$$

where, for $1 \le p \le N$,

$$\mathbf{L}_p = \begin{bmatrix} X_p & Y_p & \theta_p \end{bmatrix} \qquad \text{(location vector)},$$

$$\mathbf{S}_p = \begin{bmatrix} s_{p,1} & \cdots & s_{p,nSen} \end{bmatrix} \qquad \text{(sensor vector)}.$$

The location vectors correspond to the location of the grip point at each part perturbation and are measured in robot coordinates. The ends of the laser lines are referred to as features. The sensor vectors correspond to the locations of these features and are measured in raw pixel space. The number of "sensors", *nSen*, is the number of sensor values measured for each perturbation of the part. Since each feature has two coordinates [x,y], *nSen* is twice the number of features.

3.2.2 Determination of Process Mapping

A linear least squares technique with nonlinear basis functions[1] is used to determine the process mapping between sensor values in camera space and part location in robot space. This relationship can be described by either a *forward* mapping or an *inverse* mapping as shown:

$$\mathbf{S} = f(\mathbf{L}) \qquad \text{(forward mapping)}$$

$$\mathbf{L} = f^{+}(\mathbf{S}) \qquad \text{(inverse mapping)}.$$

[1] Note that in the strict mathematical sense, the term "basis functions" implies that the members of the set are linearly independent. In this research, this requirement is relaxed.

To determine the part location at run time, the inverse mapping can be used on the sensor data. However, it has been shown that the use of the forward mapping with a run-time iterative inversion procedure is considerably more robust to variations in part geometry than direct use of the inverse mapping [87].

Since the true process mapping f is not known, it is assumed to be a linear combination of a set of basis functions. $\overline{\mathbf{L}}$, the augmented location vector, consists of these basis functions which are chosen to be linear and quadratic combinations of the components of the location vector. Therefore,

$$\overline{\mathbf{L}} = \begin{bmatrix} 1 & X & Y & \theta & X^2 & XY & X\theta & Y^2 & Y\theta & \theta^2 \end{bmatrix}$$

can be used to yield an estimate of the sensor vector $\hat{\mathbf{S}}$:

$$\mathbf{S} \cong \hat{\mathbf{S}} \equiv \overline{\mathbf{L}} \cdot \mathbf{C}_{FOR},$$

where

$\mathbf{C}_{FOR} = nBF \times nSen$ matrix of weights, and

nBF is the number of basis functions.

For the "quadratic" set of basis functions shown, $nBF = 10$. The N calibration patterns (sets of location perturbations and their corresponding sensor values) correspond to an overdetermined system of linear equations,

$$\mathbf{S}_{CAL} = \begin{bmatrix} \overline{\mathbf{L}}_1 \\ \vdots \\ \overline{\mathbf{L}}_N \end{bmatrix}_{CAL} \cdot \mathbf{C}_{FOR} = \overline{\mathbf{L}}_{CAL} \cdot \mathbf{C}_{FOR},$$

which can be solved for the calibration matrix, $\mathbf{C}_{FOR}$, using singular value decomposition (SVD) methods for non-square matrices.

3.2.3 Determination of Part Location

An initial guess for the location of the current part, $\hat{\mathbf{L}}_0$, is run through the forward mapping to yield $\hat{\mathbf{S}}$, an estimate of the sensor vector. Since the true sensor vector $\mathbf{S}$ is known, the difference between $\hat{\mathbf{S}}$ and $\mathbf{S}$ is an indication of the adequacy of the guess of the part location and $\hat{\mathbf{L}}_0$ can be adjusted to minimize this error. This iterative inversion procedure minimizes the cost function

$$F_{II} = \frac{1}{2}\sum_{j=1}^{nSen}(\hat{s}_j - s_j)^2 .$$

The $\hat{\mathbf{L}}$ that minimizes the cost function is then used as the estimate for $\mathbf{L}$.

3.3 Process Error Evaluation

The Direct Calibration system measures part-in-hand error, but for evaluation purposes this error can be simulated by intentional offsets of the robot hand. A part is grasped, presented to the Direct Calibration system and defined as having zero part-in-hand error (home position). Then the part is presented with an

intentional offset from the home position. Since the part is still in the same grasp of the robot and has not moved relative to the gripper, the offset of the part – subject to robot repeatability – is known. Since the part location $\mathbf{L}$ is known, it can be compared to the Direct Calibration location estimate $\hat{\mathbf{L}}$ to yield and estimate of the error at the grip point. However, because the worst-case error at the perimeter of the part is critical in a tray loading or assembly operation, the error at the grip point is translated to the point on the part that is furthest from the grip point. This approach converts the rotation error at the grip point to a more meaningful offset and rotation at the extreme point. It also accounts for the fact that larger parts require a smaller error at the grip point than smaller parts for a successful insertion or assembly operation. The part location estimation error $(\mathbf{L} - \hat{\mathbf{L}})$ translated to the part extreme is used to quantify the performance of the Direct Calibration system.

CHAPTER 4: BASIS FUNCTIONS

The mapping between sensor data in camera space and part location in robot space is at the heart of the Direct Calibration technique. At the time of calibration, a part-in-hand home position is defined and the insertion (or assembly) path taught with the part in this part-in-hand home position in the robot gripper. The locations of the ends of the laser lines when the part is in the home position are stored as the sensor zeros at the time of calibration. To determine a part-in-hand location during production, raw sensor data are gathered by locating the points where the laser lines end at the edge of the part in view of the camera and the difference between the raw sensor data and the sensor zeros is calculated. These data then need to be mapped to a part-in-hand location or part offset from its home position. The basis functions represent a part-independent guess at the nature of the complex geometric relationship between these two spaces. Hence, the basis functions make or break the success of the Direct Calibration process. Not only will the omission of key basis functions hurt performance when the true geometric relationships cannot be adequately constructed, but the inclusion of too many higher-order terms has the potential to exaggerate the effect of noise in the data, cause numerical problems in the matrix inversion and optimization routines, and require much larger calibration data sets.

The best basis function set for use in Direct Calibration must thus fulfill the following criteria:

1) Provide high accuracy in mapping between the two spaces.
2) Provide equally high performance across all types of parts expected to be used with the system.
3) Cause no numerical problems in the pseudo-inversion and optimization routines.
4) Function with reasonably sized calibration data sets.

The described research is a thorough investigation of the basis functions of the Direct Calibration process mapping. The goal of this research is to develop an understanding of the underlying relationship between part geometry and Optimal Basis Functions. This investigation includes finding the best basis functions for certain representative parts to optimize the performance of the production process, as well as suggesting basis functions for new parts without having to run experiments on them. The insight gained from optimizing the basis functions for certain classes of parts leads to the desired understanding of the systematic relationship of part geometry, basis functions, and process performance.

Sets of newly developed basis functions are tested in both simulation and experiments. Real world experiments are vital to include the non-linear effects of real sensors, part surface textures and geometries, and real robot characteristics which are difficult to simulate effectively.

4.1 Comparing Sets of Basis Functions

To compare the performance of different sets of basis functions, a performance metric must be chosen. A candidate a performance metric is the residual calibration sensor error. The residual sensor error at calibration is the difference between the true sensor values and the sensor values calculated by running the true location data through the generated mapping. This error indicates how well the mapping fits the calibration data points, but it does not measure the adequacy of the model for points between calibration points. Thus it is not sufficient to use the residual sensor error at calibration as a metric for the performance of a set of basis functions. It is really the performance of the system using actual production data that is of interest and is thus chosen as a metric. To test the robustness of the Direct Calibration process with a given set of basis functions, production data that involve part-in-hand error are simulated with data created from a set of random perturbations over the range of the calibration data. Figure 4-1 shows the uniform grid of training perturbations, while Figure 4-2 illustrates the random perturbations that simulate actual part-in-hand error.

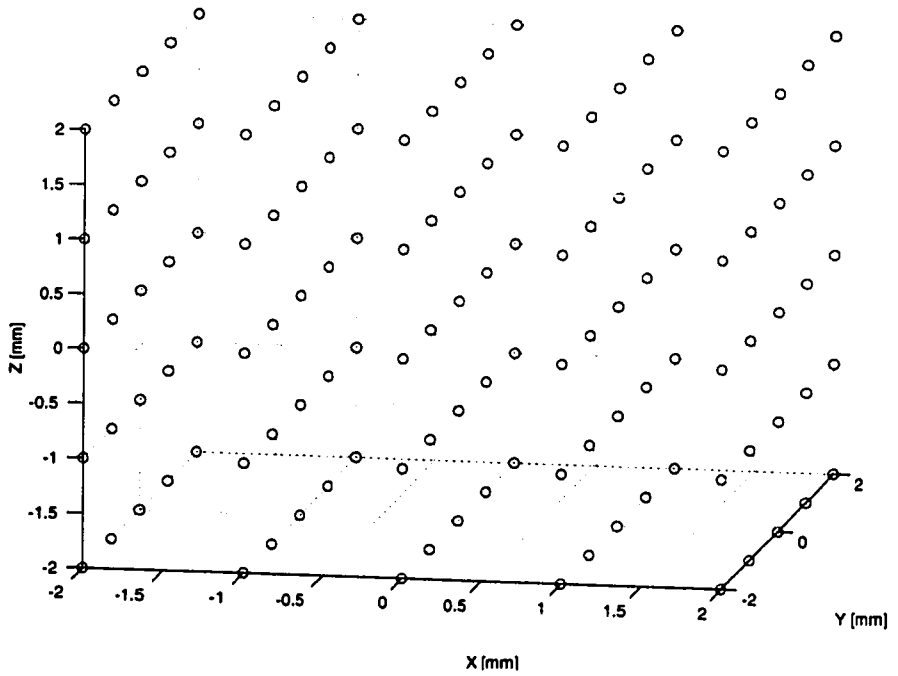


Figure 4-1: The uniform grid of perturbations used to train the Direct Calibration system.

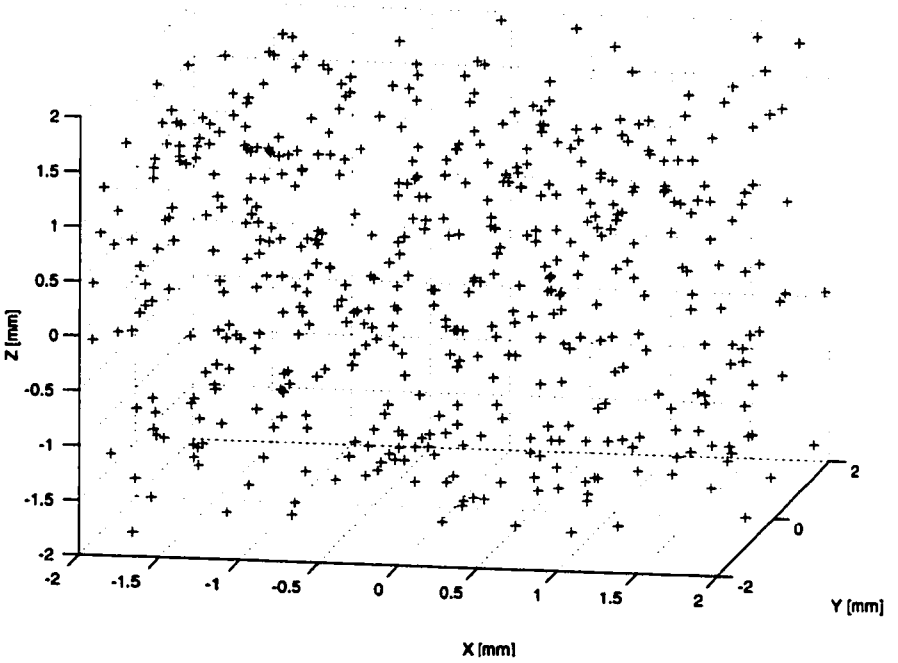


Figure 4-2: The random set of 500 evaluation points.

These evaluation data are taken in the actual cell, where the offset of the robot is specified to simulate unintended part-in-hand error. The residual location error is then defined as the difference between the part location calculated by running the sensor data through the mapping and the actual location as specified to the robot. This error is measured at the grip point of the part but is then translated to the part extreme as described in the last section of Chapter 3. The error at the part extreme is plotted for all 500 random evaluation points and the average error and maximum error are shown in an accompanying table.

The "Quadratic" Basis Functions currently used in Direct Calibration as shown on page 35 were selected without any investigation into the requirements of the Direct Calibration process on the mapping. The Quadratic Basis Functions have proven to work in the tray loader workcell, resulting in performance well within the process tolerance [87]. However, the process tolerance in the tray loader cell is ±0.2 mm, a result of the clearance around the part in the tray cavity. Automated assembly tasks are likely to require smaller process tolerances, necessitating a higher accuracy of the Direct Calibration process.

The performance of the Quadratic Basis Function set is used as a reference for any newly developed basis function sets.

4.2 Generating "Perfect" Basis Functions

To establish a performance reference point and – indeed – the performance limit for a given part, the "Perfect" Basis Functions for that part can be found and their

performance compared to the basis functions currently in use. An Perfect Basis Function (PBF) set is one that consists of the terms that are contained in the actual mathematical equations that describe the geometric relationships between a perturbation of the part in robot space and the change in its sensor values in camera space. Note that these PBF sets are highly part specific and thus would not fit the *Flexible Automation* paradigm that is integral to the Direct Calibration idea. Perfect Basis Functions do, however, provide valuable insight into the requirements of each part type on the set of basis functions that are actually selected for use in a Direct Calibration workcell.

Typical parts used with Direct Calibration have at least one planar side with sharp edges, which works well with structured light sensors. Laser lines break over edges of the part and the locations of the endpoints of these lines in image space are found. The geometric relationship between part perturbations and change in position of the ends of the laser lines can be developed analytically for simple part geometries. The feature locations then are transferred into camera pixel space, and the resulting idealized part model is used with a set of part perturbations to generate sensor data in simulation. Part perturbation sets and their corresponding sensor values can now be used to evaluate different sets of basis functions for this "simulated" part. The sensor equations are the exact relationships between sensor values and part location. The Direct Calibration process approximates these relationships with linear combinations of the basis functions specified. Hence, the Perfect Basis Functions for a given part are the

terms of the sensor equations involving the perturbation variables $[X, Y, Z, \Theta]$. These "Perfect" Basis Functions are developed for different part models and compared in the part simulations to the basis functions currently used.

While the Perfect Basis Functions can be compared to other sets of basis functions in simulation as described, they need to be validated with real parts in a real-world production process. Sensor noise, edge effects on the parts, and other non-idealities impact the success of basis functions in the real system. Thus, real production parts are used to compare the performance of the developed basis functions to the performance of Quadratic Basis Functions in the real tray loader system.

4.3 Developing Perfect Basis Functions for a Simple Part Configuration

As a simple case, a rectangular flat plate is considered. The parallel jaw gripper constrains the part to perturbations (part-in-hand errors) in a plane, thus reducing the potential part-in-hand error to three dimensions: two translations and a rotation.

A distinction is made based on whether the grip point is a web parallel or perpendicular to the plane of the part as shown in Figure 4-3.

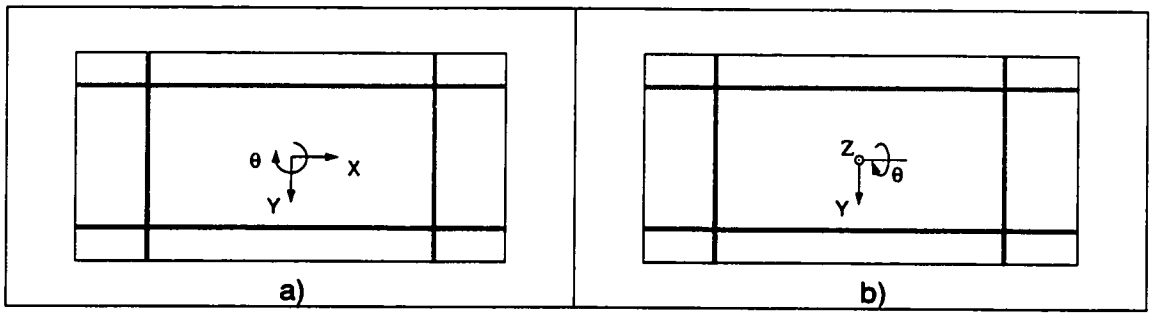


Figure 4-3: Rectangular part a) constrained to move in the plane of the part b) constrained to move in a plane normal to the part

The locations of the intersections of the laser lines projected onto the part and the part edges are the sensor values. For simple configurations, the geometrical relationship between these sensor values and the part location can be developed using simple trigonometry and the Perfect Basis Functions for this configuration extracted from the equations relating sensor values to part perturbation. The performance of these new basis functions can then be compared to the "Quadratic" Basis Functions in simulation and on real parts. This flat plate model represents the majority of parts fed in real world applications. Which side of the part is grasped by the robot and which side is exposed to the sensing system is often suggested by the part geometry, but the ultimate decision is with the engineer who trains the Direct Calibration system with the part.

4.3.1 Translation and Rotation in the Plane

The analytical relationship between the perturbation [X, Y, Θ] and the sensor values is readily determined when the laser lines are perpendicular to the part

edges and the part only moves in the plane perpendicular to the camera normal as shown in Figure 4-4.

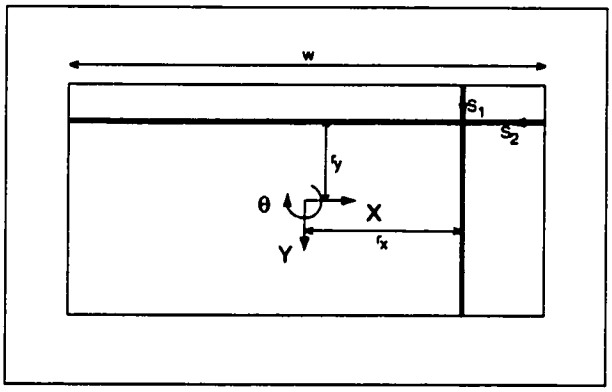


Figure 4-4: The part constrained to the plane and two of the laser lines

The equations for the two sensor values shown are found to be

$$s_1 = Y + (r_x - X)\tan(\Theta) - \left(\frac{1}{\cos(\Theta)} - 1\right)w, \text{ and}$$

$$s_2 = -X - (r_y + Y)\tan(\Theta) - \left(\frac{1}{\cos(\Theta)} - 1\right)w$$

Since the part moves only in the plane perpendicular to the camera normal, the x-coordinate of sensor 1 and the y-coordinate of sensor 2 do not change in the camera frame. The equations for the other sensors on the part are analogous.

These equations are used as an idealized model of the part. A set of random part perturbations over the range of expected part-in-hand errors can be used to generate simulated sensor data. The generated data and the part model then can

be used to evaluate different sets of basis functions for the simulated part by comparing the part locations computed by the Direct Calibration routines to the known random perturbations.

For this simple part, the Perfect Basis Functions are the terms involving the perturbation variables [X, Y, Θ] in the equations for s_1 and s_2 above, while the coefficients of these terms are the coefficients to be determined by the Direct Calibration training process. The basis functions are shown in Table 4-1.

Table 4-1: Perfect Basis Functions for a plate part moving in a plane perpendicular to the camera normal.

1	X	$X\tan(\Theta)$	$\frac{1}{\cos(\Theta)}$	$\tan(\Theta)$
	Y	$Y\tan(\Theta)$		

Simulated Part

These Perfect Basis Functions can be tested in simulation by using the equations that relate sensor values to part perturbation as a part model as described above. The model is used to generate the training data set from a uniform grid of part perturbations and the evaluation data set from a set of 500 random perturbations in the calibration range.

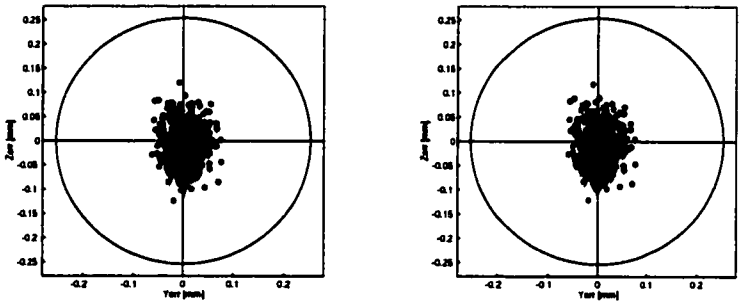
For this simulated part, the training process found the correct coefficients of the basis functions perfectly, resulting in a perfect model of the part. Also the evaluation error is zero, meaning the model fit the random evaluation data

perfectly. These results were of course expected since both the basis functions and the data come from the same model, but these results do serve as a good check of the validity of the data generation and simulation code.

To test the robustness of the system using different sets of basis functions with the simulated part, noise was added to the simulated sensor data. This noise simulates the sensor noise and the robot repeatability error of the real system and is randomly sampled from a uniform distribution and scaled to a pixel error value of a scale similar to real part evaluations. The same random noise was used for each set of basis functions to insure comparable results.

For this noisy data, the Perfect Basis Functions showed slightly better performance than the Quadratic Basic Functions as seen in Table 4-2.

Table 4-2: Part-in-hand error [mm] transferred to the part extreme for the simulated part with noise added.



	Quadratic BF	Perfect BF
Average Error	0.00537	0.00495
Std. Deviation	0.04730	0.04729
Max. Error	0.12540	0.12286

This limited improvement suggests that the Quadratic Basis Functions are fairly adequate for this part. In fact, representing the $\tan(\Theta)$ and $1/\cos(\Theta)$ terms by the first few terms of their Taylor series expansions and comparing terms of the Perfect and Quadratic Basis Function sets reveals that the only terms not covered by the simpler method are some higher order terms that have only a small impact on the mapping. Refer to Table 4-3.

Table 4-3: Basis function terms for a flat part constrained to a plane perpendicular to the camera normal.

Perfect BF	X	Y	Θ	Θ^3	$X\Theta$	$X\Theta^3$	$Y\Theta$	$Y\Theta^3$	Θ^2	1
Quadr. BF	X	Y	Θ		$X\Theta$		$Y\Theta$		Θ^2	1

Test Part

To validate these results on a real part, a test part was constructed that consists of a rectangular plate with a grip surface that is parallel to the plate. This geometry causes any part-in-hand errors to lie in a plane parallel to the plate. The test part is shown in Figure 4-5.

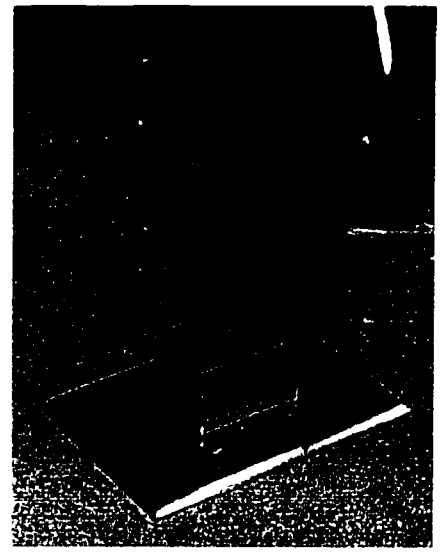
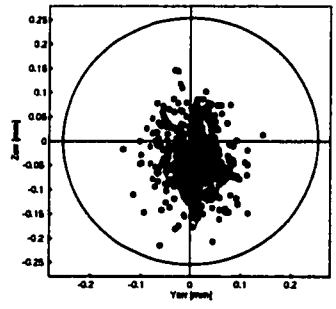
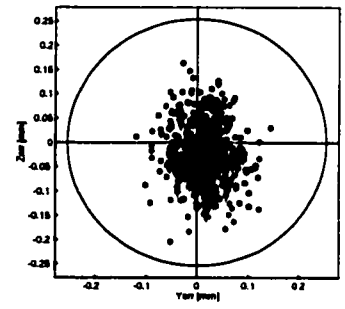


Figure 4-5: In-plane test part in the grasp of the gripper

The Quadratic and Perfect Basis Functions then were compared on this real plate part. Table 4-4 shows the results.

Table 4-4: Part-in-hand error [mm] transferred to the part extreme for the test part.





	Quadratic BF	Perfect BF
Average Error	0.0344	0.0225
Std. Deviation	0.0811	0.0780
Max. Error	0.2230	0.2107

Printer Parts

Two of the printer parts currently being used in the tray loader cell have their grip surface parallel to the sensed surface. They can thus be used as "real world" test parts for this case. They are used as an ultimate test of the basis functions generated from the analytical equations. Figure 4-6 shows the two parts.

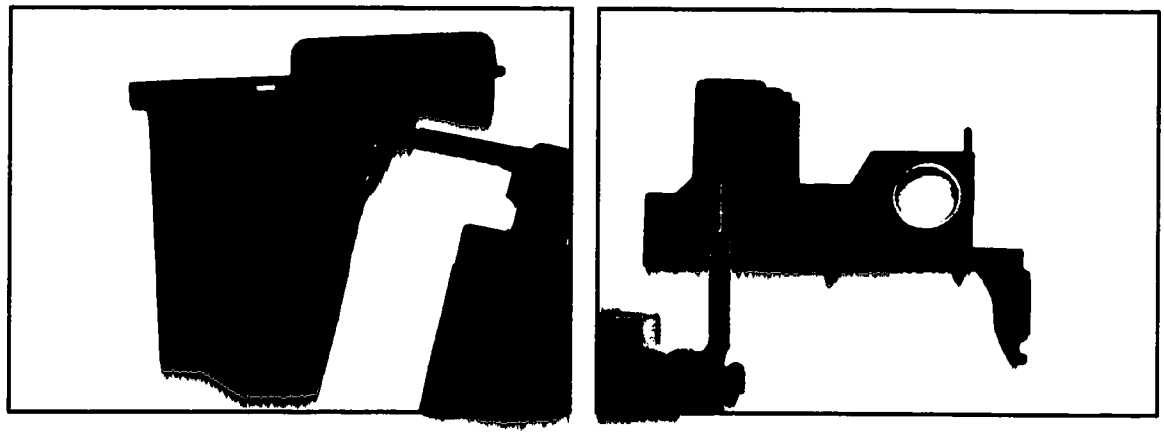
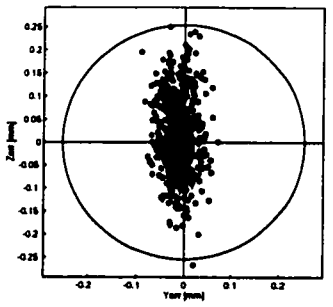
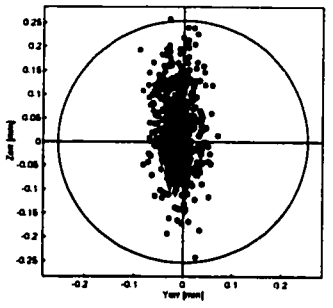


Figure 4-6: Printer parts A) and B) with gripper and axes

Table 4-5 and Table 4-6 show the results for Quadratic and Perfect Basis Function for the two parts.

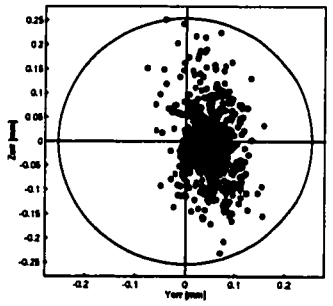
Table 4-5: Part-in-hand error transferred to the part corner for printer part A

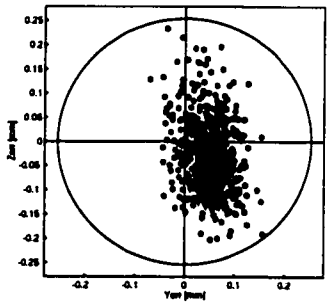




	Quadratic BF	Perfect BF
Average Error	0.0280	0.0387
Std. Deviation	0.0920	0.1003
Max. Error	0.2669	0.2592

Table 4-6: Part-in-hand error transferred to the part corner for printer part B





	Quadratic BF	Perfect BF
Average Error	0.0483	0.0508
Std. Deviation	0.1073	0.1057
Max. Error	0.2550	0.2471

For all three cases, the simulated part, the test part, and the printer part, the Perfect Basis Functions resulted in performance improvements when the

maximum error is used as the performance metric. These improvements are rather small. As previously discussed, this limited improvement is due to fact that the Perfect Basis Functions are very well represented by the Quadratic Basis Functions. For the printer parts, the average error actually increased slightly. This increase is attributed to the higher order terms in the Perfect Basis Function set amplifying the noise that is present in the sensor data of the real parts.

4.3.2 Translation and Rotation out of the Plane

When the rectangular flat part is grasped on a web perpendicular to the part plane, it is no longer constrained to move only in the plane perpendicular to the camera normal, as shown in Figure 4-7.

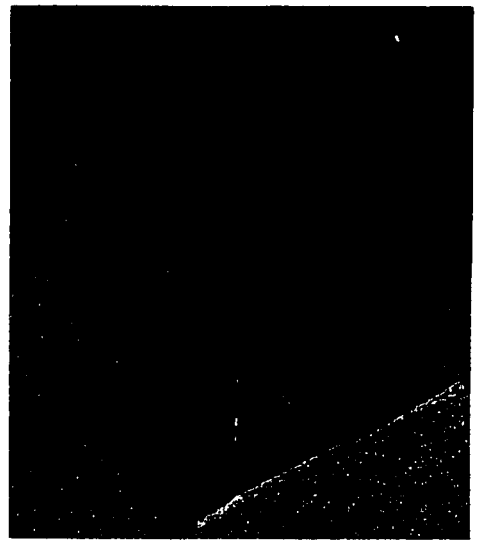


Figure 4-7: Out-of-plane test part

Determining the relationship between part perturbation and sensor values for a part where the grip surface is not parallel to the sensed plane is more involved than the case previously discussed.

A better way to model the system is to employ the spatial descriptions and transformations used in robotics theory [31]. Here, coordinate systems are attached to the components of the system and the transformations between the different coordinate systems established. The intersections of the laser planes and the part edges in the camera coordinate system then can be determined.

In this approach, the geometric relationship of the edge, the part, the laser plane, the camera, and the part perturbation are described in terms of transformation matrices [31]. The position and orientation of a body is described by a *frame* that consists of a cartesian coordinate system attached to the body and a position vector that locates the origin of the coordinate system relative to some other embedding frame. The translation from one frame to another involves a translation and a rotation that can be cast into the form of a 4x4 matrix called a *homogeneous transformation:*

$$Hmat(X,Y,Z,\alpha,\beta,\gamma) = \begin{bmatrix} c_\alpha \cdot c_\beta & c_\alpha \cdot s_\beta \cdot s_\gamma - s_\alpha \cdot c_\gamma & c_\alpha \cdot s_\beta \cdot c_\gamma + s_\alpha \cdot s_\gamma & X \\ s_\alpha \cdot c_\beta & s_\alpha \cdot s_\beta \cdot s_\gamma + c_\alpha \cdot c_\gamma & s_\alpha \cdot s_\beta \cdot c_\gamma - c_\alpha \cdot s_\gamma & Y \\ -s_\beta & c_\beta \cdot s_\gamma & c_\beta \cdot c_\gamma & Z \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where

$$c_\alpha = \cos(\alpha), \quad c_\beta = \cos(\beta), \quad c_\gamma = \cos(\gamma)$$

$$s_\alpha = \sin(\alpha), \quad s_\beta = \sin(\beta), \quad s_\gamma = \sin(\gamma).$$

The general sensor/part model geometry is illustrated in Figure 4-8 with the part in its unperturbed home position, and in Figure 4-9 with a perturbed part. Frames are denoted as {*frame*} and homogeneous transformations are shown as ${}^{from\_frame}_{to\_frame}T$.

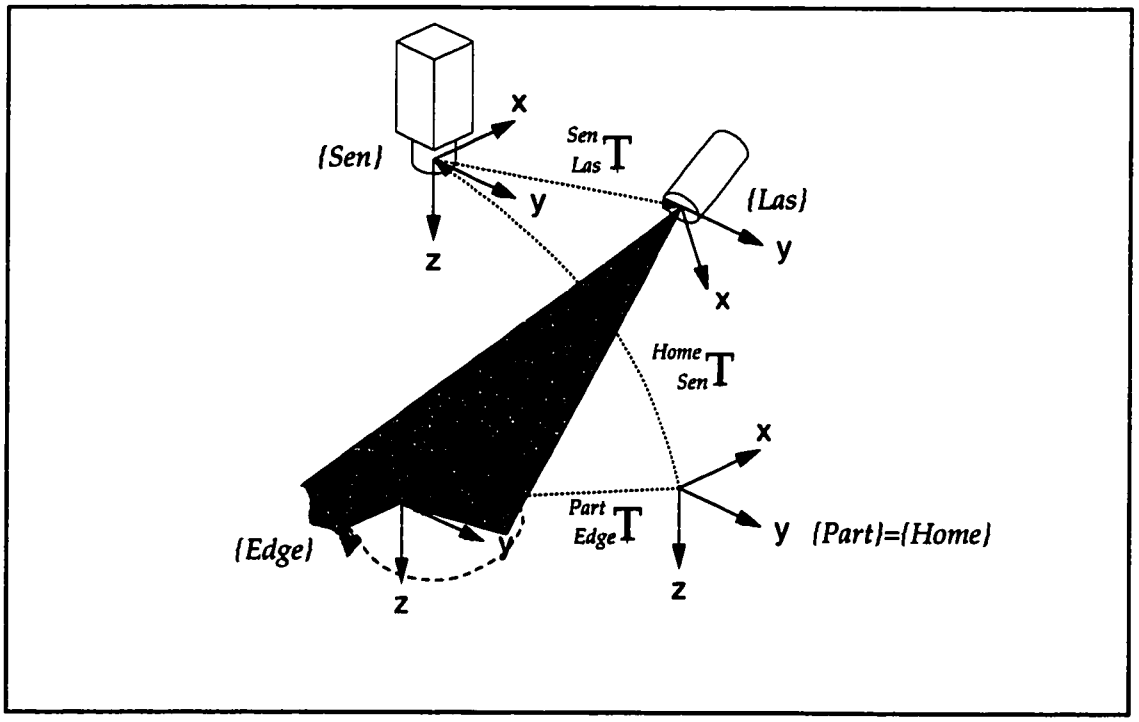


Figure 4-8: The general 6-DOF sensor/part model (unperturbed part)

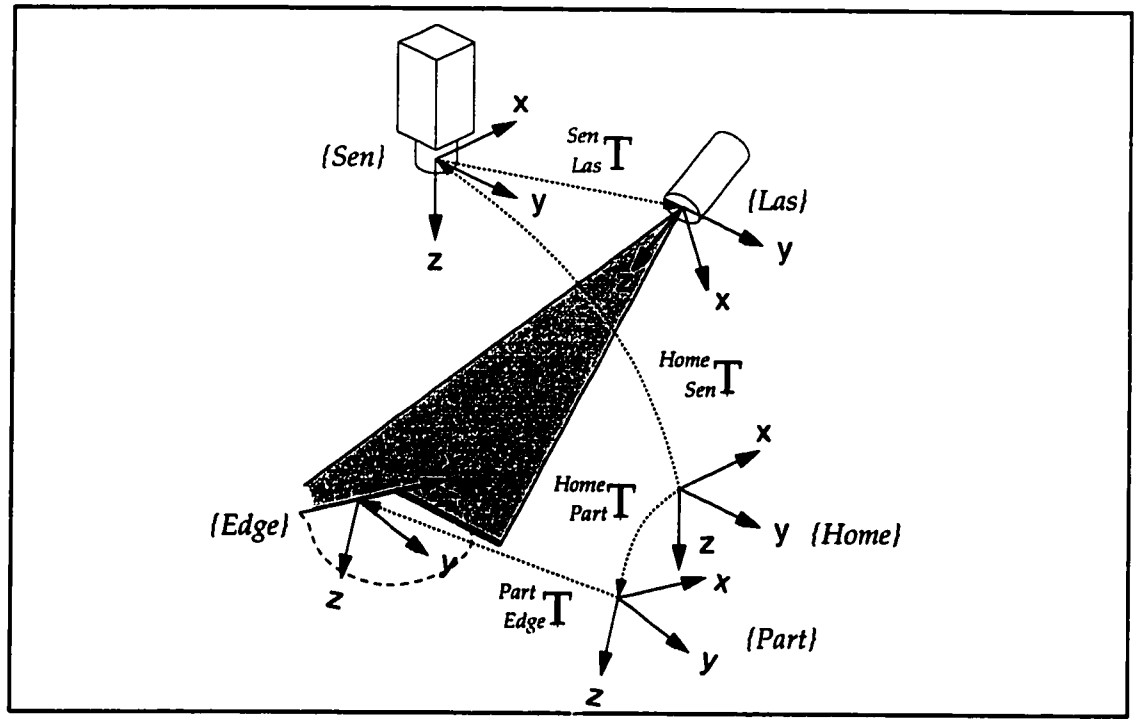


Figure 4-9: The general 6-DOF sensor/part model (perturbed part)

To determine the basis functions, the part edge is described by a parametric equation and transferred into the sensor (camera) frame. The laser plane is represented as a row vector [82] and also translated into the sensor frame. The line equation then is solved for the parameter for which the edge intersects the laser plane. Transferring this point into the camera pixel space generates the analytical equations.

The model for a part moving in Y, Z and θ_X is developed below. Only the model for a sensor point on a "horizontal" edge of the part is shown.

Transformation {Home} to {Part}

The transformation from the {Home} to the {Part} frame corresponds to the part perturbation or part-in-hand error. Since the gripper constrains the part to three degrees of freedom, the perturbation variables are Y, Z, and θ (or more specifically θ_X). The homogeneous transformation is then

$$ {}^{Home}_{Part}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\Theta) & -\sin(\Theta) & Y \\ 0 & \sin(\Theta) & \cos(\Theta) & Z \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

Transformation {Part} to {Edge}

In the {Part} frame, the {Edge} frame can be conveniently defined such that its origin is the point where the laser line intersects the part edge on the unperturbed part (refer to Figure 4-8). The transformation is then simply an offset of r_x in the x-direction, and an offset of h in the y-direction as seen in Figure 4-10.

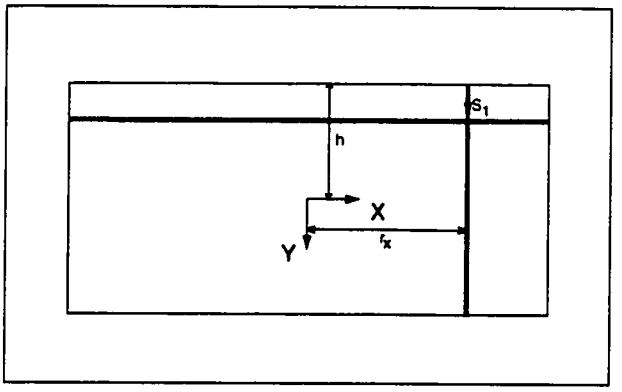


Figure 4-10: Rectangular test part with two laser lines shown

The transformation becomes

$$ {}^{Part}_{Edge}T = \begin{bmatrix} 1 & 0 & 0 & r_x \\ 0 & 1 & 0 & h \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

Transformation {Sen} to {Home}

The {Sen} frame is attached to the camera and is defined such that its origin is directly above the origin of the {Home} frame (refer to Figure 4-8). The transformation is then just a vertical offset *sh* from the {Part} frame:

$$ {}^{Sen}_{Home}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & sh \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

Transformation {Sen} to {Edge0}

The transformation from the camera frame to the unperturbed edge frame is a concatenation of the two previous transformations,

$$ {}^{Sen}_{Edge0}T = {}^{Sen}_{Home}T\,{}^{Part}_{Edge}T . $$

Transformation {Sen} to {Las}

Since it is inconsequential where the origin of the {Las} frame is, as long as it is on the laser plane, it is conveniently defined to be at the intersection of the laser plane and the unperturbed edge. Thus, the {Las} frame is coincident with the {Edge0} frame except for the laser plane inclination angle β. The transformation is thus

$$ {}^{Sen}_{Las}T = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & {}^{Sen}_{Edge0}T[1,4] \\ 0 & 1 & 0 & {}^{Sen}_{Edge0}T[2,4] \\ -\sin(\beta) & 0 & \cos(\beta) & {}^{Sen}_{Edge0}T[3,4] \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

Representation of the Edge

The edge is a straight line in space and can be defined by two points on the line. Two points u and v on the edge are chosen. Their homogeneous representation in the {Edge} frame is:

$$u_E = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

and

$$v_E = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

With the transformations developed above, the two points can be expressed in the sensor frame:

$$u_S = {}^{Sen}_{Home}T\,{}^{Home}_{Part}T\,{}^{Part}_{Edge0}T \cdot u$$

and

$$v_S = {}^{Sen}_{Home}T\,{}^{Home}_{Part}T\,{}^{Part}_{Edge0}T \cdot v.$$

Representation of the Laser Plane

The laser plane in the {Laser} frame is simply a plane that has its normal along the x-axis of the {Laser} frame,

$$L_L = [1 \quad 0 \quad 0 \quad 0].$$

In the {Sen} frame, the laser plane is

$$L_S = L_L \cdot INV({}^{Sen}_{Las}T).$$

Intersection of Edge and Laser Plane

The edge and the laser line intersect when the following condition holds:

$$L_S\{u_S + t(v_S - u_S)\} = 0.$$

When this equation is solved for t, the location of the sensed point in sensor space is

$$\begin{bmatrix} X_{Sen} \\ Y_{Sen} \\ Z_{Sen} \end{bmatrix} = u_S + t(v_S - u_S).$$

Transformation to Camera Pixel Space

The location of the sensed point then is converted to camera pixel space,

$$x = \left(\frac{X_{Sen}}{Z_{Sen}} \right) \cdot Scale$$

$$y = \left(\frac{Y_{Sen}}{Z_{Sen}} \right) \cdot Scale.$$

The equations for the vertical edges are generated in a similar manner. After the equations that describe the locations of the laser line end in camera space as a function of part perturbation in robot space are developed, the basis function terms can be extracted from the equations. While the model developed above is for a part moving in Y, Z and θ_X, the model for a part moving in X, Z, and θ_Y yields the same basis functions (except for the change in perturbation variables) and hence does not need to be treated separately. Table 4-7 shows the Perfect Basis Functions for the out-of-plane moving plate.

Table 4-7: "Perfect" Basis Functions for the out-of-plane moving plate.

$\frac{1}{y_l \sin(\Theta) + Z + sh}$	$\frac{1}{-y_l \sin(\Theta) + Z + sh}$	1
$\frac{\sin(\Theta)}{y_l \sin(\Theta) + Z + sh}$	$\frac{\sin(\Theta)}{-y_l \sin(\Theta) + Z + sh}$	
$\frac{\cos(\Theta)}{y_l \sin(\Theta) + Z + sh}$	$\frac{\cos(\Theta)}{-y_l \sin(\Theta) + Z + sh}$	
$\frac{Y}{y_l \sin(\Theta) + Z + sh}$	$\frac{Y}{-y_l \sin(\Theta) + Z + sh}$	
$\frac{Z}{y_l \sin(\Theta) + Z + sh}$	$\frac{Z}{-y_l \sin(\Theta) + Z + sh}$	
$\frac{\sin(\Theta)}{\cos(\beta)\sin(\Theta)(-Y + y_l) + \cos(\beta)\cos(\Theta)(Z + sh) + sh\sin(\beta)\sin(\Theta)}$		
$\frac{\cos(\Theta)}{\cos(\beta)\sin(\Theta)(-Y + y_l) + \cos(\beta)\cos(\Theta)(Z + sh) + sh\sin(\beta)\sin(\Theta)}$		
$\frac{Y\sin(\Theta)}{\cos(\beta)\sin(\Theta)(-Y + y_l) + \cos(\beta)\cos(\Theta)(Z + sh) + sh\sin(\beta)\sin(\Theta)}$		
$\frac{Z\cos(\Theta)}{\cos(\beta)\sin(\Theta)(-Y + y_l) + \cos(\beta)\cos(\Theta)(Z + sh) + sh\sin(\beta)\sin(\Theta)}$		
$\frac{\sin(\Theta)}{\cos(\beta)\sin(\Theta)(-Y - y_l) + \cos(\beta)\cos(\Theta)(Z + sh) + sh\sin(\beta)\sin(\Theta)}$		
$\frac{\cos(\Theta)}{\cos(\beta)\sin(\Theta)(-Y - y_l) + \cos(\beta)\cos(\Theta)(Z + sh) + sh\sin(\beta)\sin(\Theta)}$		
$\frac{Y\sin(\Theta)}{\cos(\beta)\sin(\Theta)(-Y - y_l) + \cos(\beta)\cos(\Theta)(Z + sh) + sh\sin(\beta)\sin(\Theta)}$		
$\frac{Z\cos(\Theta)}{\cos(\beta)\sin(\Theta)(-Y - y_l) + \cos(\beta)\cos(\Theta)(Z + sh) + sh\sin(\beta)\sin(\Theta)}$		

Where:

Y, Z, Θ is the part perturbation,

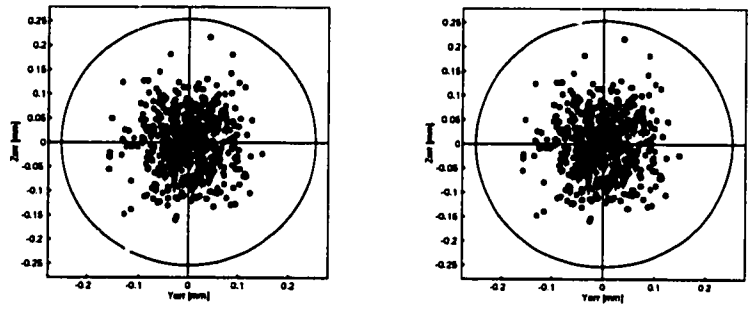
y_l is distance between laser line and x-axis,

sh is the distance between camera and part in the home position, and

β is the angle between the laser plane and the part in the home position.

Unfortunately, these Perfect Basis Functions contain part-specific information so that they are truly applicable only to the simulated part. However, the simulated part can be used as a "best case" test. The performance of these basis functions compared to the Quadratic Basis Functions for this part will show the highest possible performance improvement for the simulated rectangular part moving out of the plane. The results are shown in Table 4-8.

Table 4-8: Part-in-hand error [mm] transferred to the part extreme for the simulated rectangular part moving out of the plane with noise added.



	Quadratic BF	Perfect BF
Average Error	0.002115	0.002113
Std. Deviation	0.089990	0.090017
Max. Error	0.220856	0.220815

The expected significant difference between the two sets of basis functions for the part moving out of the plane did not materialize. This result surprisingly suggests that the Quadratic Basis Functions adequately represent the complex terms in Table 4-7. To investigate this, it is desirable to break down the Perfect

Basis Functions terms into smaller terms than can be compared to the Quadratic Basis Functions.

By factoring the basis function terms so that parts of them can be cast into the form of the binomial series:

$$\frac{1}{1+\varepsilon} = 1 - \varepsilon + \varepsilon^2 - \varepsilon^3 + \ldots \quad \text{for } |\varepsilon| < 1,$$

and using a small-angle approximation for the trigonometric terms, the terms in Table 4-7 can expanded and sorted into their elementary terms. Table 4-9 shows the second- and third-order terms of the expansions of the basis function members and how they compare to the Quadratic Basis Function (QBF) set.

Table 4-9: Comparison of the terms of the Taylor series expansion of the terms in Table 4-7 to the Quadratic Basis Function terms (QBF).

QBF Terms	Basis Function Term							
	1	2	3	4	5	6	7	8
1	1					1		
Y			Y					
Z	Z			Z		Z		Z
Θ	Θ	Θ			Θ	Θ		
Y^2						Y^2		
YZ			YZ					
$Y\Theta$			$Y\Theta$			$Y\Theta$	$Y\Theta$	
Z^2	Z^2			Z^2		Z^2		Z^2
$Z\Theta$	$Z\Theta$	$Z\Theta$		$Z\Theta$	$Z\Theta$	$Z\Theta$		$Z\Theta$
Θ^2	Θ^2	Θ^2						
			YZ^2					
			$YZ\Theta$				$YZ\Theta$	$YZ\Theta$
			$Y\Theta^2$		$Y\Theta^2$	$Y\Theta^2$		
	Z^3							Z^3
	$Z^2\Theta$	$Z^2\Theta$		$Z^2\Theta$	$Z^2\Theta$			$Z^2\Theta$
	$Z\Theta^2$	$Z\Theta^2$		$Z\Theta^2$	$Z\Theta^2$			$Z\Theta^2$
	Θ^3	Θ^3			Θ^3			

Table 4-9 illustrates how well the rather complex basis function terms of Table 4-7 are approximated by the simple terms of the Quadratic Basis Function set in Table 4-1. Only eight terms are shown, since the remaining terms break down identical to the ones shown. All first- and second-order terms in columns 2 through 9 of Table 4-9 are covered by the Quadratic Basis Function set. The third

order terms shown in the table contribute only minimally as their coefficients are very small in the Taylor Series expansions.

While the very limited performance improvement could be seen as disappointing, the real value of this discovery is the acquired certainty that the Quadratic Basis Function set is a great choice for a general application of Direct Calibration. The next chapter examines whether the Quadratic Basis Function set can be improved upon.

CHAPTER 5: VARIATIONS OF THE QUADRATIC BASIS FUNCTION SET

After the validity of the Quadratic Basis Function set was demonstrated in the previous chapter, this chapter investigates potential performance improvements from modifying the QBF set. Adding more terms to the set has the potential for performance improvement, while reducing the number of terms in the set may make the method more efficient and also may aid in understanding the underlying relationship between basis functions and process performance.

5.1 Cubic Basis Functions

In the previous analysis into the elementary terms of the Perfect Basis Function sets for both the in-plane and out-of-plane cases, the Quadratic Basis Functions (QBF) approximated the Perfect Basis Function (PBF) sets well since the QBF terms correspond to the lowest order terms of the PBF sets. The use of the QBF sets also resulted in performance comparable to that of the PBF sets in simulation. As can be seen from the breakdown into elementary terms in the previous chapter, the PBF sets can be matched even closer by a Cubic Basis Function (CBF) set that includes the QBF set but adds third-order combinations of the perturbation variables as shown in Table 5-1.

Table 5-1: The CBF set

X	X^2	X^3	XY	XY^2	X^2Y	$XY\theta$	1
Y	Y^2	Y^3	$Y\theta$	$Y\theta^2$	$Y^2\theta$		
θ	θ^2	θ^3	θX	θX^2	$\theta^2 X$		

The effect of including the third-order terms is evaluated first in simulation for the part that is constrained to the plane (*in-plane* or *IP* case) and the part moving out of the plane (*out-of-plane* or *OOP*). Then, the effect on real production parts is investigated.

5.1.1 Simulated Part

The QBF and CBF sets are compared on both the simulated in-plane moving (IP) part and the out-of-plane (OOP) moving part. To investigate the effect of noise in the data on the performance difference between the two different basis function sets, three levels of added noise are used. The change in performance is analyzed in terms of mean error and maximum error. Of interest is especially whether the change in performance is dependent on the level of noise in the data. It had previously been supposed [87] that the effect on process performance of noise in the data is greater for the CBF method than for the QBF method. Table 5-2 shows the test results.

Table 5-2: Change in CBF process error when compared to QBF for the IP and OOP parts at three levels of noise

	Noise Level		
	Low	Med.	High
IP mean error	+8%	+5%	+7%
IP max. error	+3%	+2%	+7%
OOP mean error	+5%	+5%	+7%
OOP max. error	+11%	+11%	+12%

For both the IP part and the OOP part, the performance degrades for all three levels of noise. Both the mean error and the maximum error increase while the change in error between QBF and CBF is not found to depend greatly on the noise level.

The results in Table 5-2 suggest that in general, both the QBF and CBF methods are robust to noisy data. Regarding the notion mentioned above that the CBF set might be more sensitive to noisy data, the rather uniform degradation of performance between QBF and CBF for different levels of noise shows that this is not the case. The poor performance of the CBF method are obviously due to other causes.

To further investigate the cause of the performance degradation, the process by which the mapping is learned is examined. The training error is the difference between the true sensor reading and the sensor reading calculated by running

the true location through the learned mapping. It is a measure of how well the mapping fits the training data. The training error of all 125 calibration points is analyzed separately for each of the sensors for a part. Thus, eight distinct error distributions are compared for each case. Table 5-3 shows the results of these comparisons qualitatively. The OOP part has twice as many sensor data points since its laser line end points move along the direction of the part edge as well as normal to it.

Table 5-3: Change in CBF training error compared to QBF for the IP and OOP parts at three levels of noise (Notation: + increase, - decrease)

	Sensor Number															
	1		2		3		4		5		6		7		8	
IP mean error	-		-		-		-		-		-		-		-	
IP max. error	+		-		-		-		-		+		+		+	
OOP mean error	-	-	-	-	-	+	-	+	-	-	-	-	-	-	-	-
OOP max. error	+	-	-	-	-	+	+	+	-	+	+	+	-	-	-	-

For the maximum error, about half of the sensors show an increase with the other half showing a decrease, regardless of which basis function set is used for both IP and OOP parts. It is interesting to note that the distribution of sensors that show a similar change does not correspond to their geometric distribution on the parts.

The average error shows a decrease for all sensors on the IP part and for 88% of the sensors on the OOP part. Increasing the level of noise does not change this

behavior since noisy data is no harder to learn than exact data. So overall, Table 5-3 shows a trend of improved training for the CBF method, suggesting that the training data fit the CBF mapping well. Since the overall error increased, but the training error decreased, the source of the poor performance must be a poor fit of the mapping to the evaluation data, which falls on the model between the training data. In an analogy to curve fitting, a higher order curve is more likely to fit the data points accurately, but may exhibit many maxima and minima between points that do not correspond to the nature of the data. A likely scenario is thus that more training data points are needed to properly "tie down" the more flexible higher-order CBF model. To test this theory, Table 5-4 shows a comparison of performance resulting from four different training data sets with medium noise on the IP part, while Table 5-5 shows the same comparison using a high noise level. Table 5-6 and Table 5-7 show the corresponding results for the OOP part.

Table 5-4: Change in CBF process error compared to QBF for the IP part with a medium noise level

Training Data Size	Change in Mean Error	Change in Max. Error
125	+12%	+12%
343	-1%	+1%
729	+3%	+3%
2197	-2%	-1%

Table 5-5: Change in CBF process error compared to QBF for the IP part with a high noise level

Training Data Size	Change in Mean Error	Change in Max. Error
125	+8%	+11%
343	-2%	+1%
729	+3%	+3%
2197	-2%	-1%

Table 5-6: Change in CBF process error compared to QBF for the OOP part with a medium noise level

Training Data Size	Change in Mean Error	Change in Max. Error
125	+5%	+11%
343	-2%	+6%
729	+1%	+1%
2197	-27%	-1%

Table 5-7: Change in CBF process error compared to QBF for the OOP part with a high noise level

Training Data Size	Change in Mean Error	Change in Max. Error
125	+7%	+12%
343	-2%	+5%
729	+1%	+1%
2197	-23%	-1%

The four tables show that increasing the training data set size does indeed close the margin between the two methods, but that a training set of 13 x 13 x 13 points is required before the CBF set exhibits superior performance.

The peculiar increase in error between the 343-point set and the 729-point set that is seen in some of the cases above is due to the fact that the smaller set is not a subset of the larger set. The set with 729 (9 x 9 x 9) points does not include any of the points in the 343 (7 x 7 x 7) set other than the midpoint and endpoints. The increase in error thus means that the smaller set contains points that have more crucial information about the true mapping. As a check, the 2197-point set is a superset of the 343-point set, and the 729-point set is a superset of the 125-point set. In both cases, going from the subset to the superset, the performance improves for all instances.

This performance improvement over the QBF method is substantial in the mean of the error in the OOP case when the largest calibration set is used, while the maximum error did not change appreciably. Depending on the nature of the production process that the Direct Calibration method is used in, mean error or maximum error can be the critical metric. In a tray-filling operation, an extreme outlier may result in an empty tray pocket, while a higher average error would cause a low tray fill rate, resulting in decreased cycle time. In an assembly process, however, an outlier may cause a failed assembly attempt with resulting fatal damage to the whole assembly. The usefulness of the CBF over QBF thus depends on the nature of the process. Another potential deciding factor is the

time and effort required to collect a training data set that is almost 20 times as large as the one currently in use.

5.1.2 Production Parts

Most of the printer parts used in the tray loader cell were calibrated using 81-point calibration sets. Since Park's [81] investigation into calibration set size, a 125-point set has been used. In this research, it was found that an 81-point set is too small to train a Cubic Basis Function set. The resulting performance is worse by several orders of magnitude compared to the quadratic set. For this investigation, the calibration sets with 125 and 729 points were compared since they were available from Park's research. Larger sets could not be generated since at the time of writing the experimental setup was no longer available. The results for three different printer parts are shown in Table 5-8 through Table 5-10.

Table 5-8: Change in CBF process error compared to QBF for printer part A

Training Data Size	Change in Mean Error	Change in Max. Error
125	+11.3%	+18%
729	+11.0%	+6%

Table 5-9: Change in CBF process error compared to QBF for printer part B

Training Data Size	Change in Mean Error	Change in Max. Error
125	-0.05%	+0.7%
729	-0.1%	-3%

Table 5-10: Change in CBF process error compared to QBF for printer part C

Training Data Size	Change in Mean Error	Change in Max. Error
125	+2%	+55%
729	-1%	+35%

The data available confirms that, as in simulation, larger calibration sets are required for the CBF set. The 81-point calibration sets resulted in completely unacceptable performance, while for the 125-point sets, the CBF performance is getting closer to the QBF performance. Increasing the calibration set size to 729 points improves performance even more. However, the three cases above show that different parts require calibration sets of different size before the CBF method shows an improvement in performance. For Part A, even when using a 729-point calibration set, the QBF method shows substantially better performance than the CBF method. Part B shows virtually identical performance using either method, even with the 125-point set. Increasing the size of the calibration set results in a greater benefit of using the CBF set. The results for

Part C suggest that a much larger calibration set is required to make the CBF set work as well as the QBF method.

In summary, while a Cubic Basis Function set has the potential for a small performance improvement, the cost is just too great for the general application of the Direct Calibration method. To realize any improvement in performance, huge calibration sets are required. In certain situations, where maximum performance is required and the cost of acquiring large calibration sets is not an issue, this may be worthwhile. However, it must be realized that changes to the layout or the configuration of the cell, intended or unintended, require taking new calibration sets.

5.2 Reducing the Size of the Quadratic Basis Function Set

Since the broad applicability of the Quadratic Basis Function set has now been demonstrated, it is interesting to investigate if this set can be at all reduced without sacrificing performance. In other words, are all of the terms of the set significant or can certain terms be left out?

Each term of the set is removed in turn, and the performance of the system evaluated. Table 5-11 shows the results for the simulated OOP part and three real production parts used in the tray loader cell.

Table 5-11: Percent change in error when one term at a time is left out of the Quadratic Basis Function Set. Improvements due to leaving out a certain terms are shaded.

	Simulated Part		Printer Part A		Printer Part B		Printer Part C	
Term left out	Average Error Change	Max. Error Change	Average Error Change	Max. Error Change	Average Error Change	Max. Error Change	Average Error Change	Max. Error Change
Y	> +100%	> +100%	Crashed	Crashed	> +100%	> +100%	> +100%	> +100%
Z	> +100%	> +100%	> +100%	> +100%	> +100%	> +100%	Crashed	Crashed
Θ	> +100%	> +100%	> +100%	> +100%	> +100%	> +100%	-49%	> +100%
YY	+34%	+1%	+26%	+1%	+10%	+1%	+7%	-1%
YZ	+1%	+1%	Crashed	Crashed	+1%	+1%	+1%	-1%
YΘ	+27%	-10%	-4%	+30%	-1%	-4%	-2%	-5%
ZZ	+11%	+6%	Crashed	Crashed	+5%	-1%	+15%	-4%
ZΘ	-8%	-4%	-3%	+16%	+1%	+0%	+1%	-3%
ΘΘ	+39%	-4%	Crashed	Crashed	+8%	+0%	-82%	-25%
1	-13%	+6%	Crashed	Crashed	+56%	+16%	-54%	-20%

If any of the rows in Table 5-11 showed an error reduction (indicated by shading) for all elements in that row, there would be hope that the basis function corresponding to the row could be omitted from the QBF set for all parts. But since each row showing an error reduction also shows a sizeable error increase elsewhere, it is clear that none of the terms can be left out of the Quadratic Basis Function set without penalty. While leaving out a term of the set can bring worthwhile performance improvements for *one* part, Table 5-11 shows that other parts rely heavily on that same basis function term. If a certain assembly operation required the highest possible accuracy, an analysis similar to the one

above could be performed and the basis function set be tailored to this one part. Note that generating this custom set would not require any special knowledge about the part such as a CAD model.

CHAPTER 6: FUTURE WORK

This chapter describes possible topics for future research based on the work presented.

6.1 Hardware Improvements

This dissertation presents research into improving the performance of the Direct Calibration method through better basis functions. The limits to performance of a general application of the Direct Calibration technique have now been established, which leaves the possible improvement of process performance through updated hardware. The experiments in the tray loader cell were performed with a robot designed for high payloads, which sacrifices accuracy. A smaller, more accurate robot would have a much higher repeatability, resulting in lower process error. The improvements possible through the use of a higher resolution camera to minimize vision errors in the final sensing stage should also be investigated.

6.2 Cubic Basis Functions for Real Parts

The performance potential of the CBF set has been sufficiently explored only in simulation. Since the largest available calibration data sets for real parts were only 9 x 9 x 9 points, the point where a performance improvement due to the

CBF set could be realized was not reached. Larger calibration sets should be explored in future work.

6.3 Constructing an Optimal Part-Specific Set

The experiments described in Section 5.2 illustrate that eliminating a certain basis function from the QBF set can be beneficial for one part, but detrimental for another part. This suggests that, if general application of the BF set is not important, a significant performance improvement can be achieved by tailoring the set to a specific part.

Finding this optimal set is straightforward in theory. For each combination of basis functions, train the Direct Calibration system with the training data set then evaluate the performance with the evaluation data set of 500 random points. The combination with the lowest error will be selected as the Optimal Basis Function set for this part.

However, the number of runs necessary makes this brute force approach infeasible. The CBF set comprises 20 basis function terms. Evaluating all combinations of these would require 2^{20} (or more than one million) runs. A method that finds the best set, but only requires a feasible number of runs, is required. Such an approach is presented in the following subsection. The method is not implemented, but enough detail is provided so that future implementation would require only minimal study of statistics theory.

6.3.1 Designing the Experiment

The approach presented here uses techniques from experimental design and statistics [59, 73, 85]. In the language of experimental design, the problem of finding the Optimal Basis Function set for one part is an experiment with the goal of finding the statistical *significance* of a number of *factors*. The factors are the 20 terms of the CBF set, each having two *levels*, 1 (include term in set) and 0 (omit term from set). Finding a factor to be significant confirms that changing the level of this factor produces a change in the measured quantity. An experiment in which the significance of more than one factor is studied is called a factorial experiment. In a *full factorial* experiment, n experimental units (or n *replicates*) are tested at each factor combination, for all combinations of the factor levels. When a full factorial experiment requires too many runs, as in this case, a *fractional factorial* experiment can be performed by evaluating only a carefully prescribed subset of all possible factor combinations. The careful choice of combinations allows an analysis of the sensitivity of the process performance to each factor with a greatly reduced number of runs. The tradeoff is that some information is lost when less than a full factorial experiment is performed. A fractional factorial *confounds* some effects with others, meaning that an observed effect can no longer be attributed to one factor alone. Fewer runs lead to more confounding of effects. Selecting the size of the experiment and which effects to confound is the "art" in designing the experiment. As a general rule, fractional factorial experiments must always be followed by a confirmation experiment

(usually of smaller size than the fractional factorial) to confirm the findings in a full factorial experiment.

A fractional factorial experiment attempts to estimate not only the effects of the factors, but also the effects of their interactions. In the basis function problem, it is possible that the importance of including a certain term changes based on whether another term is included in the set or not. The effect of interactions needs to be accounted for in a fractional factorial unless *a priori* process knowledge indicates that there are no significant interactions between factors. In the case of the basis function problem, no such knowledge exists, so effects from factor interactions should not be confounded with main factor effects.

How much confounding is taking place is defined by the *resolution* of the experiment.

Resolution III designs confound main effects with two-way interaction effects. For example, in a Resolution III experiment, a high significance of factor A cannot be distinguished from a high significance of the interaction between factors B and C. It is obvious that this type of design is only useful if there is enough process knowledge to eliminate the possibility of interactions being significant.

Resolution IV designs do not allow the confounding of main effects with two-way interactions, but do permit the confounding of main effects with three-way interactions and two-way interactions with other two-way interactions.

Resolution V designs do not allow the main effects to be confounded with two- or three-way interactions, nor do they permit any two-way interaction effect to be confounded with any other two-way interaction. Main effects are confounded with four-way interaction, and two-way interactions are confounded with three-way interaction.

Clearly, a Resolution III design will not be sufficient for the basis function problem, since there is no process knowledge that allows ruling out two-way interactions as being important. While a Resolution V design would be desirable, the large number of factors would require too many runs. A Resolution IV design does allow the confounding of main effects with three-way interactions, but it is generally assumed [59] that three-way or higher interactions, even when significant, are not likely to be important (i.e., explain a great deal of variability). A Resolution IV design is thus chosen.

Fractional factorial designs often take the shape of *orthogonal arrays* [85]. In an orthogonal array, each row represents one experimental run, while the columns correspond to the factors. The cells of the array then show the levels of all factors for each run. The size of the array is determined by the number of factors in the experiment and the desired resolution of the design.

As an example, consider an experiment to study eight factors at two levels each. A Resolution IV fractional factorial is desired so that no main effects are confounded with any interaction effects. From a set of widely published designs [85], the L_{16} orthogonal array is chosen. The number 16 indicates that sixteen

runs will have to be performed in this experiment. Together with each orthogonal array, an interaction table is generally published, that in this case permits the choice of columns for the factors such that there is no confounding between main effects and interaction effects. The resulting design is illustrated in Table 6-1. Note that the two-way interactions shown in the last row do not fall on any main effect columns.

Table 6-1: A Resolution IV fractional factorial design using a n L_{16} orthogonal array for 8 factors at 2 levels each

Run	Factor														
	A	B		C			F	E			G		H	D	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
3	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
4	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
5	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
6	1	0	0	1	1	0	0	0	0	1	1	0	0	1	1
7	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1
8	1	0	0	0	0	1	1	0	0	1	1	1	1	0	0
9	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
10	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1
11	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1
12	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0
13	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
14	0	0	1	1	0	0	1	0	1	1	0	0	1	1	0
15	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0
16	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1
2-way Interaction			AB CF EG HD		AC BF EH GD	AF BC ED GH			AE BG CH FD	AG BE CD FH		AH BD CE FG			AD BH CG FE

Conducting the experiment as prescribed in the orthogonal array of Table 6-1 results in estimates of the main effects and estimates of seven effects that confound four two-way interactions each. Prior process knowledge indicated

that two-way interactions are not important in this problem, so that only the main effects are of interest.

In the basis function problem, there are twenty factors, namely the terms of the Cubic Basis Function set (see Table 5-1). There is no process knowledge that eliminates the possibility of important interaction effects between factors, but knowledge of the geometric relationships between movement of the laser lines on the part and part perturbation reveals that the three perturbation variables definitely need to be included in an optimal set. When the part is perturbed in one degree of freedom only, it is easy to see that the resulting motion of the sensor value depends directly on the perturbation variable of that degree of freedom. Also, a constant offset term is definitely needed in the basis function set to account for constants in the geometric equations. Thus, by removing the perturbation variable terms and the constant offset term from the factors of the experiment, the number of factors is now reduced to sixteen. A Resolution IV design with sixteen factors at two levels each can be constructed from an L_{32} orthogonal array as shown in Appendix A. Using the corresponding interaction table [85], the design for the basis function experiment is developed from the L_{32} array such that the main effects are not confounded with any 2-way interactions. Analogous to the example in Table 6-1, sixteen columns are used for the factors, and fifteen columns describe confounded interaction effects. Table 6-2 illustrates which basis functions to include for each run, the first run includes all basis functions, the second run omits the second half of the basis functions, etc. The

last row of the table shows which interaction effects are confounded and which columns these groups are associated with.

Table 6-2: Orthogonal array design for the basis function problem

	Column																												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Run	Factors																														
#	A	B		C			D	E			F		G	H		I			J		K	L			M	N		O			P
1	1	1		1			1	1			1		1	1		1			1		1	1			1	1		1			1
2	1	1		1			1	1			1		1	1		0			0		0	0			0	0		0			0
3	1	1		1			1	0			0		0	0		1			1		1	1			0	0		0			0
4	1	1		1			1	0			0		0	0		0			0		0	0			1	1		1			1
5	1	1		0			0	1			1		0	0		1			1		0	0			1	1		0			0
6	1	1		0			0	1			1		0	0		0			0		1	1			0	0		1			1
7	1	1		0			0	0			0		1	1		1			1		0	0			0	0		1			1
8	1	1		0			0	0			0		1	1		0			0		1	1			1	1		0			0
9	1	0		1			0	1			0		1	0		1			0		1	0			1	0		1			0
10	1	0		1			0	1			0		1	0		0			1		0	1			0	1		0			1
11	1	0		1			0	0			1		0	1		1			0		1	0			0	1		0			1
12	1	0		1			0	0			1		0	1		0			1		0	1			1	0		1			0
13	1	0		0			1	1			0		0	1		1			0		0	1			1	0		0			1
14	1	0		0			1	1			0		0	1		0			1		1	0			0	1		1			0
15	1	0		0			1	0			1		1	0		1			0		0	1			0	1		1			0
16	1	0		0			1	0			1		1	0		0			1		1	0			1	0		0			1
17	0	1		1			0	1			0		0	1		1			0		0	1			0	1		1			0
18	0	1		1			0	1			0		0	1		0			1		1	0			1	0		0			1
19	0	1		1			0	0			1		1	0		1			0		0	1			1	0		0			1
20	0	1		1			0	0			1		1	0		0			1		1	0			0	1		1			0
21	0	1		0			1	1			0		1	0		1			0		1	0			0	1		0			1
22	0	1		0			1	1			0		1	0		0			1		0	1			1	0		1			0
23	0	1		0			1	0			1		0	1		1			0		1	0			1	0		1			0
24	0	1		0			1	0			1		0	1		0			1		0	1			0	1		0			1
25	0	0		1			1	1			1		0	0		1			1		0	0			0	0		1			1
26	0	0		1			1	1			1		0	0		0			0		1	1			1	1		0			0
27	0	0		1			1	0			0		1	1		1			1		0	0			1	1		0			0
28	0	0		1			1	0			0		1	1		0			0		1	1			0	0		1			1
29	0	0		0			0	1			1		1	1		1			1		1	1			0	0		0			0
30	0	0		0			0	1			1		1	1		0			0		0	0			1	1		1			1
31	0	0		0			0	0			0		0	0		1			1		1	1			1	1		1			1
32	0	0		0			0	0			0		0	0		0			0		0	0			0	0		0			0
		AB		AC	AD			AE	AF		AG			AH		AI	AJ		AK			AL	AM			AN			AO	AP	
		CD		BD	BC			BF	BE		BH			BG		BJ	BI		BL			BK	BN			BM			BP	BO	
		EF		EG	EH			CG	CH		CE			CF		CK	CL		CI			CJ	CO			CP			CM	CN	
2-way		GH		FH	FG			DH	DG		DF			DE		DL	DK		DJ			DI	DP			DO			DN	DM	
Int.		IJ		IK	IL			IM	IN		IO			IP		EM	EN		EO			EP	EI			EJ			EK	EL	
		KL		JL	JK			JN	JM		JP			JO		FN	FM		FP			FO	FJ			FI			FL	FK	
		MN		MO	MP			KO	KP		KM			KN		GO	GP		GM			GN	GK			GL			GI	GJ	
		OP		NP	NO			LP	LO		LN			LM		HP	HO		HN			HM	HL			HK			HJ	HI	

6.3.2 Analysis

Which factors are significant is determined in an analysis of variance (ANOVA). An ANOVA tests the hypothesis that several population means are the same based on a sample from each population. Each sample is replicated a number of times as mentioned above, so that the experimental error can be characterized. This estimate of experimental error becomes a basic unit of measurement for determining whether observed differences in the data are really *statistically* different. In other words, the analysis of variance answers the question "Is the observed difference due to chance or an actual difference in the population means?"

A statistical software package is generally used to perform the ANOVA [59, 73], therefore the computations that make up the analysis of variance are not discussed here.

6.3.3 Procedure

To be able to characterize the experimental error, the experiments need to be replicated. Thus, several sets of calibration and evaluation data need to be generated in the workcell. A very large calibration set should be chosen to ensure that the CBF model is well trained. The number of replicates n should be at least three, but if the expense of performing these runs is low, more runs can be performed to enhance statistical accuracy. Selecting the factor levels for each run from the orthogonal array of Table 6-2 and replicating each run n times, the

resulting error statistics are generated and recorded. The metric that constitutes the *observation* and quantifies the effect of the factors needs to be chosen depending on the process. Average evaluation error, maximum evaluation error or a combination of the two could be selected.

The observations, along with the factors and levels, then are read into the statistical software package and the ANOVA is performed. The result of this procedure is a table that indicates which factors are statistically significant and which factors are *important*, which is the case if more than 5% of variance is explained by the factor in question. First, attention should be given to the significance and importance of the interaction effects.

If none of the confounded blocks of interaction effects are significant and important, the confirmation experiment, which is performed next, consists of a full factorial including only the factors that were found to be significant and important in the screening experiment. The ANOVA table of this confirmation experiment and a table of observations can be used to select the combination of basis functions which results in the lowest evaluation error. This combination represents the Optimal Basis Function set for this part.

If some of the confounded blocks of interaction effects turn out to be significant and important in the screening experiment, the confirmation experiment includes the factors that were found to be significant and important, plus those factors that are part of significant and important interaction effects. Should the resulting number of runs required for the full factorial be prohibitive, another

fractional factorial can be performed to determine which of the interaction effects that are part of a confounded block are actually significant and important. A full factorial confirmation experiment would again follow.

The outlined procedure produces an Optimal Basis Function set for one particular part in one particular process and maximizes the performance of the Direct Calibration method for this part only. Unlike the development of Perfect Basis Functions for the simulated parts in Chapter 4, no model of the part is required for this emperical procedure.

CHAPTER 7: SUMMARY AND CONCLUSIONS

7.1 Summary

Flexible Automation enables manufacturers to get their products to market quickly and in high volume. Frequently, in the design of the flexible automation workcell, part feeding is the bottleneck when getting the part into a known position and orientation becomes the biggest challenge. Chapter 2 illustrates different approaches for either preserving, enforcing, or sensing the part location in flexible feeders. It also introduces the concept of grasp error and illustrates that the most flexible method of dealing with this part-in-hand error is to measure it. When the part-in-hand error is known, it can be accommodated by modifying the robot path to cancel out the error so that expensive reorienting of the part is not required. The Direct Calibration approach is based on this idea, but also introduces an elegant technique for calibrating the robot, the sensors, and the parts to be handled in a single procedure. The described research analyzes the significance of the basis functions used in the process mapping that is at the heart of the Direct Calibration technique. The questions posed in Chapter 1 can now be answered:

- Can the Direct Calibration performance be improved by a better choice of basis functions?

  The performance can be improved, but not without a loss of general

applicability of the method. Significant improvements are seen when the basis function set is optimized for a specific part.

- How can the Perfect Basis Functions for certain classes of parts be found in simulation?

  When the geometric relationship between part perturbation in robot space and sensor values in pixel space can be accurately modeled, the Perfect Basis Functions can be extracted from the equations.

- How do these Perfect Basis Functions differ from the basis functions in use in the Direct Calibration workcell?

  When the Perfect Basis Functions are broken down into their elementary terms, these terms can be compared to the Quadratic Basis Function terms.

- Which set of basis functions should be used for a general application of the Direct Calibration technique?

  For a general application of the Direct Calibration technique the Quadratic Basis Function set provides wide applicability and very good performance.

- If a general applicability of the Direct Calibration technique is not important, how can the performance for a specific part be optimized without requiring a model of the part?

  An empirical technique has been presented which employs methods from experimental design and statistics to select the Optimal Basis Functions for a certain part.

7.2 Recommended Practices

For the Direct Calibration user who is more concerned about performance than about gaining insight into the process, some straightforward recommendations can be made.

For a general application of Direct Calibration, continue to use the Quadratic Basis Function set with a 125-point calibration data set. The results from Chapters 4 and 5 show that a set of basis functions is not likely to be found that results in appreciably better performance for all parts.

If, however, the system does not have to accommodate more than one part, and the added cost of maximizing performance is justified, the procedure described in Section 6.3 should be followed. In this case, a very large set of calibration data would be acquired, and methods from design of experiments and statistics would be used to select from the Cubic Basis Function set the combination of terms that results in best performance for the part in question.

7.3 Key Contributions

The key contributions of this research are summarized as follows:

- The limits to performance improvement of the Direct Calibration technique due to changes in basis functions for certain classes of parts has been established in simulation.

- A set of basis functions for the general application of Direct Calibration has been recommended.
- Perfect Basis Functions have been developed for the most common classes of parts. These Perfect Basis Functions were broken down into their elemental terms and were compared to the basis function set that was found to be best for a general application of Direct Calibration.
- The performance potential of the Cubic Basis Function set has been evaluated in simulation.
- A technique for finding the optimal part-specific basis functions, without the need for a CAD model of the part, has been presented but not implemented.

BIBLIOGRAPHY

[1] "Feeds Components in Robotic Workcells," *Production Engineering*, vol. 34(6), pp. 20.

[2] *Material and Parts Handling in Manufacturing*, in *Tool and Manufacturing Handbook*, P. Mitchell, Editor. Society of Manufacturing Engineers: p. 18-13.

[3] "New Alternative to Hard-Tooled Parts Feeders in Robotic Assembly Workcells," *Design News*, vol. 43(15), pp. 107.

[4] S. Akella, W.H. Huang, K.M. Lynch, and M.T. Mason, "Sensorless Parts Orienting with a One-Joint Manipulator," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 2383-2390, 1997.

[5] S. Akella and M.T. Mason, "Parts Orienting with Shape Uncertainty," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 565-572, 1998.

[6] J. Alvite and L. Hendrickson, "Innovative Packaging for Electrical Components," in *Proc. RIA Flexible Parts Feeding for Automated Handling and Assembly Workshop*, Robotic Industries Association, pp. 139-146, 1993.

[7] A. Arnstroem and P. Groendahl, "A High Speed Small Batch Automatic Assembly System with Flexible Feeding and Automatic Setup," in *Proc. 8th Int. Conf. on Assembly Automation*, IFS (Publications) Ltd., pp. 237-248, 1993.

[8] J.P. Baartman, A.E. Brennemann, S.J. Buckley, and M.C. Moed, "Using Coarse/Fine Manipulation with Vision to Place Fine Pitch SMD Components," in *Proc. IEEE/CHMT European International Electronic Manufacturing Technology Symposium*, IEEE, pp. 262-266, 1989.

[9] D.R. Berkowitz and J. Canny, "Designing Parts Feeders using Dynamic Simulation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 1127-1132, 1996.

[10] R.-P. Berretty, *et al.*, "Trap Design for Vibratory Bowl Feeders," in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2558-2563, 1999.

[11] R.-P. Berretty, K. Goldberg, M.H. Overmars, and A.F.v.d. Strappen, "On Fence Design and the Complexity of Push Plans for Orienting Parts," *13th Symposium on Computational Geometry*, pp. 21-29, 1997.

[12] J. Birk, R. Kelley, and H. Martins, "An Orienting Robot for Feeding Workpieces Stored in Bins," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-11(2), pp. 151-160, 1981.

[13] D. Boehlke, "Smart Design for Flexible Feeding," *Machine Design*, vol. 66(23), pp. 132-134, 1994.

[14] R. Bolles, P. Horaud, and M.J. Hannah, "3DPO: A Three-Dimensional Part Orientation System," in *Proc. 1st Int. Symposium on Robotics Research*, MIT Press, pp. 413-424, 1984.

[15] G. Boothroyd, C. Poli, and L.E. Murch, *Automatic Assembly*, New York, Marcel Dekker, 1982.

[16] M.S. Branicky, G.C. Causey, and R.D. Quinn, "Toward a Science in Flexible Feeding," in *Proc. Int. Conf. on Advanced Intellectual Mechatronics*, 1999.

[17] R. Brost, "CAD-Based Approach to Feeders," in *Proc. RIA Flexible Parts Feeding for Automated Handling and Assembly Workshop*, Robotic Industries Association, pp. 117-130, 1993.

[18] S. Buckley, "Force-Field Sensing for QC," *Tooling and Production*, vol. 54(11), pp. 57-59, 1989.

[19] A. By, "Modular Gauging and Feeding for Lock Assembly," in *Proc. RIA Flexible Parts Feeding for Automated Handling and Assembly Workshop*, Robotic Industries Association, pp. 193-206, 1993.

[20] A. By, *et al.*, "An Adaptable Feeding Approach Based on the Modular Automated Reconfigurable Assembly System Concept," in *Proc. RIA Flexible Parts Feeding for Automated Handling and Assembly Workshop*, Robotic Industries Association, 1994.

[21] M. Caine, "The Design of Shape Interactions using Motion Constraints," in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 366-371, 1994.

[22] J. Calhoun and R. Baird, "Silhouette Imaging for Feeding and Orienting Parts," *Assembly Automation*, vol. 15(3), pp. 24-25, 1995.

[23] J.B. Calhoun and R.K. Baird, "Successful Application of Silhouette Imaging," *Vision*, vol. 11(4), pp. 4-7, 1995.

[24] J.F. Canny and K.Y. Goldberg, "A RISC Approach to Sensing and Manipulation," *Journal of Robotic Systems*, vol. 12(6), pp. 351-363, 1995.

[25] B.R. Carlisle, "Flexible Parts Feeder," *US Patent Number 5,687,831*, 1997.

[26] G.C. Causey and R. Quinn, "Design of a Flexible Parts Feeding System," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 1235-1240, 1997.

[27] G.C. Causey, R.D. Quinn, and M.S. Branicky, "Testing and Analysis of a Flexible Feeding System," in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2564-2571, 1999.

[28] A.D. Christiansen, A.D. Edwards, and C.A.C. Coello, "Automated Design of Part Feeders using a Genetic Algorithm," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 846-851, 1996.

[29] A. Cokayne, "The Way of the World," *Assembly Automation*, vol. 11(4), pp. 29-32, 1991.

[30] M.G. Coutinho and P.M. Will, "A General Theory for Positioning and Orienting 2D Polygonal or Curved Parrts using Intelligent Motion Surfaces," in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 856-862, 1998.

[31] J.J. Craig, *Introduction to Robotics: Mechanics and Control*, Addison-Wesley, 1989.

[32] W.F. Davis, "The Systems Approach to Parts Handling," in *Proc. RIA Flexible Parts Feeding for Automated Handling and Assembly Workshop*, Robotic Industries Association, pp. 169-180, 1993.

[33] R. Eade, "Automatic Assembly: The Key is Orientation," *Manufacturing Engineering*, vol. 102(3), pp. 64-69, 1987.

[34] M.A. Erdmann and M.T. Mason, "An Exploration of Sensorless Manipulation," *IEEE Journal of Robotics and Automation*, vol. 4(4), pp. 369-378, 1988.

[35] M.E. Friedberg, M.J. Jakiela, and K.T. Ulrich, "A Computer-Based Technical and Economic Model Choosing Automated Assembly Parts Presentation Equipment," in *Proc. ASME Flexible Assembly Systems*, ASME, vol. DE-28, pp. 85-89, 1990.

[36] T. Fujita, K. Sato, and S. Inokuchi, "Bin-Picking of Curved Object using Range Finder," in *Proc. Japan/USA Symposium on Flexible Automation*, ASME, pp. 565-572, 1986.

[37] M.A. Glabicky-Fegan, "Elimination of Downtime in an Automated Assembly Operation by Nondestructive Qualification of Feed Parts," *Materials Evaluation*, vol. 45(11), pp. 1280-1284, 1987.

[38] K. Goldberg, "Removing "Black Art" from Feeder Design," in *Proc. RIA Flexible Parts Feeding for Automated Handling and Assembly Workshop*, Robotic Industries Association, pp. 109-115, 1993.

[39] K. Goldberg, J. Craig, B. Carlisle, and R. Zanutta, "Estimating Throughput for a Flexible Part Feeder," in *Proc. Th 4th Annual Symposium on Experimental Robotics*, Stanford, CA, Springer-Verlag, pp. 486-497, 1995.

[40] K.Y. Goldberg, "Orienting Polygonal Parts without Sensors," *Algorithmica*, vol. 10(2-4), pp. 201-225, 1993.

[41] J.L. Goodrich and W.L. Devlin, "Programmable Parts Feeder," *US Patent Number 4,608,646*, 1986.

[42] S. Gordon, "Elimination of Customized Fixtures Through High-Performance 3D Vision," in *Proc. ASME 1st Conf. on Flexible Assembly Systems*, ASME, pp. 9-14, 1989.

[43] S.J. Gordon, "On-Line-Reconfigurable Parts Feeder," in *Proc. 2nd ASME Design Technical Conference on Flexible Assembly Systems*, ASME, pp. 127-131, 1990.

[44] S.J. Gordon, "Programmable Reconfigurable Parts Feeder," *US Patent Number 5,314,055*, 1994.

[45] D. Gudmundsson and K. Goldberg, "Tuning Robotic Part Feeder Parameters to Maximize Throughput," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 2440-2445, 1997.

[46] D. Herndon, R.C. MacKenzie, R.S. Newmann, and E. Trager, "Robotically Controlled Component Feed Mechanism Visually Monitoring Part Orientation," *US Patent Number 4,909,376*, 1990.

[47] J.W. Hill, "Programmable Bowl Feeder Design Based on Computer Vision," *Assembly Automation*, pp. 21-25, November 1980.

[48] H. Hitakawa, "Advanced Parts Orientation has Wide Application," *Assembly Automation*, vol. 8(3), pp. 147-150, 1988.

[49] Y. Ho and S. El-Gizawy, "A Programmable, Multi-Part Presentation System for Robot Assembly," in *Proc. IFAC Workshop on Intelligent Manufacturing Systems*, International Federation of Automatic Control, pp. 519-523, 1994.

[50] Y. Ho, M. Shelat, and A.S. El-Gizawy, "Development of a Sensory-Based Part Feeding and Sorting Mechanism," in *Proc. IASTED Control and Robotics*, International Association of Science and Technology for Development, pp. 133-136, 1992.

[51] J. Hollingum, "Sweeping it over the Carpet," *Assembly Automation*, vol. 15(3), pp. 29-30, 1995.

[52] W. Iversen, "Vision-Guided Robots for Assembly," *Assembly*, pp. 30-31, June 1993.

[53] P. Lamoreaux, "Vision Guided Parts Pick-Up and Placement," in *Proc. RIA Flexible Parts Feeding for Automated Handling and Assembly Workshop*, Robotic Industries Association, pp. 147-156, 1993.

[54] K.-M. Lee, "Flexible Part-Feeding System for Machine Loading and Assembly: Part II. A cost-effective Solution," *Journal of Production Economics*, vol. 25pp. 155-168, 1991.

[55] K.-M. Lee and Y. Qian, "Intelligent Vision-Based Part-Feeding on Dynamic Pursuit of Moving Objects," in *Proc. SPIE Modeling, Simulation, and Control Technologies for Manufacturing*, International Society for Optical Engineering, vol. 2596, pp. 172-183, 1995.

[56] L.E.N. Lim, *et al.*, "A Computer-Aided Framework for the Selection and Sequencing of Orienting Devices for the Vibratory Bowl Feeder," *International Journal of Production Research*, vol. 32(11), pp. 2513-2524, 1994.

[57] E.K. Lo and M.D. Dick, "The Solid Modeling of Part Orienting Tracks for Automated Assembly," *Int. Journal on Production Research*, vol. 28(8), pp. 1513-1525, 1990.

[58] R.M. Lougheed and R.E. Sampson, "3-D Imaging Systems and High-Speed Processing for Robot Control," *Machine Vision and Applications*, vol. 1pp. 41-57, 1988.

[59] J.T. Luftig and V.S. Jordan, *Design of Experiments in Quality Engineering*, McGraw-Hill, 1998.

[60] J.E. Luntz, W. Messner, and H. Choset, "Discrete Actuator Array Vectorfield Design for Distributed Manipulation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2235-2241, 1999.

[61] K.M. Lynch, "Toppling Mnipulation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2551-2557, 1999.

[62] R. Mackenzie, "Electronic Components: Robotic Test and Assembly," *Assembly Engineering*, pp. 38-41, August 1987.

[63] W.A. Mahoney, "Shallow Pocket Mechanically Interlocking Tapes for Odd Form Surface Mount Automation," in *Proc. RIA Flexible Parts Feeding for Automated Handling and Assembly Workshop*, Robotic Industries Association, pp. 148-153, 1993.

[64] K. Mangle, G. Richard, and W. Desrude, "Robots + Vision = Flexible Assembly," *Assembly*, March 1997.

[65] M. Mani and W.R.D. Wilson, "A Programmable Orienting System for Flat Parts," *Manufacturing Engineering Transactions*, pp. 427-432, Sept. 1985.

[66] G. Maul and J. Hildebrand, "Research for Low-Cost Flexible Feeding of Headed Parts using Bi-Directional Belts," *Int. Journal of Production Research*, vol. 23(6), pp. 1121-1130, 1985.

[67] G.P. Maul and N.I. Jaksic, "Sensor-Based Solution to Contiguous and Overlapping Parts in Vibratory Bowl Feeders," *Journal of Manufacturing Systems*, vol. 13(3), pp. 190-195, 1994.

[68] G.P. Maul and M.B. Thomas, "A System Model and Simulation of the Vibratory Bowl Feeder," *Journal of Manufacturing Systems*, vol. 16(5), pp. 309-314, 1997.

[69] F.L. Merat, N.A. Barendt, and R.D. Quinn, "Advances in Agile Manufacturing," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 1216-1222, 1997.

[70] B. Mirtich, *et al.*, "Estimating Pose Statistics for Robotic Part Feeders," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 1140-1146, 1996.

[71] A. Miyakawa, S. Hata, M. Uno, and M. Matsunaga, "A Flexible Assembly Station with Visual Sensors," in *Proc. IECON 1984*, IEEE Industrial Electronics Society, pp. 1036-1040, 1984.

[72] O. Mohammed, "State-of-the-Art in Part Feeding, Part Presentation, and Part Mating in Flexible Assembly Automation," *ISA Advances in Instrumentation*, vol. 43/4pp. 1415-1423, 1988.

[73] D.C. Montgomery, *Design and Analysis of Experiments*, John Wiley & Sons, Inc., 1997.

[74] W.R. Murray and D.A. Billingsley, "A Neural Network for Simplified Pose Recognition in Vision-Based Parts Feeders," *Robotics and Computer Integrated Manufacturing*, in press.

[75] W.R. Murray and C.M. Pohlhammer, "A Technical Approach for Flexible Tray Loading: Proof-of-Concept Experiments on Using Direct Calibration for Determining Part-in-Hand Location," *Robotics and Computer Integrated Manufacturing*, in press.

[76] I.A.D. Nesnas, "Computer Vision Strategies for Flexible Parts Feeding," in *Proc. Int. Robots and Vision Conference*, 1997.

[77] T. Nonaka and K. Otsuka, "Photoelectric Control System for Parts Orientation," *US Patent Number 4,333,558*, 1982.

[78] K. Ohba, Y. Sato, and K. Ikeuchi, "Visual Learning and Object Verification with Illumination Invariance," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robot and Systems*, IEEE and Robotics Society of Japan, vol. 2, pp. 1044-1050, 1997.

[79] G. Orelind, "Flexible Part Feeding For Robotic Workcells," *Robotics World*, vol. 11(2), pp. 12-14, 1993.

[80] C. Ou-Yang and G. Maul, "A Computer Analysis of Orientation Devices for Vibratory Bowl Feeders," *Int. Journal of Production Research*, vol. 31(3), pp. 555-578, 1993.

[81] J.S. Park, *Performance Evaluation of the Direct Calibration Technique Applied to Printer Parts*, University of Washington, 1997.

[82] R.E. Parkin, *Applied Robotic Analysis*, Prentice Hall, 1991.

[83] E. Paulos and J. Canny, "Accurate Insertion Strategies Using Simple Optical Sensors," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 1656-1662, 1994.

[84] M.A. Peshkin and A.C. Sanderson, "Planning Robotic Manipulation Strategies for Workpieces that Slide," *IEEE Journal of Robotics and Automation*, vol. 4(5), pp. 524-531, 1988.

[85] M.S. Phadke, *Quality Engineering Using Robust Design*, Prentice-Hall, 1989.

[86] D. Pherson, G. Boothroyd, and P. Dewhurst, "Programmable Feeder for Non-Rotational Parts," *Manufacturing Systems*, vol. 13(2), pp. 112-122, 1983.

[87] C.M. Pohlhammer, *Sensing for Automated Assembly: Direct Calibration Techniques for Determining Part-in-Hand Location*, University of Washington, 1997.

[88] R.D. Quinn, *et al.*, "Design of an Agile Manufacturing Workcell for Light Mechanical Applications," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 858-863, 1996.

[89] A. Rao, D. Kriegman, and K. Goldberg, "Complete Algorithms for Reorienting Polyhedral Parts using a Pivoting Gripper," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 2242-2248, 1995.

[90] A.S. Rao and K.Y. Goldberg, "Friction and Part Curvature in Parallel-Jaw Grasping," *Journal of Robotic Systems*, vol. 12(6), pp. 365-381, 1995.

[91] A.H. Redford, E.K. Lo, and P.J. Killeen, "Parts Feeder for a Multi-Arm Assembly Robot," *Manufacturing Systems*, vol. 13(2), pp. 145-149, 1983.

[92] G.P. Reimann, "Concurrent Engineering of a Polaroid Camera and Flexible Automation," in *Proc. Japan/USA Symposium on Flexible Automation*, ASME, vol. 2, pp. 895-900, 1996.

[93] D. Reznik and J. Canny, "A Flat Rigid Plate is a Universal Planar Manipulator," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 1471-1477, 1998.

[94] M. Rockland and R. Stetter, "Flexibles Ordnen und Zuführen," *Zeitschrift für Wirtschaftliche Fertigung und Automatisierung*, vol. 89(1-2), pp. 55-57, 1994.

[95] B. Ross, "Flexible Feeding Devices for Robotic Workcells," in *Proc. RIA Flexible Parts Feeding for Automated Handling and Assembly Workshop*, Robotic Industries Association, pp. 157-168, 1993.

[96] B. Ross, "Flexible Part Feeders for Vision Guided Robots," in *Proc. RIA Flexible Parts Feeding for Automated Handling and Assembly Workshop*, Robotic Industries Association, 1994.

[97] E.M. Ross, "Flexible Parts Feeders For Robotic Assembly," *Assembly*, pp. 24-28, October 1994.

[98] S. Rusaw, K. Gupta, and S. Payandeh, "Determining Polygon Orientation using Model Based Force Interpretation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 544-549, 1998.

[99] S. Rusaw, K. Gupta, and S. Payandeh, "Part Orienting with a Force/Torque Sensor," in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2545-2550, 1999.

[100] A. Salvarinov and S. Payandeh, "Flexible Part Feeder: Manipulating Parts on Conveyor Belt by Active Fence," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 863-868, 1998.

[101] M. Shirai and A. Saito, "Parts Supply in Sony's General-Purpose Assembly System "SMART"," *Japan Journal of Advanced Automation Technology*, vol. 1(108), pp. 28-31, 1989.

[102] S. Su and I. Uzmay, "Dynamic Model for High-Speed Pushing as a Manipulator Operation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 850-855, 1998.

[103] Y. Sugino, K. Yamaki, Y. Shiote, and K. Miyashita, "Flexible Assembly Cell for Parts Supplied in Disarray on a Kit Tray," in *Proc. Japan/USA Symposium on Flexible Automation*, ASME, pp. 737-742, 1988.

[104] T. Suzuki, T. Sakata, T. Kawana, and M. Kohno, "An Approach to a Flexible Part-Feeding System," in *Proc. 1st Conf. on Assembly Automation*, IFS Publications, pp. 275-286, 1980.

[105] M.L. Tay and P.S.K. Chua, "Feed Orientation of Rectangular and Cylindrical Parts on Vibratory Bowl Feeder Tracks," in *Proc. Int. Conf. on Computer Integrated Manufacturing (ICCIM)*, GINTIC Institute of CIM, pp. 353-356, 1991.

[106] A.S. Wallack and J.F. Canny, "Generalized Polyhedral Object Recognition and Localization using Crossbeam Sensing," *Int. Journal of Robotics Research*, vol. 16(4), pp. 473-496, 1997.

[107] J. Wiegley, K. Goldberg, M. Peshkin, and M. Borowski, "A Complete Algorithm for Designing Passive Fences to Orient Parts," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 1133-1139, 1996.

[108] W. Wolfson and S.J. Gordon, "Designing a Parts Feeding System for Maximum Flexibility," *Assembly Automation*, vol. 17(2), pp. 116-121, 1997.

[109] M. Yeong, L. Ruff, and W.R.D. Vries, "A Survey of Part Presentation, Feeding, and Fixturing in Automated Assembly Systems," in *Proc. ASME Flexible Assembly Systems*, ASME, vol. DE-33, pp. 83-90, 1991.

[110] K. Yoshida, "High Speed Product Assembly," in *Proc. RIA Flexible Parts Feeding for Automated Handling and Assembly Workshop*, Robotic Industries Association, pp. 91-108, 1993.

[111] R. Zhang and K. Gupta, "Automatic Orienting of Polyhedra through Step Devices," in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, pp. 550-556, 1998.

[112] N.B. Zumel and M.A. Erdmann, "Nonprehensile Manipulation for Orienting Parts in the Plane," in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2433-2439, 1997.

APPENDIX A: L_{32} ORTHOGONAL ARRAY

Run #	Column																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
4	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
5	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
6	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
7	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
8	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
9	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
10	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
11	1	0	0	1	1	0	0	0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0	0	0	1	1	0	0	1	1
12	1	0	0	1	1	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0
13	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1
14	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
15	1	0	0	0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0	0	0	1	1	0	0	1	1	1	1	0	0
16	1	0	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	1	1
17	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
18	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
19	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1
20	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0
21	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1
22	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0
23	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0
24	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1
25	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
26	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
27	0	0	1	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0	0	1	1	0
28	0	0	1	1	0	0	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	1
29	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0
30	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1
31	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1
32	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0

VITA

Ulix Goettsch

PhD Mechanical Engineering 2001	University of Washington
MS Mechanical Engineering 1994	University of Arizona
BS Mechanical Engineering 1992	University of South Alabama