Automatic Part Localization for Fixtureless Assembly of Automotive Sheet Metal Parts

by

Edward J. Park

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Mechanical and Industrial Engineering
University of Toronto

Abstract

This thesis presents a solution to the problem of part localization in fixtureless assembly. Fixtureless assembly is a new technology in which complex and expensive dedicated fixtures are replaced by robots. However, the elimination of the fixtures implies that the robots must now be able to achieve the same locational accuracy of parts as precision fixtures provide. The implementation of an Automatic Part Localization System (APLS) allows the robots to determine the accurate position and orientation of the parts that are placed in arbitrary locations. In this thesis, the problem of part localization is studied in the context of an application to automotive sheet metal part assembly, specifically car body fenders. The development of the APLS is summarized. The APLS uses laser beam sensors to estimate, in real time, the spatial position and orientation of a sheet metal part held by a robot. It comprises two groups of sensors: proximity sensors that measure the location of the part's surface, and edge sensors that measure the location of the part's edges. From these surface and edge feature data, the APLS is able to localize the part for all possible coordinates. In other words, it is able to solve 6 DOF localization problems. The implementation of an existing localization method is described. The localization algorithm presented uses minimization of least-square to generate the best-fit mapping between the part's geometry and the corresponding sensor information. Experiments are conducted to investigate the feasibility of the proposed part localization concept. The experimental results show that the presented method meets the target performance of this investigation – to localize the part within the industrial tolerance of $\pm 0.5mm$ in selected positional coordinates. The experiments are carried out using a 6 DOF Fanuc commercial robot.

Acknowledgements

I foremost wish to sincerely thank Prof. Mills for his invaluable guidance, support, and encouragement throughout the course of this thesis.

I am grateful for all past and present members of the Nonlinear Control Systems Laboratory. Their presence and help made this work very enjoyable. In particular, I wish to thank Weihua Xu for the numerous discussions we had.

I give my appreciation to General Motors of Canada, and Mr. Carl Wintermeyer in particular, for supplying the Fanuc S-110 robot.

I extend my appreciation to the Institute for Intelligent and Robotic Systems for providing the financial support.

I express my gratitude to Prof. Benhabib for giving me the access to resources in the Computer Integrated Manufacturing Laboratory. I also extend my thanks to Martin Bonert and Gene Zak for their time and comments in helping me to set up the robot.

I wish to thank all my friends at Yae Dalm Church for their continuing friendship and prayers.

I thank my beloved parents and my sister for their never-ending support and encouragement throughout this work.

I give special thanks to Eunice for her love and companionship. We have been through a lot of wonderful things together during these years.

Finally, I thank God who made everything possible. He is my strength and delight.

Table of Contents

List of Figures

List of Tables

Nomenclature

• Roman Letters

$a_{i,j}$	Elements in inverse calibration matrix
$b_{i,j}$	Elements in forward calibration matrix
C_{FOR}	Forward calibration matrix
C_{INV}	Inverse calibration matrix
F	Function
∇F	Gradient of F
F_{FOR}	Forward calibration function
F_{INV}	Inverse calibration function
k	Number of sensors in effect
L	True part location vector
$\hat{L}$	Estimated part location vector
$\overline{L}$	Augmented part location vector
L_{CAL}	Part location calibration set
L_i	Part location vector
$\hat{L}_o$	Initial part location
$\Delta\hat{L}_o$	Adjustment for part location
m	Total number of degrees of freedom
N	Total number of perturbations
n	Maximum number of iterations

q_i Part location deviation in coordinate i

$\hat{q}_i$ Approximated part location deviation in coordinate i

S True sensor reading vector

$\hat{S}$ Estimated sensor reading vector

$\overline{S}$ Augmented sensor reading vector

S_{CAL} Sensor reading calibration set

S_i Sensor reading vector

s_i Sensor reading deviation for sensor i

$\hat{s}_i$ Approximated sensor reading deviation for sensor i

$\nabla\hat{s}_i$ Gradient of $\hat{s}_i$

• Greek Letters

ε Prescribed limit value

η Step length parameter

λ_i Scalar sensitivity weight for sensor i

• Abbreviations

APLS Automatic Part Localization System

DOF Degrees of Freedom

LNSC Laboratory of Nonlinear Systems Control

SPF Sensor Positioning Frame

SVD Singular Value Decomposition

Chapter 1

Introduction

1.1 Fixtureless Assembly of Sheet Metal Parts

Current automotive robotic assembly processes require fixtures to accurately position and orient parts. A typical automotive body, for example, is comprised of numerous number of pieces of sheet metal parts. During assembly, the different sheet metal parts are held by dedicated fixtures while robotic manipulators spot-weld them together. For each set of parts, a different set of fixtures is needed. In addition, these fixtures for the automotive sheet metal parts are very complex and difficult to design.

Fixtures in automotive assembly require considerable capital to design and manufacture. A manufacturer may buy a $100,000 robot manipulator for various tasks of assembly, but fixtures for a single task may cost an additional $200,000 to $400,000 [18]. Annual introduction of new car models or changes in styling demands thousands of engineering hours for the design of fixtures. The corresponding fixtures must be finalized before assembly robots and other automation machinery can be installed, programmed, and debugged. Lead time for such fixtures may easily exceed three years, thus resulting in a long lead time for the subsequent changes in car models [29]. The increased pace in

technology over the past decade has reduced lead time and cost of general automotive production. However, the traditional approach of using dedicated and complex fixtures remains a fundamental limitation in our pursuit of improved time and cost to market. Fixtureless assembly is a new technological basis for providing solutions to this problem.

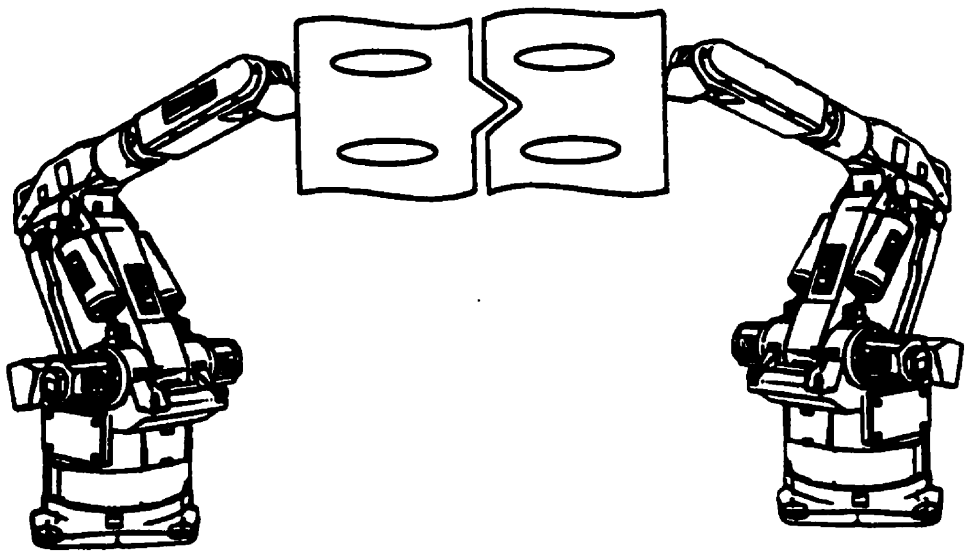

Figure 1-1: Schematic of Fixtureless Assembly of Sheet Metal Parts

Fixtureless assembly is a process that eliminates the use of dedicated fixtures by using multiple robots (Figure 1-1). In the fixtureless assembly of automotive sheet metal parts, for example, two robots grasp one sheet metal part each for mating while a third robot welds them together. In this scheme, robots provide several advantages over fixtures [19, 29]: 1) robots can handle different types and shapes of parts by simply programming them to do so, hence eliminating the need for inflexible "dedicated" fixtures; 2) they can easily be reprogrammed to deal with modifications or perform new tasks, therefore eliminating the change-over time, which could take several weeks, required in installing new fixtures; 3) programming can be done off-line in conjunction with the product

development and setup of other automation machinery and robots, thereby eliminating the long lead time required for the implementation of fixtures.

Acoording to [19] and [29], fixtureless assembly tasks involve the following three stages. The first stage is called the pre-contact stage. In this stage, the two robots would each pick up a part, then determine the misaligment present between two parts and move the parts into correct positions for mating. The second stage is called the mating stage. In this stage, the robots would approach each other allowing the parts to make contact within a specified contact force. The third stage is called the maintaining stage. Here, a third robot performs welding of the parts. While the welding robot bond the parts together, the two robots must maintain the parts' position and force configurations during the presence of disturbances introduced by the welding process.

This thesis focuses on the first, pre-contact stage - to determine the misalignment present in parts picked up by robots. To solve this problem, a methodology called "part localization" is needed [15]. The use of part localization in fixtureless assembly allows for a robot to be able to determine the accurate location of production parts, hence eliminating the need for dedicated fixtures.

1.2 Part Localization

Over the last decade, driven by a need to automate various robotic manufacturing processes, the problem of part localization has received a great deal of attention. The term "part localization" refers to the determination of a part's location, i.e., both the position and orientation, with respect to a reference frame. Part localization is necessary in many manufacturing situations. For example, in the task of grasping of an object by a robot the successful grasping depends on knowing the precise location of the object with respect to the gripper of the robot before attempting the grasp. Consider another part localization example - in robotic welding, the location of the weld groove must be accurately known

with respect to the weld torch held by the robot. Figure 1-2 gives a pictorial description of such part localization problem. The planned location of the sheet metal part is shown in dashed lines and the actual location in shown in solid lines. Without localization, the robot will unsuccessfully try to grasp the part in the planned location. In Figure 1-2, part localization determines the uncertainty or misalignment between the locations of the actual and planned parts in the robot's world frame; hence, enabling the robot to grasp the actual part successfully.

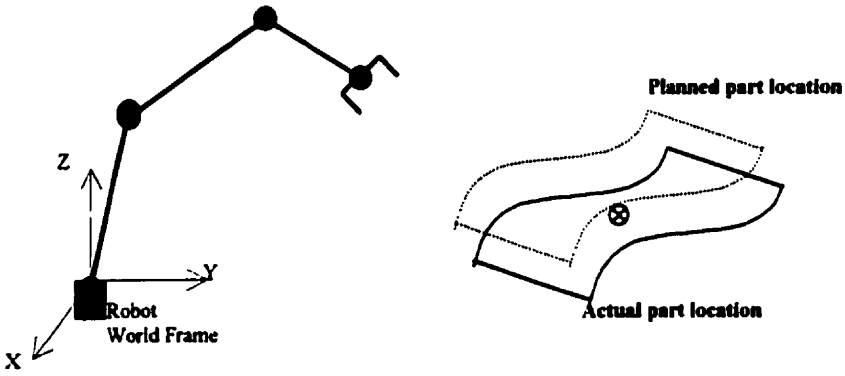

Figure 1-2: Schematic of Part Localization

In this thesis, the problem of part localization is studied in the context of an application to automobile sheet metal parts, specifically car body fenders. Figure 2-3 shows a picture of a car body fender used for this thesis. In automotive body assembly, the position tolerance required to accurately position sheet metal parts is about $\pm 0.5mm$ [24]. Therefore the acceptable orientation tolerance is obtained by assuming that the corners of the sheet metal parts can be displaced up to $\pm 0.5mm$, the position tolerance [27]. The orientation tolerance is estimated as $\pm 0.07°$. Thus the elimination of dedicated fixtures means that robots must now position the parts within the target performance given above.

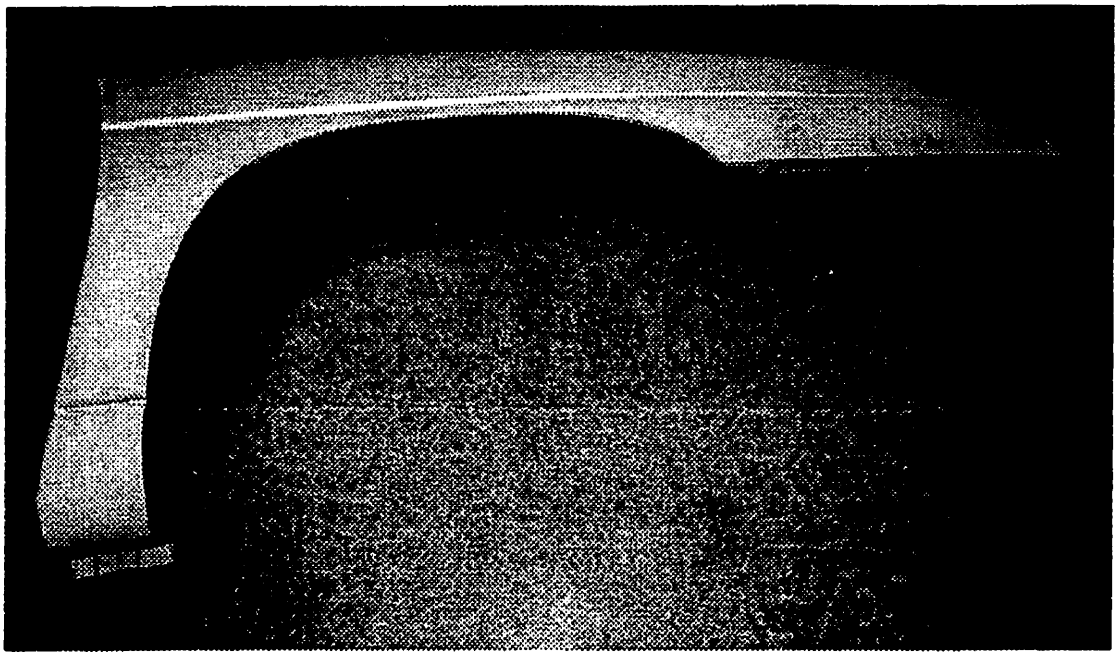

Figure 1-3: Picture of Car Body Fender

Before we can go further, there are two assumptions involved in applying part localization to fixtureless assembly: 1) the approximate location of parts are known and 2) the uncertainty in location of parts is small relative to the dimensions of the parts. To satisfy these two assumptions, low precision, general purpose positioners can be employed. Using these positioners, the parts are not placed in a precise location, but they are placed within a few millimeters (or even centimeters) of the location. The robots grasp the parts from the positioners and bring them within the tolerance of the precise location using part localization method.

From the literature survey, there are mainly two types of methods available in solving the problem of part localization: the CAD model-based method and the direct calibration method. In order to perform part localization, both methods integrate vision or sensor systems with the robot systems. Using cameras or sensors such as proximity, tactile, ultrasonic, part's location information can be obtained. The intent of this thesis is

to provide a solution using direct calibration method. Both methods, however, are described in Section 1.3.

1.2.1 Localization of Sheet Metal Parts

In part localization, the stability of the part during the taking of sensor measurements is critical for the accuracy of the localization results. However, maintaining the stability of a typical sheet metal part such as the car body fender (Figure 2-3) is difficult because of its irregular geometry and flexibility [4]. There are two issues to be considered in ensuring the stability of a sheet metal part: 1) the forces acting on the sheet metal part during localization and 2) the deformation in the shape of the part as a consequence. The shape of the part could be affected by the following forces or disturbances: gravity, vibration of robot manipulator, and grasping force of robot's gripper. In industry, this problem is solved by using multiple grippers, each a vaccum-suction cup, which can support the whole surface of the sheet metal part together at the same time.

1.3 Literature Review

Part localization has been primarily studied by researchers working in the areas of robotic vision and scene analysis [15]. Scene analysis is not generally intended for the use in the manufacturing and assembly and its main consideration is not in the accurate localization of parts, but only in object recognition [5, 8]. A primary objective of scene analysis is identification and recognition of the object in the scene. It is in the area of robotic vision that part localization has received a great deal of attention. Here, in many cases, the part to be localized is assumed to be known [15]. Therefore the main concern is not in the identification of the part, but on the accurate localization of it. This is justified in many manufacturing situations and the situation of this thesis is no exception. There are

reportedly two different approaches in part localization: the CAD model-based method and the direct calibration method.

To solve the problem of part localization, the CAD model-based method has been studied by a number of researchers [i.e., 15, 22, 25, 32]. The work by Gunnarsson and Prinz [15], best summarizes the subject. According to their work, the CAD model-based method is based on matching a CAD database description of the part's surface with data measured on an actual surface. The matching can be found through a rigid body transformation between the measurement data and the CAD database. The measurement data can be obtained using a system of tactile or non-tactile sensors. An optimal transformation is found by minimizing the sum of the squared distances of the of the measurement data from the corresponding point on the CAD database. Menq et al. [25], Li et al. [22], and Sahoo et al. [32] all provide solutions and algorithms to this minimization problem. For the purpose of localization, the CAD database must account for every point on the edge or the surface so that for any measured point of the actual part, there is a corresponding point on the database. Gunnarsson and Prinz [15] accomplishes this by using interpolation functions, such as splines, which are explicit equations of edges or surfaces of the part.

A new approach reported by Murray and Pohlhammer [27] is significantly different from the previous one. They call it a direct calibration. The concept of the direct calibration is a simpler approach that does not require the CAD-model database. It uses a calibration database, instead of the CAD database. The calibration database is a mapping that relates possible locations of the part to the corresponding sensor measurements taken from the part. By directly calibrating the actual part and the sensor measurements, the relationship between the part location space and sensor measurement space is mapped. The mapping between the two spaces can be obtained using the neural networks [26] or the minimization of least squares technique [27]. However, Murray and Pohlhammer [27]

reports that the least squares technique has a performance advantage over the neural network. Also, it is easier to implement because of its simplicity in the algorithm.

Now we are forced to ask the following question: Which one of the two part localization methods gives a better solution for the localization of sheet metal parts? The direct calibration has the following advantages over the CAD model-base method in localizing sheet metal parts:

- There is no reliance on CAD models of the part. When CAD models are used in estimating part location, the resulting estimation can be very sensitive to the accuracy which the CAD model represents the actual part. In addition, it is very difficult to build corresponding CAD models for sheet metal parts such as car body fenders because of their complexity in shape and flexibility. Especially, the flexibility could make an actual part grasped by a robot significantly different from its CAD model. Also, the direct calibration algorithm is robust with respect to large part-to-part variations in geometry [27].

- There is no need to know the locations of sensors. Another limitation in using the CAD model-based method is a lack of an efficient and simple procedure to find the location of the sensors, which comprise the measurement system. In the CAD model-based method, the measurement points are expressed in a Cartesian (world) coordinate frame [15]. What this means is, the locations of the sensors also have to be known in the Cartesian coordinate frame (see [23], [33], and [35] for the possible solutions to this problem).

For the reasons above, the direct calibration method is chosen in solving the problem of part localization in fixtureless assembly of sheet metal parts. The work by Murray and Pohlhammer [27], which is solely based on computer simulations, provides a

theoretical basis for the experimental results obtained in this thesis. The next chapter will develop the part localization algorithm in detail using the direct calibration method.

1.4 Thesis Objective

The main objective of this thesis is to develop and implement an automatic part localization system (APLS) on a robotic manipulator in a fixtureless assembly scheme. As a consequence, the robot will be equipped with intelligence that enables it to find the accurate position and orientation of sheet metal parts that are misplaced (or misaligned) in arbitrary locations. To provide focus, this is to be performed on a specific industrial application of car body fender assembly. The robotic assembly of fender is a delicate task which requires the APLS to correct a fender's misalignment within the positional tolerance of $\pm 0.5mm$ and the orientational tolerance of $\pm 0.07°$. The corresponding part localization algorithm must be developed to meet this challenge. Moreover, the feasibility of implementing the proposed algorithm and the APSL has to be demonstrated through testing and experiments.

1.5 Thesis Outline

This thesis presents the results of theoretical and experimental investigation. Chapter 2 describes the theoretical part of the thesis – the development of part localization algorithm. There are three components to the development of the algorithm: 3 DOF edge localization, 3 DOF surface localization, and 6 DOF localization. Each component is a modification of a generic localization theory presented in the very beginning of the chapter. In Chapter 3, the experimental APLS is presented. Here, the development of the system's hardware and software components are discussed. Since this is an essential part of the thesis, each component and the interaction between the components are described in great detail. Also additional issues involved in setting up the system for experiments are

addressed. Subsequently, the experimental results, which are obtained using the APSL, are discussed in Chapter 4. For both 3 DOF and 6 DOF misalignment problems, a series of experiments were conducted to compare the performance between the inverse localization and the forward localization methods. Finally, Chapter 5 presents conclusions to the thesis and makes recommendations for future work.

1.6 Thesis Contributions

The main contributions of this thesis are as follows:

1. The first contribution is the formulation of part localization algorithm in application to car body fender in fixtureless assembly scheme. The investigation by Murray and Pohlhammer [27] provided a theoretical basis for this expanded formulation.

2. The second contribution is the extension of the algorithm to the 6 DOF localization of a very complex shaped part – the car body fender. Very few investigations in the literature review successfully achieved the 6 DOF part localization and performed experiments.

3. The third contribution is the development of an automatic part localization system (APLS). The APSL was designed and developed by the author to demonstrate the feasibility of implementing the part localization algorithm on a commercial robot. It is capable of localizing up to six coordinate misaligements of the fender. The system uses a novel sensor system to carry out the localization: combined use of proximity sensors and edge sensors which are mounted on a sensor positioning frame. Most other investigations use cameras, ultrasonic sensors, or tactile sensors to localize a part.

4. The experimental results indicated the localization performance matches the industrial standards required for the assembly of car body fenders.

1.7 Summary

In this chapter, the introduction, objective and outline of the thesis, and its contributions are presented. Background information about the subject of part localization and the issues involved in localizing sheet metal parts (i.e., car body fender) are given. In addition, relevant works on the subject by other researchers are described in the literature review.

Chapter 2

Description of Localization Algorithms

2.1 Theory

Murray and Pohlhammer's theoretical work [27] provides the basis for the formulation of part localization presented in this chapter. The concept of part localization is to relate part location deviations, called q_i, to the corresponding sensor reading deviations, called s_i. The part location deviations are measured with respect to a planned (or home) location of the part, which is a car body fender in this thesis. In order to associate the two deviations, the following three steps are involved [27]: acquisition of calibration data, mapping, and localization.

2.1.1 STEP 1: Acquisition of Calibration Data

The calibration data is obtained with the part to be localized in the grasp of a robot. The robot moves the part by known amounts under the view of the sensors - this gives perturbations in the part location about the desired home part location - and then measurements are taken from readings of the sensors.

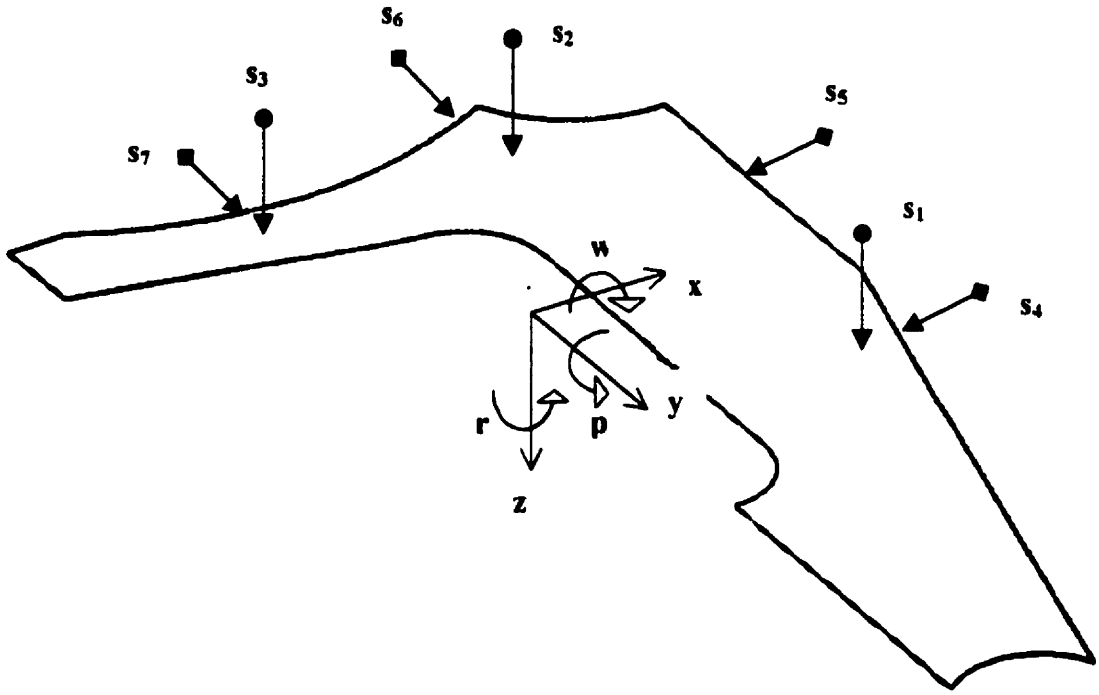

Figure 2-1: Representation of Coordinate Axes (x, y, z, w, p, r) and Sensor Vectors (s_1, s_2, s_3, s_4, s_5, s_6, s_7) in Part Space

The localization of a sheet metal part, specifically a car fender in this investigation, can be considered a 6 degrees of freedom (DOF) problem: three in position (x, y, z) coordinates and three in orientation (yaw or w, pitch or p, roll or r) coordinates. Figure 2-1 shows the representation of the six positions and orientation coordinate axes in the part space. It also shows the location of sensor reading vectors (s_1, s_2, s_3, s_4, s_5, s_6, s_7) from seven laser beam sensors in the part space: s_1, s_2, and s_3 correspond to the readings from proximity sensors and s_4, s_5, s_6, and s_7 correspond to edge sensors. The description of these sensors is given in Chapter 3. A calibration set consists of N perturbations formed by taking all combinations of a small number of uniform subdivisions of the pertinent ranges for the position and orientation; the N perturbations should be chosen to evenly span the part location space [27].

Now, let q_i be a part location deviation in a coordinate axis, m is defined as the total number of degrees of freedom or coordinate axes considered, k the number of sensors in effect, N the total number of perturbations in a calibration set. Then, for $1 \le i \le N$, a set of calibration vectors is [27]

$$L_{CAL} = \begin{bmatrix} L_1 \\ \vdots \\ L_N \end{bmatrix}_{CAL} \tag{2.1}$$

where:

$$L_i = \left[(q_1)_i, (q_2)_i, \cdots, (q_m)_i\right] \text{ part location vector,} \tag{2.2}$$

and

$$S_{CAL} = \begin{bmatrix} S_1 \\ \vdots \\ S_N \end{bmatrix}_{CAL} \tag{2.3}$$

where:

$$S_i = \left[(s_1)_i, (s_2)_i, \cdots, (s_k)_i\right] \text{ corresponding sensor reading vector.} \tag{2.4}$$

2.1.2 STEP 2: Mapping

In order to determine the relationship between the part location and sensor data, the calculation of mapping between the two is necessary. A linear least squares technique [21] is used to find the mapping.

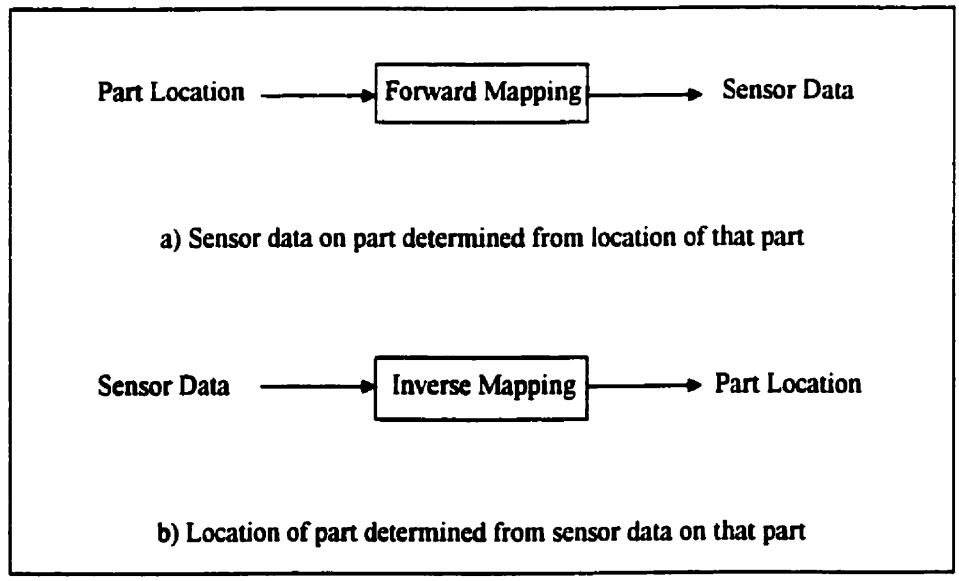

Figure 2-2: Relationship between Part Location and Sensor Data
a) Forward Mapping and b) Inverse Mapping

Analogous to the forward and inverse kinematics of robotic link manipulators, the process mapping of Equations (2.1) and (2.3) can also be formulated in both directions: forward mapping and inverse mapping [27]. These two mappings are shown in Figure 2-2.

Generally in part localization, the location of a part is to be determined from the sensor data for that part. This corresponds to the inverse mapping in Figure 2-2. However, [27] shows that the forward mapping scheme also works. If the forward mapping from the part locations to sensor data is learned and iteratively inverted during the localization of the part, the resulting performance is similar to that of the inverse mapping for the same

part. Also it notes that the forward mapping is considerably more robust to variations in part geometry and noise in sensor readings. However, to use the forward mapping, the formulation of the inverse mapping is also necessary because its equations are used to calculate the initial guess for the part location at the start of the forward mapping's iterative procedure.

a) Inverse Mapping

The inverse mapping process determines a part's location from sensor readings. The true location vector (L) is approximated by using a second-order polynomial (curvilinear) regression model [27]. Hence the response function for the regression model is [27]

$$L \cong \hat{L} \equiv \bar{S} \cdot C_{INV} \tag{2.5}$$

where:

$\hat{L}$ is an estimate of the location vector,

C_{INV} is called inverse mapping calibration matrix, and

$$\bar{S} = \begin{bmatrix} 1 & s_1 & s_2 & \cdots & s_k & s_1^2 & s_1 s_2 & s_1 s_3 & \cdots & s_1 s_k & s_2^2 & s_2 s_3 & \cdots & s_{k-1} s_k & s_k^2 \end{bmatrix} \tag{2.6}$$

is an augmented sensor vector which contains additive linear and quadratic combinations of the components of the sensor readings.

The N calibration sets contain more equations than unknowns. This is called an overdetermined set of linear equations [31] and can be expressed as [27]

$$L_{CAL} = \bar{S}_{CAL} \cdot C_{INV} \tag{2.7}$$

where:

$$\bar{S}_{CAL} = \begin{bmatrix} \bar{S}_1 \\ \vdots \\ \bar{S}_N \end{bmatrix}_{CAL} . \tag{2.8}$$

Equation (2.7) can be solved for the inverse calibration matrix,

$$C_{INV} = \overline{S}_{CAL}^{-1} \cdot L_{CAL}, \tag{2.9}$$

using singular value decomposition (SVD) methods presented by Lawson and Hanson [21]. Appendix A contains a detailed explanation of SVD. Press et al. [31] points out that SVD is the method of choice for solving most linear least squares problems such as this one. The linear least squares - SVD solution of (2.7) minimizes [27]

$$F_{INV} = \frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{m}\left[(\hat{q}_i)_j - (q_i)_j\right]^2 \tag{2.10}$$

where:

$\hat{q}_i$ is an approximated part location deviation in a coordinate frame axis.

b) Forward Mapping

The forward mapping process determines sensor readings from a part's location data. Analogous to the inverse mapping in the previous section, the true sensor reading vector (S) is approximated by using a second-order polynomial (curvilinear) regression model. Hence the response function for the regression model is [27]

$$S \cong \hat{S} \equiv \overline{L} \cdot C_{FOR} \tag{2.10}$$

where:

$\hat{S}$ is an estimate of the sensor reading vector,

C_{FOR} is called forward mapping calibration matrix, and

$$\overline{L} = \left[1 \quad q_1 \quad q_2 \quad \cdots \quad q_p \quad q_1^2 \quad q_1q_2 \quad q_1q_3 \quad \cdots \quad q_1q_p \quad q_2^2 \quad q_2q_3 \quad \cdots \quad q_{p-1}q_p \quad q_p^2\right] \tag{2.11}$$

is an augmented part location vector which contains additive linear and quadratic combinations of the components of the part location coordinate axes.

As in the inverse mapping, the N calibration sets have more equations than unknowns. Therefore, it is also an overdetermined set of linear equations and can be expressed as [27]

$$S_{CAL} = \overline{L}_{CAL} \cdot C_{FOR} \tag{2.12}$$

where:

$$\overline{L}_{CAL} = \begin{bmatrix} \overline{L}_1 \\ \vdots \\ \overline{L}_N \end{bmatrix}_{CAL} . \tag{2.13}$$

Then Equation (2.12) can also be solved for the forward calibration matrix,

$$C_{FOR} = \overline{L}_{CAL}^{-1} \cdot S_{CAL} , \tag{2.14}$$

using the SVD methods (see Appendix A). The linear least squares - SVD solution of (2.12) minimizes [27]

$$F_{FOR} = \frac{1}{2} \sum_{j=1}^{N} \sum_{i=1}^{k} \left[(\hat{s}_i)_j - (s_i)_j \right]^2 . \tag{2.15}$$

where:

$\hat{S}_i$ is an approximated sensor reading.

2.1.3 STEP 3: Localization

In this step, the part location is determined from the sensor data ($\overline{L}$) on the grasped part. If the inverse mapping is used, this can be simply calculated directly from Equation (2.5),

$$\hat{L} = \overline{S} \cdot C_{INV} .$$

However, if the forward mapping is used, an iterative procedure is needed to calculate the part location.

The iterative inversion of the forward mapping starts with $\hat{L}_o$, which is an initial guess for the location of the part in grasp. The initial guess is calculated from the approximate inverse mapping in Equation (2.5) [27]:

$$\hat{L}_o = \overline{S} \cdot C_{INV}. \tag{2.16}$$

Once $\hat{L}_o$ is obtained, it is substituted into the forward mapping for $\overline{L}$ in Equation (2.10),

$$\hat{S} = \overline{L} \cdot C_{FOR},$$

to obtain the estimated sensor vector, $\hat{S}$. This estimated vector is then compared with, S, the actual sensor vector. If the two sensor vectors do not agree to within some tolerance, it means that the error in the location estimate is too large and must be adjusted accordingly. An adjusted initial location called $\hat{L}_o$ is then calculated. This process is repeated iteratively until the difference between the estimated and actual sensor vectors are acceptably small [27] or until it reaches a preset number of iterations (i.e., 500).

The iterative procedure minimizes [27]

$$F = \frac{1}{2}\sum_{i=1}^{k} \lambda_i^2 \left(\hat{s}_i - s_i\right)^2 \tag{2.17}$$

where λ_i's are called scalar sensitivity weights for the sensors. These weights can be set differently for each sensor for fine tuning of the mapping performance and could reduce localization errors for some coordinate axes in the forward mapping. The minimization of Equation (2.17) can be achieved using a simple gradient algorithm which involves the gradient (∇) of F [27]:

$$\nabla F = \begin{bmatrix} \frac{\partial F}{\partial \hat{q}_1} \\ \vdots \\ \frac{\partial F}{\partial \hat{q}_k} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{p} \lambda_i \left(\hat{s}_i - s_i\right)\left(\frac{\partial \hat{s}_i}{\partial \hat{q}_1}\right) \\ \vdots \\ \sum_{i=1}^{p} \lambda_i \left(\hat{s}_i - s_i\right)\left(\frac{\partial \hat{s}_i}{\partial \hat{q}_k}\right) \end{bmatrix}. \tag{2.18}$$

where $\frac{\partial \hat{s}_i}{\partial \hat{q}_j}$ are calculated by finding the gradient (∇) of $\hat{s}$ in Equation (2.10). A detailed explanation of the gradient can be found in [31]. From the gradient, a new initial location ($\hat{L}_o$) can be calculated. First, the adjustment ($\Delta\hat{L}_o$) for the new location is obtained:

$$\Delta\hat{L}_o = -\nabla F \cdot \eta \tag{2.19}$$

where η is called a step length parameter [27] which is set equal to 0.01 in this thesis. Then the new initial location is

$$\hat{L}_o = \hat{L}_o + \Delta\hat{L}_o, \tag{2.20}$$

which replaces Equation (2.16) and is substituted back into Equation (2.10). This iterative procedure is repeated until F is smaller than a specified limit (ε) or until the procedure reaches a preset number (n) of iterations.

In the following sections, the above algorithm will be applied to three different cases of misalignment problem in the localization of the misaligned fender:

1. 3-DOF surface misalignment: misalignments are present in z, w, p coordinates.
2. 3-DOF edge misalignment: misalignments are present in x, y, r coordinates.
3. 6-DOF misalignment: misalignments are present in all six coordinates.

For each case, the application and modification of the previously formulated equations will be presented.

2.2 3 DOF Surface Localization

In 3 DOF surface localization, misalignments in only the z, w, and p coordinate axes are considered. The part location deviations, which were previously noted q_i, becomes z, w, and p specifically. The measurements are taken from only three of the seven sensors: s_1, s_2, and s_3. These three sensors are used for detecting the surface of the car fender. A detailed description of the sensors is presented in Chapter 3. Figure 2-4

shows a schematic diagram of the position of the coordinates and sensor vectors with respect to the fender. The equations shown in this section are modifications to the general part localization algorithm, formulated in Section 2.1. See Section 2.1 for the reasoning and explanation behind each step.

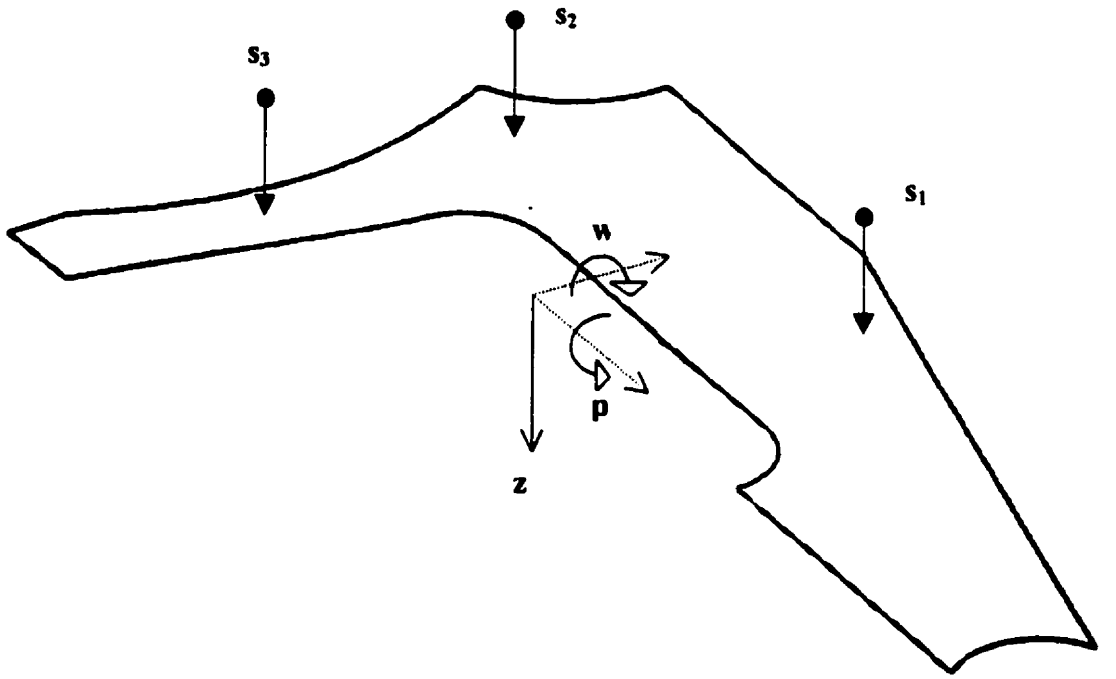

Figure 2-3: Schematic of 3 DOF Surface Localization Problem

Now the corresponding part location vector in Equation (2.2) becomes

$$L_i = \begin{bmatrix} z_i & w_i & p_i \end{bmatrix} \tag{2.21}$$

and the sensor reading vector in Equation (2.4) is given by

$$S_i = \begin{bmatrix} (s_1)_i & (s_2)_i & (s_3)_i \end{bmatrix}. \tag{2.22}$$

For the given configuration in the coordinate axes and sensors, the inverse mapping in Equation (2.5) becomes

$$\hat{L} = \overline{S} \cdot C_{INV} = \overline{S} \cdot \begin{bmatrix} a_{1,z} & a_{1,w} & a_{1,p} \\ a_{2,z} & a_{2,w} & a_{2,p} \\ a_{3,z} & a_{3,w} & a_{3,p} \\ a_{4,z} & a_{4,w} & a_{4,p} \\ a_{5,z} & a_{5,w} & a_{5,p} \\ a_{6,z} & a_{6,w} & a_{6,p} \\ a_{7,z} & a_{7,w} & a_{7,p} \\ a_{8,z} & a_{8,w} & a_{8,p} \\ a_{9,z} & a_{9,w} & a_{9,p} \\ a_{10,z} & a_{10,w} & a_{10,p} \end{bmatrix} \tag{2.23}$$

where the augmented sensor ($\overline{S}$) vector in Equation (2.6) for the above inverse mapping is rewritten as

$$\overline{S} = \begin{bmatrix} 1 & s_1 & s_2 & s_3 & s_1^2 & s_1 s_2 & s_1 s_3 & s_2^2 & s_2 s_3 & s_3^2 \end{bmatrix}. \tag{2.24}$$

Similarly, for the given configuration, the forward mapping in Equation (2.10) becomes

$$\hat{S} = \overline{L} \cdot C_{FOR} = \overline{L} \cdot \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \\ b_{4,1} & b_{4,2} & b_{4,3} \\ b_{5,1} & b_{5,2} & b_{5,3} \\ b_{6,1} & b_{6,2} & b_{6,3} \\ b_{7,1} & b_{7,2} & b_{7,3} \\ b_{8,1} & b_{8,2} & b_{8,3} \\ b_{9,1} & b_{9,2} & b_{9,3} \\ b_{10,1} & b_{10,2} & b_{10,3} \end{bmatrix} \tag{2.25}$$

where the augmented part location vector ($\overline{L}$) for the above forward mapping is also rewritten as

$$\overline{L} = \begin{bmatrix} 1 & z & w & p & z^2 & zw & zp & w^2 & wp & p^2 \end{bmatrix}. \tag{2.26}$$

The function F in Equation (2.17) becomes

$$F = \frac{1}{2}\left[\lambda_1^2\left(\hat{s}_1 - s_1\right)^2 + \lambda_2^2\left(\hat{s}_2 - s_2\right)^2 + \lambda_3^2\left(\hat{s}_3 - s_3\right)^2\right] \tag{2.27}$$

where λ_1, λ_2, and λ_3 are the scalar sensitivity weights for the sensor readings s_1, s_2, and s_3 respectively. Now the gradient (∇F) is written as

$$\nabla F = \begin{bmatrix} \frac{\partial F}{\partial z} \\ \frac{\partial F}{\partial w} \\ \frac{\partial F}{\partial p} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{3} \lambda_i\left(\hat{s}_i - s_i\right)\left(\frac{\partial \hat{s}_i}{\partial \hat{z}}\right) \\ \sum_{i=1}^{3} \lambda_i\left(\hat{s}_i - s_i\right)\left(\frac{\partial \hat{s}_i}{\partial \hat{w}}\right) \\ \sum_{i=1}^{3} \lambda_i\left(\hat{s}_i - s_i\right)\left(\frac{\partial \hat{s}_i}{\partial \hat{p}}\right) \end{bmatrix} \tag{2.28}$$

where $\frac{\partial \hat{s}_i}{\partial \hat{z}}$, $\frac{\partial \hat{s}_i}{\partial \hat{w}}$, and $\frac{\partial \hat{s}_i}{\partial \hat{p}}$ are obtained by calculating the gradient of $\hat{S}$ in Equation (2.25):

$$\nabla \hat{s}_i = \begin{bmatrix} \frac{\partial \hat{s}_i}{\partial \hat{z}} \\ \frac{\partial \hat{s}_i}{\partial \hat{w}} \\ \frac{\partial \hat{s}_i}{\partial \hat{p}} \end{bmatrix} = \begin{bmatrix} b_{2,i} + 2b_{5,i}\hat{z} + b_{6,i}\hat{w} + b_{7,i}\hat{p} \\ b_{3,i} + b_{6,i}\hat{z} + 2b_{8,i}\hat{w} + b_{9,i}\hat{p} \\ b_{4,i} + b_{7,i}\hat{z} + b_{9,i}\hat{w} + 2b_{10,i}\hat{p} \end{bmatrix}. \tag{2.29}$$

Using the above equations for the surface localization, the experimental results are given in Section 4.3. The performance analysis of the surface localization algorithm is also presented.

2.3 3 DOF Edge Localization

In 3 DOF edge localization, misalignments in only the x, y and r coordinate axes are considered. The part location deviations, which were previously noted q_i, becomes x, y, and r. The measurements are taken from four of the seven sensors: s_4, s_5, s_6, and s_7. These three sensors are used for detecting the edges of the car fender. A

detailed description of these sensors is presented in Chapter 3. Figure 2-5 shows a schematic diagram of the position of the coordinates and sensor vectors with respect to the fender. The equations shown in this section are modifications to the general part localization algorithm, formulated in Section 2.1. See Section 2.1 for the reasoning and explanation behind each step.

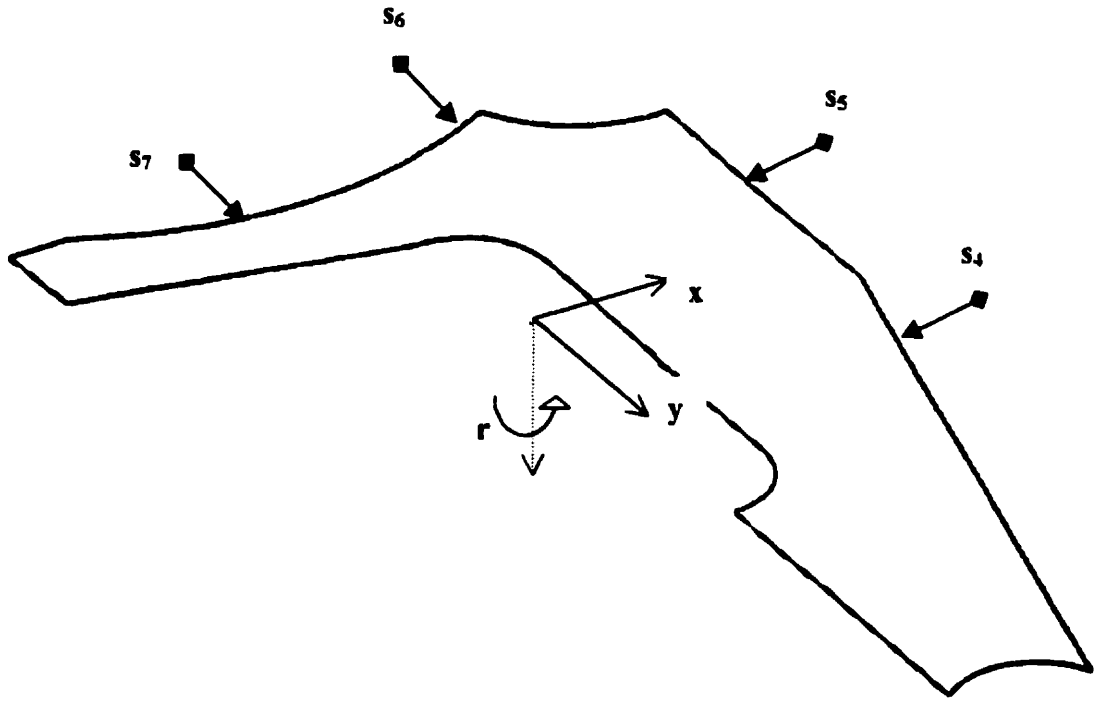

Figure 2-4: Schematic of 3 DOF Edge Localization Problem

Now the corresponding part location vector in Equation (2.2) becomes

$$L_i = \begin{bmatrix} x_i & y_i & r_i \end{bmatrix} \tag{2.30}$$

and the sensor reading vector in Equation (2.4) becomes

$$S_i = \begin{bmatrix} (s_4)_i & (s_5)_i & (s_6)_i & (s_7)_i \end{bmatrix}. \tag{2.31}$$

For the given configuration in the coordinate axes and sensors, the inverse mapping in Equation (2.5) becomes

$$\hat{L} = \overline{S} \cdot C_{INV} = \overline{S} \cdot \begin{bmatrix} a_{1,x} & a_{2,y} & a_{3,r} \\ \vdots & \vdots & \vdots \\ a_{15,x} & a_{15,y} & a_{15,r} \end{bmatrix} \tag{2.32}$$

where the augmented sensor ($\overline{S}$) vector in Equation (2.6) for the above inverse mapping is rewritten as

$$\overline{S} = \begin{bmatrix} 1 & s_4 & s_5 & s_6 & s_7 & s_4^2 & s_4 s_5 & s_4 s_6 & s_4 s_7 & s_5^2 & s_5 s_6 & s_5 s_7 & s_6^2 & s_6 s_7 & s_7^2 \end{bmatrix}. \tag{2.33}$$

Similarly, for the given configuration, the forward mapping in Equation (2.10) becomes

$$\hat{S} = \overline{L} \cdot C_{FOR} = \overline{L} \cdot \begin{bmatrix} b_{1,4} & b_{1,5} & b_{1,6} & b_{1,7} \\ \vdots & \vdots & \vdots & \vdots \\ b_{15,4} & b_{15,5} & b_{15,6} & b_{15,7} \end{bmatrix} \tag{2.34}$$

where the augmented part location vector ($\overline{L}$) for the above forward mapping is also rewritten as

$$\overline{L} = \begin{bmatrix} 1 & x & y & r & x^2 & xy & xr & y^2 & yr & r^2 \end{bmatrix}. \tag{2.35}$$

The function F in Equation (2.17) becomes

$$F = \frac{1}{2}\left[\lambda_4^2\left(\hat{s}_4 - s_4\right)^2 + \lambda_5^2\left(\hat{s}_5 - s_5\right)^2 + \lambda_6^2\left(\hat{s}_6 - s_6\right)^2 + \lambda_7^2\left(\hat{s}_7 - s_7\right)^2\right] \tag{2.36}$$

where λ_4, λ_5, λ_6, and λ_7 are the scalar sensitivity weights for the sensor readings s_4, s_5, s_6, and s_7 respectively. Now the gradient (∇F) is written as

$$\nabla F = \begin{bmatrix} \frac{\partial F}{\partial x} \\ \frac{\partial F}{\partial y} \\ \frac{\partial F}{\partial r} \end{bmatrix} = \begin{bmatrix} \sum_{i=4}^{7} \lambda_i (\hat{s}_i - s_i)\left(\frac{\partial \hat{s}_i}{\partial x}\right) \\ \sum_{i=4}^{7} \lambda_i (\hat{s}_i - s_i)\left(\frac{\partial \hat{s}_i}{\partial y}\right) \\ \sum_{i=4}^{7} \lambda_i (\hat{s}_i - s_i)\left(\frac{\partial \hat{s}_i}{\partial r}\right) \end{bmatrix} \tag{2.37}$$

where $\frac{\partial \hat{s}_i}{\partial x}$, $\frac{\partial \hat{s}_i}{\partial y}$, and $\frac{\partial \hat{s}_i}{\partial r}$ are obtained by calculating the gradient of $\hat{S}$ in Equation (2.34):

$$\nabla \hat{s}_i = \begin{bmatrix} \frac{\partial \hat{s}_i}{\partial x} \\ \frac{\partial \hat{s}_i}{\partial y} \\ \frac{\partial \hat{s}_i}{\partial r} \end{bmatrix} = \begin{bmatrix} b_{2,i} + 2b_{5,i}\hat{x} + b_{6,i}\hat{y} + b_{7,i}\hat{r} \\ b_{3,i} + b_{6,i}\hat{x} + 2b_{8,i}\hat{y} + b_{9,i}\hat{r} \\ b_{4,i} + b_{7,i}\hat{x} + b_{9,i}\hat{y} + 2b_{10,i}\hat{r} \end{bmatrix}. \tag{2.38}$$

Using the above equations for the edge localization, the experimental results are given in Section 4.2. The performance analysis of the edge localization algorithm is also presented.

2.4 6 DOF Localization

In 6 DOF localization, misalignments in all (x, y, z, w, p, r) coordinate axes are considered (Figure 2-6). The part location deviations, which were previously noted q_i, becomes x, y, z, w, p, and r specifically. The measurements are taken from all seven sensors: s_1, s_2, s_3, s_4, s_5, s_6, and s_7. The 6 DOF localization uses both the proximity sensors (s_1, s_2, s_3) and the edge sensors (s_4, s_5, s_6, s_7). See Figure 2-1 for the schematic representation of the coordinates and sensor vectors with respect to the fender. The equations shown in this section are modifications to the general part localization

algorithm, formulated in Section 2.1. See Section 2.1 for the reasoning and explanation behind each step.

Now the corresponding part location vector in Equation (2.2) becomes

$$L_i = \begin{bmatrix} x_i & y_i & z_i & w_i & p_i & r_i \end{bmatrix} \tag{2.39}$$

and the sensor reading vector in Equation (2.4) changes to

$$S_i = \begin{bmatrix} (s_1)_i & (s_2)_i & (s_3)_i & (s_4)_i & (s_5)_i & (s_6)_i & (s_7)_i \end{bmatrix}. \tag{2.40}$$

For the given configuration in the coordinate axes and sensors, the inverse mapping in Equation (2.5) becomes

$$\hat{L} = \overline{S} \cdot C_{INV} = \overline{S} \cdot \begin{bmatrix} a_{1,x} & a_{1,y} & a_{1,z} & a_{1,w} & a_{1,p} & a_{1,r} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{36,x} & a_{36,y} & a_{36,z} & a_{36,w} & a_{36,p} & a_{36,r} \end{bmatrix} \tag{2.41}$$

where the augmented sensor ($\overline{S}$) vector in Equation (2.6) for the above inverse mapping is rewritten as

$$\overline{S} = \begin{bmatrix} 1 & s_1 & s_2 & \cdots & s_7 & s_1^2 & s_1 s_2 & s_1 s_3 & \cdots & s_1 s_7 & s_2^2 & s_2 s_3 & \cdots & s_7^2 \end{bmatrix}. \tag{2.42}$$

Similarly, for the given configuration, the forward mapping in Equation (2.10) becomes

$$\hat{S} = \overline{L} \cdot C_{FOR} = \overline{L} \cdot \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} & b_{1,5} & b_{1,6} & b_{1,7} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{28,1} & b_{28,2} & b_{28,3} & b_{28,4} & b_{28,5} & b_{28,6} & b_{28,7} \end{bmatrix} \tag{2.43}$$

where the augmented part location vector ($\overline{L}$) for the above forward mapping is also rewritten as

$$\overline{L} = \begin{bmatrix} 1 & x & y & \cdots & r & x^2 & xy & xz & \cdots & xr & y^2 & yz & \cdots & r^2 \end{bmatrix}. \tag{2.44}$$

The function F in Equation (2.17) becomes

$$F = \frac{1}{2}\sum_{i=1}^{7} \lambda_i^2 \left(\hat{s}_i - s_i\right)^2 \tag{2.45}$$

where λ_1 to λ_7 are the scalar sensitivity weights for the sensors readings from s_1 to s_7 respectively. Now the gradient (∇F) is written as

$$\nabla F = \begin{bmatrix} \frac{\partial F}{\partial x} \\ \frac{\partial F}{\partial y} \\ \frac{\partial F}{\partial z} \\ \frac{\partial F}{\partial w} \\ \frac{\partial F}{\partial p} \\ \frac{\partial F}{\partial r} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{7} \lambda_i \left(\hat{s}_i - s_i\right)\left(\frac{\partial \hat{s}_i}{\partial x}\right) \\ \sum_{i=1}^{7} \lambda_i \left(\hat{s}_i - s_i\right)\left(\frac{\partial \hat{s}_i}{\partial y}\right) \\ \sum_{i=1}^{7} \lambda_i \left(\hat{s}_i - s_i\right)\left(\frac{\partial \hat{s}_i}{\partial z}\right) \\ \sum_{i=1}^{7} \lambda_i \left(\hat{s}_i - s_i\right)\left(\frac{\partial \hat{s}_i}{\partial w}\right) \\ \sum_{i=1}^{7} \lambda_i \left(\hat{s}_i - s_i\right)\left(\frac{\partial \hat{s}_i}{\partial p}\right) \\ \sum_{i=1}^{7} \lambda_i \left(\hat{s}_i - s_i\right)\left(\frac{\partial \hat{s}_i}{\partial r}\right) \end{bmatrix} \tag{2.46}$$

where $\frac{\partial \hat{s}_i}{\partial x}$ to $\frac{\partial \hat{s}_i}{\partial r}$ are obtained by calculating the gradient of $\hat{S}$ in Equation (2.43):

$$\nabla \hat{s}_i = \begin{bmatrix} \frac{\partial \hat{s}_i}{\partial x} \\ \frac{\partial \hat{s}_i}{\partial y} \\ \frac{\partial \hat{s}_i}{\partial z} \\ \frac{\partial \hat{s}_i}{\partial w} \\ \frac{\partial \hat{s}_i}{\partial p} \\ \frac{\partial \hat{s}_i}{\partial r} \end{bmatrix} = \begin{bmatrix} b_{2,i} + 2b_{8,i}\hat{x} + b_{9,i}\hat{y} + b_{10,i}\hat{z} + b_{11,i}\hat{w} + b_{12,i}\hat{p} + b_{13,i}\hat{r} \\ b_{3,i} + b_{9,i}\hat{x} + 2b_{14,i}\hat{y} + b_{15,i}\hat{z} + b_{16,i}\hat{w} + b_{17,i}\hat{p} + b_{18,i}\hat{r} \\ b_{4,i} + b_{10,i}\hat{x} + b_{15,i}\hat{y} + 2b_{19,i}\hat{z} + b_{20,i}\hat{w} + b_{21,i}\hat{p} + b_{22,i}\hat{r} \\ b_{5,i} + b_{11,i}\hat{x} + b_{16,i}\hat{y} + b_{20,i}\hat{z} + 2b_{23,i}\hat{w} + b_{24,i}\hat{p} + b_{25,i}\hat{r} \\ b_{6,i} + b_{12,i}\hat{x} + b_{17,i}\hat{y} + b_{21,i}\hat{z} + b_{24,i}\hat{w} + 2b_{26,i}\hat{p} + b_{27,i}\hat{r} \\ b_{7,i} + b_{13,i}\hat{x} + b_{18,i}\hat{y} + b_{22,i}\hat{z} + b_{25,i}\hat{w} + b_{27,i}\hat{p} + 2b_{28,i}\hat{r} \end{bmatrix}. \tag{2.47}$$

Using the above equations for the 6 DOF localization, the experimental results are obtained in Section 4.4. The performance analysis of the 6 DOF localization algorithm is also presented.

2.5 Summary

This chapter presents the development of a part localization algorithm with reference to the method presented in [27]. The formulated algorithm is intended for the purpose of localizing car body fenders. In addition, the modifications to the algorithm are described in detail in application to three cases of misalignment problem in the localization of the misaligned fender: 3 DOF surface misalignment, 3 DOF edge misalignment, and 6 DOF misalignment.

Chapter 3

APL System Description

3.1 Introduction

This chapter describes the development of Automatic Part Localization System (APLS) in the Laboratory for Nonlinear Systems Control. This is a demonstration system which was built to prove the feasibility of the proposed part localization concept. APLS comprises the following hardware and software components:

1. Robot System Hardware
    a) Fanuc S-110 Robot
    b) Karel R-F Controller
    c) Vacuum-Cup Gripper
2. Sensor System Hardware
    a) NaiS LM100 Laser Beam Proximity Sensors
    b) NaiS UZL110 Laser Beam Edge Sensors
    c) Daytronic System 10 Data Acquisition System
    d) Sensor Positioning Frame
3. Software Modules
    a) Software Modules in Host Computer (PC):

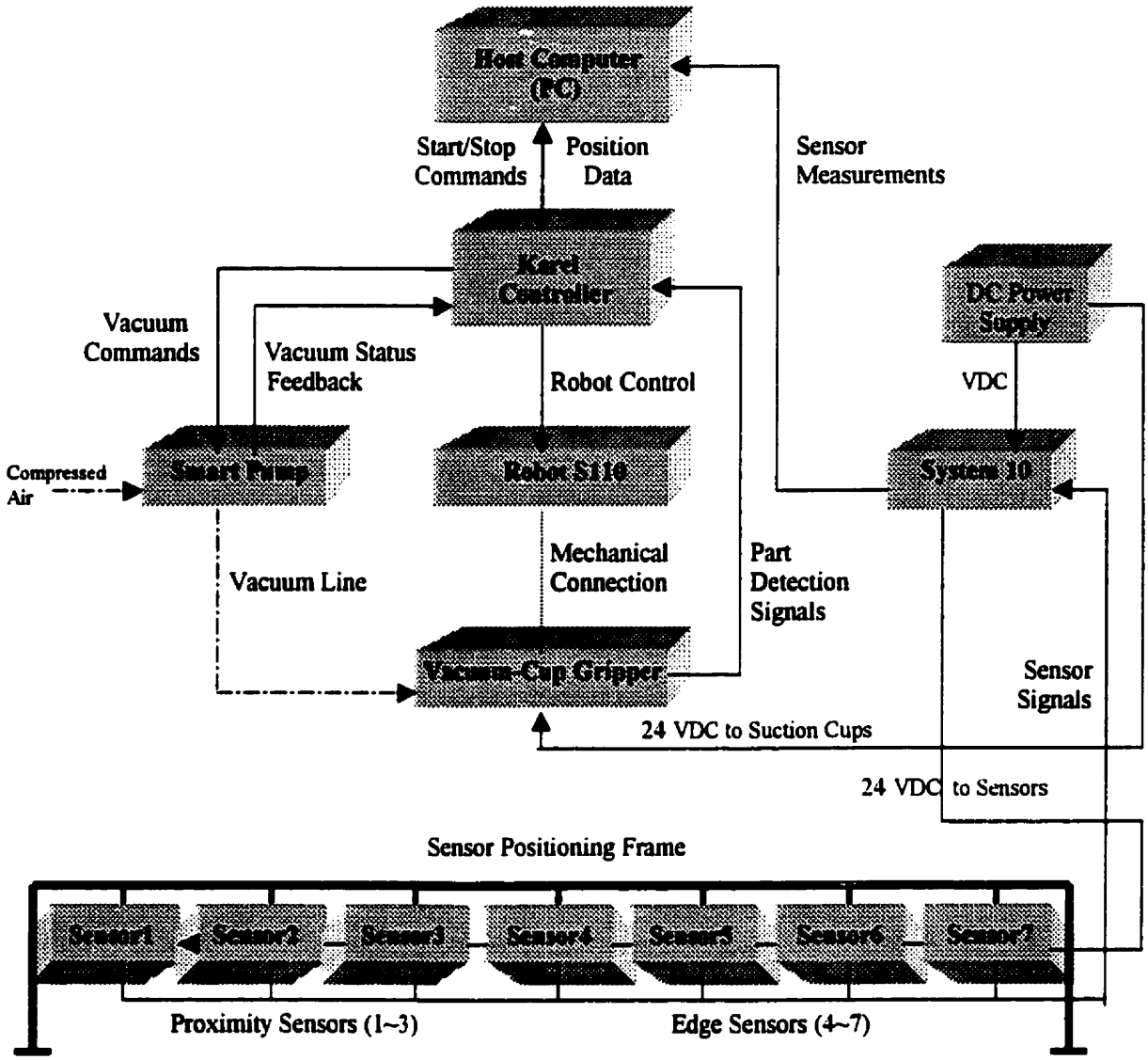

Figure 3-1: Schematic Block Diagram of APLS's Hardware

Borland C/C++, Karel Communication Software, Visual Designer and MATLAB

b) Software Modules in Karel Controller:

Karel Programming Language

Figure 3-1 presents a schematic description of the APLS's hardware interaction in block diagram. In the following sections, the functions of the above listed main and sub components are described in detail. In addition, APLS's overall part localization procedure and experimental steps are thoroughly presented.

3.2 Robot System Hardware

3.2.1 Robot

The Laboratory for Nonlinear Systems Control (LNSC) is equipped with a FANUC S-110 R-type industrial robot. The S-110 is a floor mounted, six axes of motion robot. A diagram of the robot is provided in Figure 3-2. It has a maximum load capacity of 10 kg at its wrist. This maximum payload allows the robot to lift both the car fender and vacuum gripper. The S-110 has a repeatability of $\pm 0.2mm$ at maximum speed and extension.

The six axes motion are the base rotation (1), waist bend (2), shoulder bend (3), wrist pitch (4), arm rotation (5), and wrist roll (6). Each axis is driven by a DC servo motor via a precision harmonic reducer (axis 1, 4, 5, 6) or a precision ball screw and nut (axis 2, 3). The robot calculates positions to which the faceplate of end effector moves relative to a user-defined coordinate frame called the user frame. The position of the user frame can be defined relative to the world coordinate frame. The world coordinate frame is a fixed frame inside of the robot, at the intersection of axis 1 and axis 2.

Figures 3-2 and 3-3 were obtained from [12]

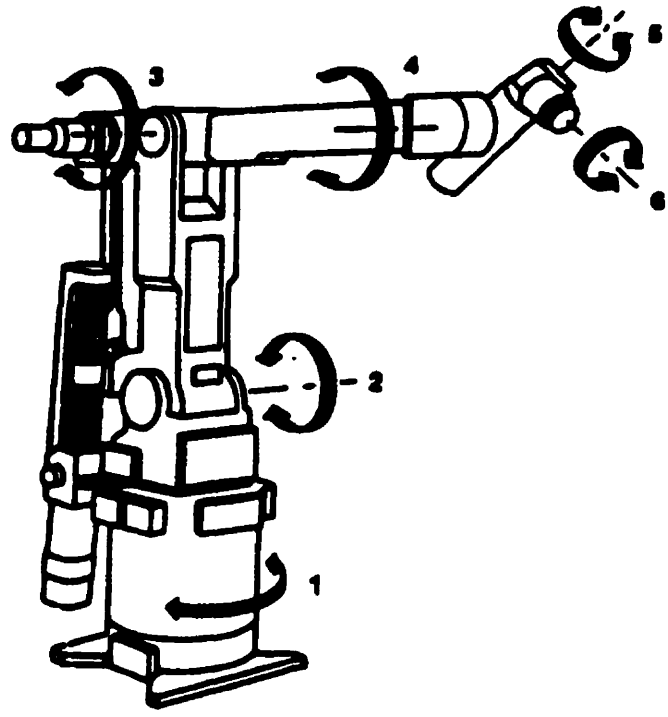

Figure 3-2: FANUC S-110 R-type Industrial Robot

3.2.2 Karel Robot Controller

The S-110 robot is controlled by a FANUC Karel R-F controller. The Karel controller, shown in Figure 3-3, is a simultaneous-axis control system. It incorporates multiple Motorola 68,000 microprocessors, several types of memory (RAM and Bubble Memory), and input/output (I/O) devices. The controller electronics, supported by the system software, directs the operation and motion of the robot and allows communication with peripheral devices. In this thesis, the peripheral devices are a host computer and vacuum generator. The controller is able to communicate with the host computer, a Pentium 166 MHz, through one of its I/O ports. It is equipped with a RS-232-C serial port connector for interfacing with the computer. Both devices are directly cabled together via a null-modem cable. A detailed description about the serial communication and host computer is presented in Section 3.4.

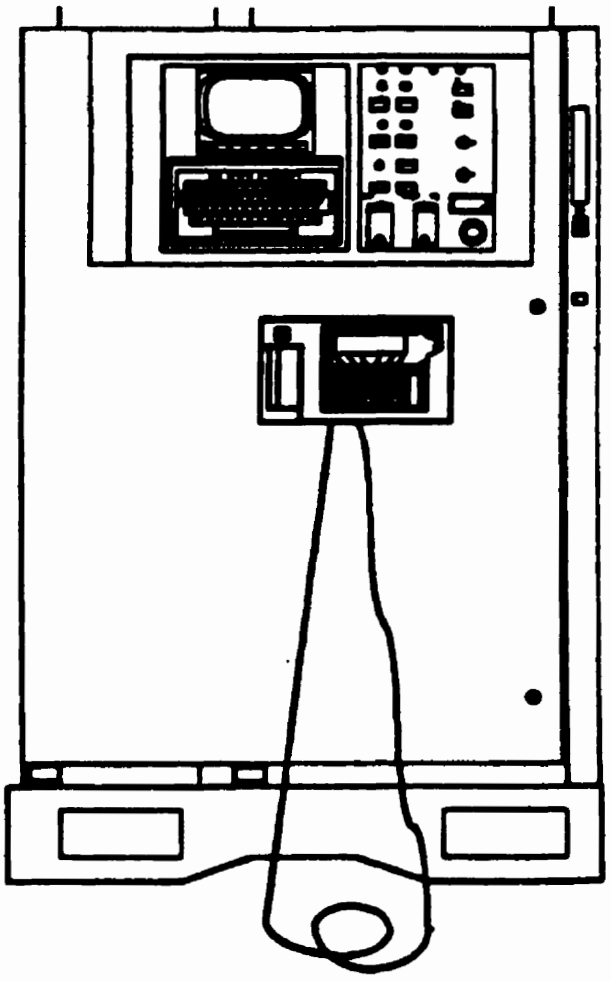

Figure 3-3: Karel R-F Controller

The Karel controller has an I/O rack, so that additional I/O modules can be added. A wide variety of I/O modules are available from FANUC to allow customization of the robot to the individual requirements of the application. To facilitate the implementation of APLS, three additional modules have been inserted: one DC digital input module (ID08C), one DC digital output module (OD08C), and one AC digital input module (IA16E). These added modules allow the operation of a vacuum-cup gripper from the controller.

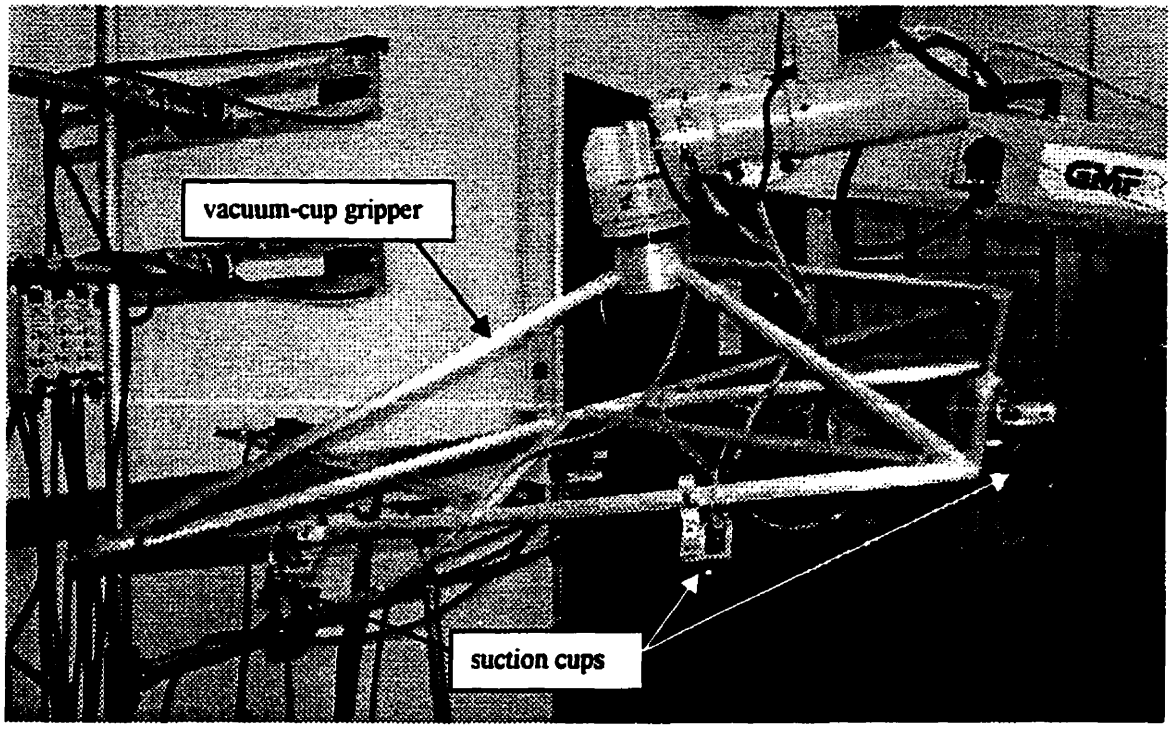

Figure 3-4: Vacuum-Cup Gripper

3.2.3 Vacuum-Cup Gripper

The vacuum gripper was designed and built in the laboratory by Tran [34]. The gripper is able to support up to 18 kg. As shown in Figure 3-4, it is comprised of three components: 3 knobbed suction cups with sensors, an aluminum structure, and a vacuum generator. A technical drawing of the knobbed suction cup is presented in Figure 3-5. The sensor is a metal detecting proximity switch that turns ON in the presence of the part. The power (24VDC) to the switch is provided by an external variable voltage power supply unit (OMRON S82L). When the suction cup in the gripper touches the part, the proximity switch sends DC signal to the Karel controller. The controller then, starts the operation of the vacuum generator. The generator provides vacuum for the three suction cups via a vacuum distribution manifold. Figure 3-6 is a schematic diagram of vacuum pressure line and electrical connections in the vacuum-cup gripper.

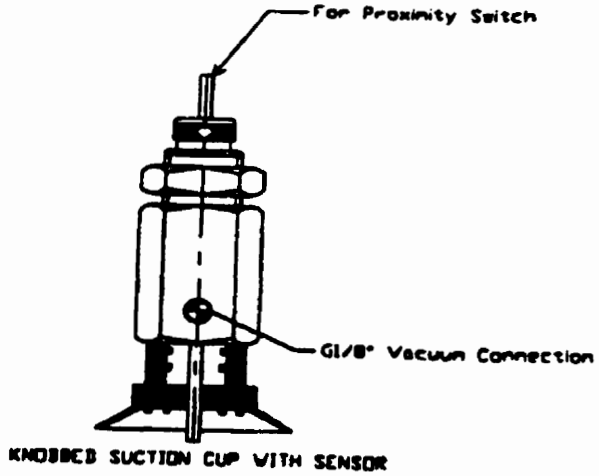

Figure 3-5: Knobbed Suction Cup with Proximity Switch

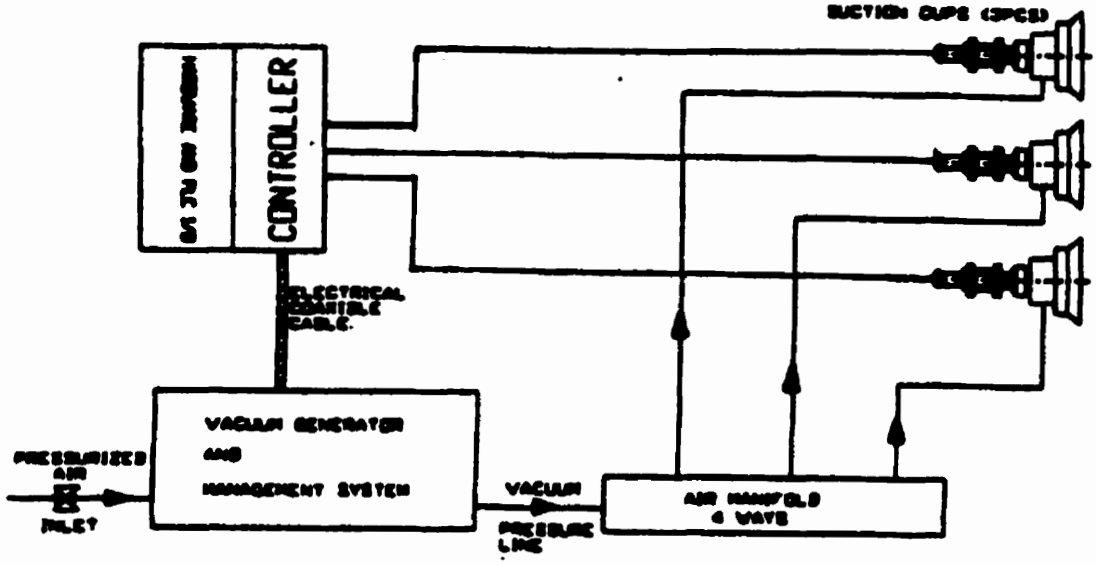

Figure 3-6: Schematic of Vacuum and Electrical Connections in Vacuum-Cup Gripper

Vaccum Line Electrical Line

Figure 3-4 was obtained from [34]

The following is the sequence of operation for the vacuum-cup gripper:

1. The robot moves the gripper near the fender.
2. The Karel controller starts to monitor the feedback signals from the proximity switches on each of the three suction cups.
3. The robot moves the gripper onto the part.
4. When the suction cups touch the part and signals from all proximity switches become ON, the controller turns ON the vacuum signal of the generator. This mode of operation is called Vacuum Mode.
5. While the generator supplies vacuum to the suction cups, the controller monitors feedback signals for vaccum status in the vacuum chamber. When vacuum level reaches a pre-selected level, the vacuum status becomes ON.
6. When the vacuum status becomes ON, the robot moves the part to a new location for the part localization sequence.
7. The part is released when the controller sets the generator's Vacuum Mode OFF. If the vacuum generator's sensing circuitry detects vacuum, pressurized air is automatically sent into the chamber to destroy the vacuum. This is mode of operation is called Automatic Blowoff Mode.
8. The controller also can place the generator into blowoff by turning ON the Blowoff signal. This mode of operation is called Forced Blowoff Mode.

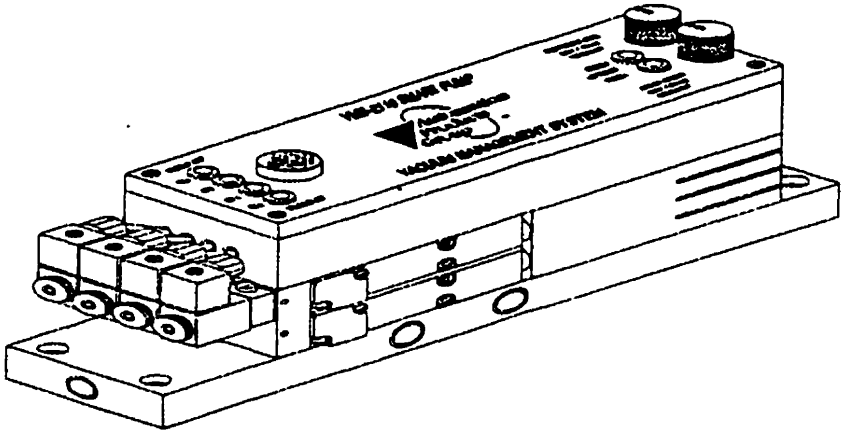

Figure 3-7: ISI Automation VMS-2110 Smart Pump

The vacuum generator is an ISI Automation's Smart Pump® (Figure 3-7). It is called a "smart pump" because of its intelligent material handling capabilities. The smart pump is able to automatically monitor vacuum level, maintain and compensate for correct vacuum level, and "sense" when it has picked up the part. It also combines an exhaust silencer which allows compressed air exiting the pump to be quieted (73 dBA). The electronic control of the pump is provided by the Karel controller via a PLC cable to its input/output interface.

3.3 Sensor System Hardware

The sensor system consists of 7 sensors: 3 proximity sensors and 4 edge sensors. Both type of sensors utilize a laser beam as the light source. These sensors are mounted on a metal frame called a sensor positioning frame (SPF), which is specifically designed for the application of APLS. The sensor system also includes System 10 which is a data acquisition mainframe that preprocesses the signals from the sensors. All these components are described thoroughly in the following sections.

3.3.1 Proximity Sensors

In APLS, three NAiS™ LM100 laser beam sensors - called proximity sensors in this thesis - are used to detect the location of surface of the part. A diagram of the proximity sensor is provided in Figure 3-8. The proximity sensors are diffuse-deflective type sensors. Referring to Figure 3-9, the word "diffuse-deflective type" refers to the method of detection which the laser beam projecting section and the receiving section are set either co-axially or with their axes closely adjacent. As a target passes in front of the sensor or approaches within its measuring range to cause the laser beam to be reflected, the distance between the target and sensor is detected. According to the sensor's manual [2], the laser beam is emitted from a light emitting element (semiconductor laser) which passes through the laser aperture to the part (fender) in the grasp of the robot. When the

laser beam makes contact, a part of the diffuse-reflected laser beam passes through the receiver lens to a spot on a position sensitive device inside the sensor. The position of the light spot varies according to the detection distance and from this variance, the distance of the fender is measured. The distance is interpreted as analog output ($\pm 5V$) and sent to System 10, the data acquisition system. The dotted arrow in Figure 3-9 is the direction of the distance measurement taken. As it is shown, the proximity sensor measures the surface of the fender.

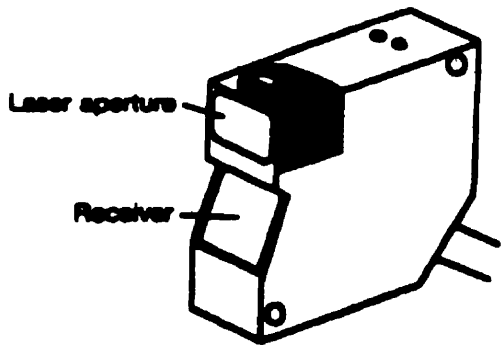

Figure 3-8: NAiS™ LM100 Laser Beam Sensor (Proximity Sensor)

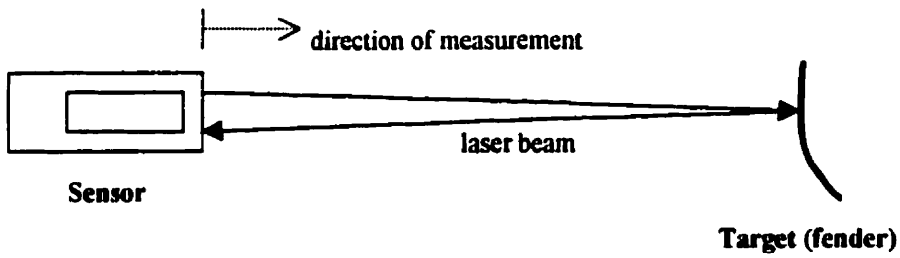

Figure 3-9: Method of Detection for Proximity Sensor

The measurable range of the proximity sensor is from $50mm$ to $100mm$ away from the sensor. One of the main features of the sensor is that it is insensitive to roughness, reflectivity (glossiness), or different colors in the part [2]. Also, its high-precision lens provides excellent linearity characteristics. The manual specifies that the repeat accuracy (resolution) is $0.02mm$ or less at a selected response time of $40ms$. The material error, which is related to the material of the target, for the fender (steel plate) is suggested as about $\pm 0.125mm$.

3.3.2 Edge Sensors

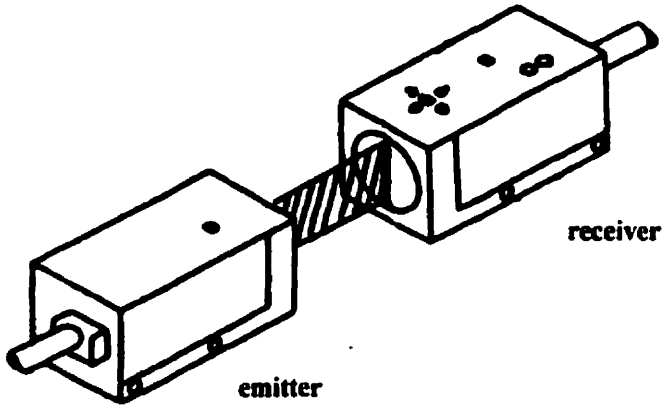

Figure 3-10: NAiS™ UZL110 Laser Beam Sensor (Edge Sensor)

The APLS's sensor system is also includes four NAiS™ UZL110 laser beam sensors, called edge sensors in this thesis. These sensors are used detect the location of edges of the part. A diagram of the sensor is provided in Figure 3-10. The edge sensors are through beam type sensors. As Figure 3-11 shows, the word "through beam type" refers to an another method of detection which the sensor is comprised of a laser beam emitter and a laser beam receiver. The emitter and receiver are positioned opposite to

each other. When the fender passes between the two, some of the emitted laser rays gets blocked, and hence the received laser beam level at the receiver end changes. The receiver produces analog voltage output of $5V$ when all beams are received and $1V$ when all beams are blocked. The output voltage is processed by the data acquisition system. Referring to Figure 3-11, note that the direction of measurement (dotted arrow) is different from the case of the proximity sensor. The measurable range is $15mm$. The emitter and receiver can be placed apart up to $500mm$.

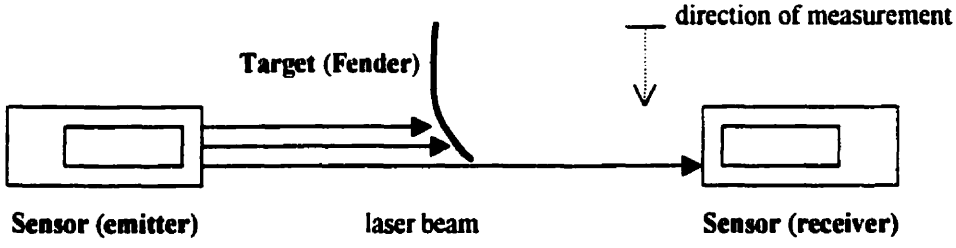

Figure 3-11: Method of Detection for Edge Sensor

The emitter projects invisible laser beam. The operating manual [1] specifies that it has repeat accuracy (resolution) of $0.01mm$ or less. The response time is $0.5ms$ or less. The through beam type sensors are not affected by the material of the target; therefore, there is no specification for the material error as in that of proximity sensor.

Figure 3-8 and Figure 3-10 were obtained from [3] and [1] respectively

3.3.3 Data Acquisition System

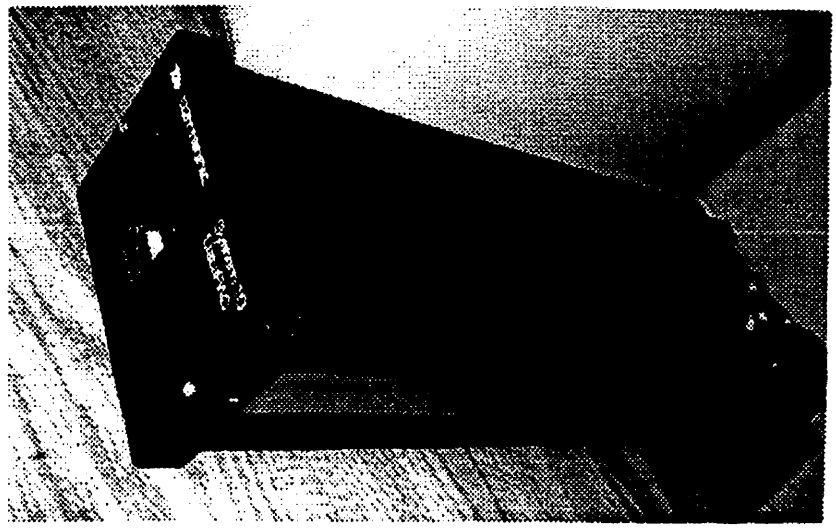

Figure 3-12: Daytronic System 10 KU-KD Data Acquisition Mainframe

The Daytronic System 10 (10KU-KD) is an external bus data acquisition and control hardware (Figure 3-12). To interface it with the proximity and edge sensors, an eight channel, DC voltage conditioning card (10A64-8C) is installed into System 10's mainframe via its one of four available "slots". Since there are seven sensors in APLS, only seven of the eight channels are used. The conditioning card preprocesses the signals from the different sensors to make them compatible to System 10. Besides the eight channels, the card also includes 2-wire DC voltage sources, which provide 24VDC power to all seven sensors. The two wires require their own power supply, which is provided by a variable voltage power supply unit (OMRON S82L). In addition, an universal logic I/O card (10AIO-16) is installed to allow System 10 to perform automatic safety monitoring and other programmable logic control functions in connection with the sensors. A detailed description of System 10 and its functional cards can be found in [7].

The communication with the host computer (PC) is done through a RS232 serial port to COM3 of the PC. The fundamental function of System 10 is to convert the analog

voltage inputs from the sensors into corresponding digital format. The System 10's analog-to-digital (A/D) converter transforms the original analog sensor signals into computer-readable data (a digital binary code). In addition to the A/D converter, several components such as an amplifier, a multiplexer, timing and synchronization circuits are included in System 10 to obtain optimum sensor data. The obtained data, which is converted to *mm*, is sent to the PC through the serial communication port. A software called Visual Designer handles the necessary communications protocols and accesses the data in the PC. Visual Designer is described in Section 3.4.2.

3.3.4 Sensor Positioning Frame (SPF)

All proximity and edge sensors are mounted on a sensor positioning frame (SPF). The use of the SPF allows the part's location data from all sensors to be obtained simultaneously. After the robot picks the fender up, it brings the part to the SPF, close to the sensors. Then, at the command of the PC, measurements from all sensors are taken together. The SPF is a cost-effective component, made of aluminum pipes with chrome finish. The pipes have outer diameter of $26.7mm$ and inner diameter of $19.05mm$. This system is shown in Figure 3-13. It has the following approximate dimensions: height = $2.1m$, width = $1.1m$, and depth = $0.4m$. The pipes are connected to each other using several kinds of pipe fittings. These pipe fittings are used for the following purposes: to firmly secure the connection between the pipes, to allow easy modification of the frame for different applications (parts) and, most importantly, to allow accurate positioning of the sensors' laser beams on the fender. The last part is done by using a set of "crossover" fitting and aluminum "leg" (Figure 3-13). As it can be seen, there are a total number of four such sets on the frame. At the end of each leg, a $0.66m$ long aluminum plate is installed. All sensors are bolted to these sensor mounting plates. Figure 3-14 shows how a set of edge sensor and a proximity sensor are positioned on the plate. The custom made L-shaped mounting brackets allow to make the laser beam alignments accurate and easy. The combination of the above components permits the positioning of the sensors in any direction.

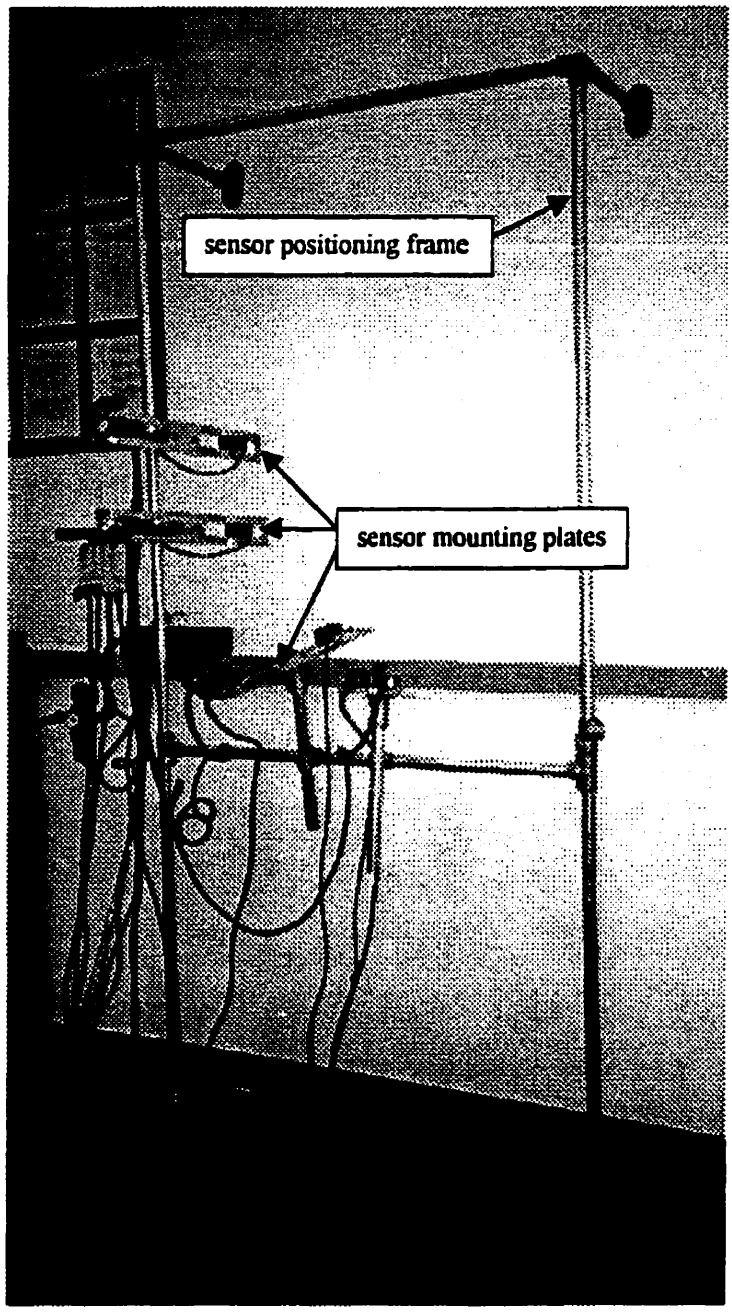

Figure 3-13: Sensor Positioning Frame (SPF)

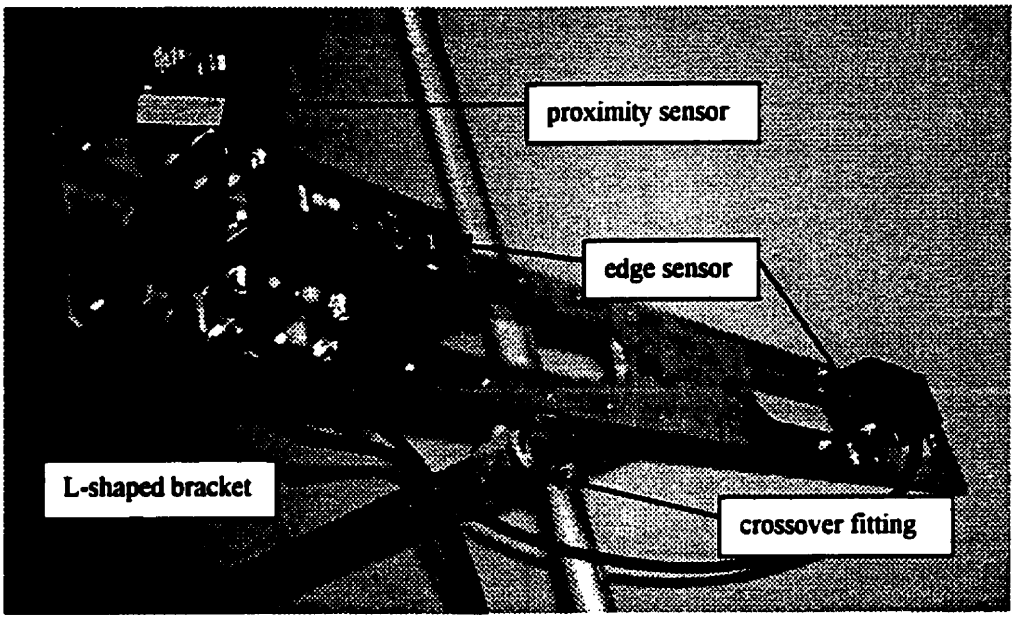

Figure 3-14: Close Up View of Sensor Mounting Plate

3.4 Software Architecture

Figure 3-15 shows all major APLS software components and how they interact with each other. In the following sections, the functions of each component are described in detail.

3.4.1 Software Modules in Host Computer (PC)

A. Borland C/C++

Borland C/C++ is the main component of APLS's software architecture in the PC. There are two main source codes that are programmed in C: *part_cal.c* and *part_loc.c*. These codes are compiled using Borland C/C++ compiler (Version 4.0) to create the corresponding *.exe* executable files. All other software modules in PC are directed by these two modules.

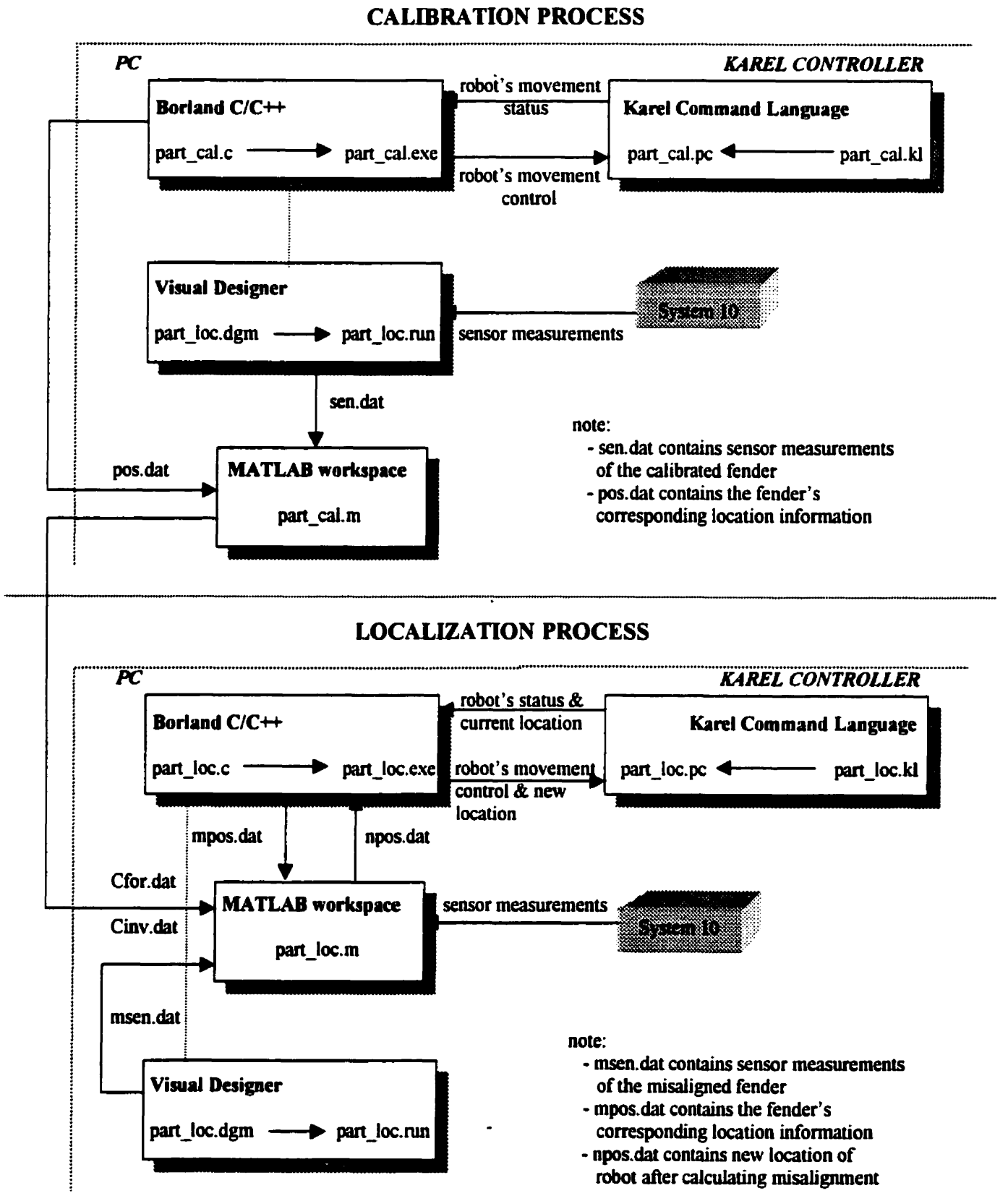

Figure 3-15: APLS's Software Architecture

The two source codes contain routines that allow the PC to communicate with a Karel controller's program via a null modem cable. They are specific to COM1 of the PC in this thesis; but, can easily be switched to other comports if necessary. The routines are able to send or receive character or numeric data between the two devices. They are also capable of confirming for correct transmission or reception of data.

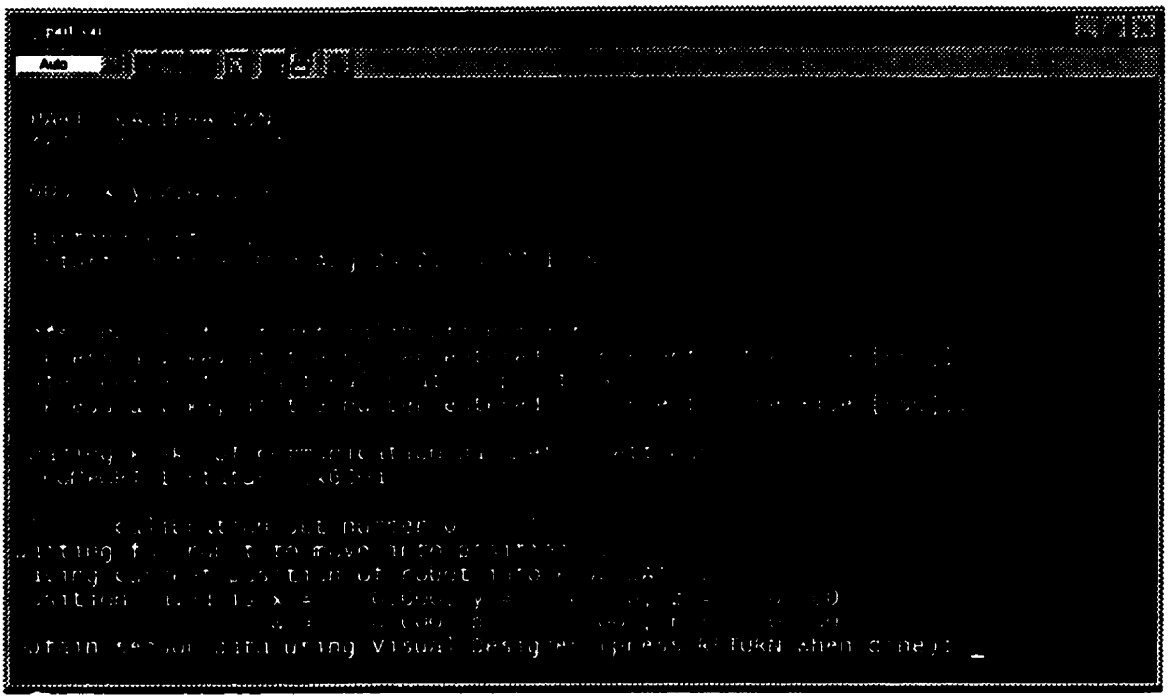

Figure 3-16: *Part_cal.exe* User-Interface

Part_cal.c is written to direct the process of collecting the calibration data (see Section 2.1.1 for details). It continuously monitors the movement of the robot and commands other software modules in the PC accordingly. The corresponding program in the robot's Karel controller is called *part_cal.pc*. By making connection to the program, the communication routines receive the information on initiation and termination of the robot's each movement from the Karel controller. The location measurements of the fender are entered manually and the corresponding sensor measurements are obtained through Visual Designer. The number of measurement sets taken is decided by the

number of calibration sets required by the specific localization problem. The location and sensor measurements of the fender are saved in *.dat* data files for later use by MATLAB (*part_cal.m*) in the mapping of the two.

Part_loc.c is written to direct the actual process of locating the fender (see Section 2.1.3 for details). Its purpose is also to monitor the movements of the robot and control the process by directing the operation of the other software modules. The corresponding program in the robot's Karel controller is called *part_loc.pc*. When the robot picks the misaligned fender up and brings it to the view of the sensors, the code enables for the PC to receive the current coordinates of the robot from the Karel controller and the corresponding sensor data via Visual Designer. Then, MATLAB (*part_loc.m*) is called to calculate the misalignment of the fender. Based on the calculation, new coordinates of the robot is determined to compensate for the fender's misalignment. The new coordinates are sent back to the Karel controller to move the robot accordingly.

B. Karel Communications System (KCS)

The PC can also communicate with the Karel controller without direct interaction with a Karel program [12] in the controller. This is done by installing Fanuc Robotics' Karel Communications System (KCS) into the PC's DOS environment. KCS allows the following common operations on the Karel controller [11]:

- Reading the Karel controller's file directory
- Transferring files between the PC and Karel controller
- Starting and stopping a Karel program
- Monitoring for unsolicited alarms and
- Obtaining the current status of the Karel controller

To exercise all the above functionality of KCS in a user-friendly manner, an interactive program called Karel Access Program (KAP) is available in KCS. In APLS, KCS is mainly used to transfer Karel program source codes (*.kl* files) and translated p-codes (*.pc* files). A brief description of the Karel programs is given in Section 3.4.2.

To run KCS in the PC, it requires the GMFCOMM communications program to be installed first. GMFCOMM is a communications software program which implements and sets up the communications protocols needed to run KCS. Once installed, GMFCOMM can be invoked in DOS environment, in a manner similar to the way other DOS accessed files are opened. When it is opened, GMFCOMM itself executes any steps necessary to make the connection to the Karel controller – it implements a communications protocol called DDCMP (Digital Data Communications Messaging Protocol). This protocol remains in the memory until the PC is shut down or restarted. DDCMP provides synchronized data transfer, which ensures that both ends of the communications link are timely correct and performs an error detection and recovery functions in the data transfer [13].

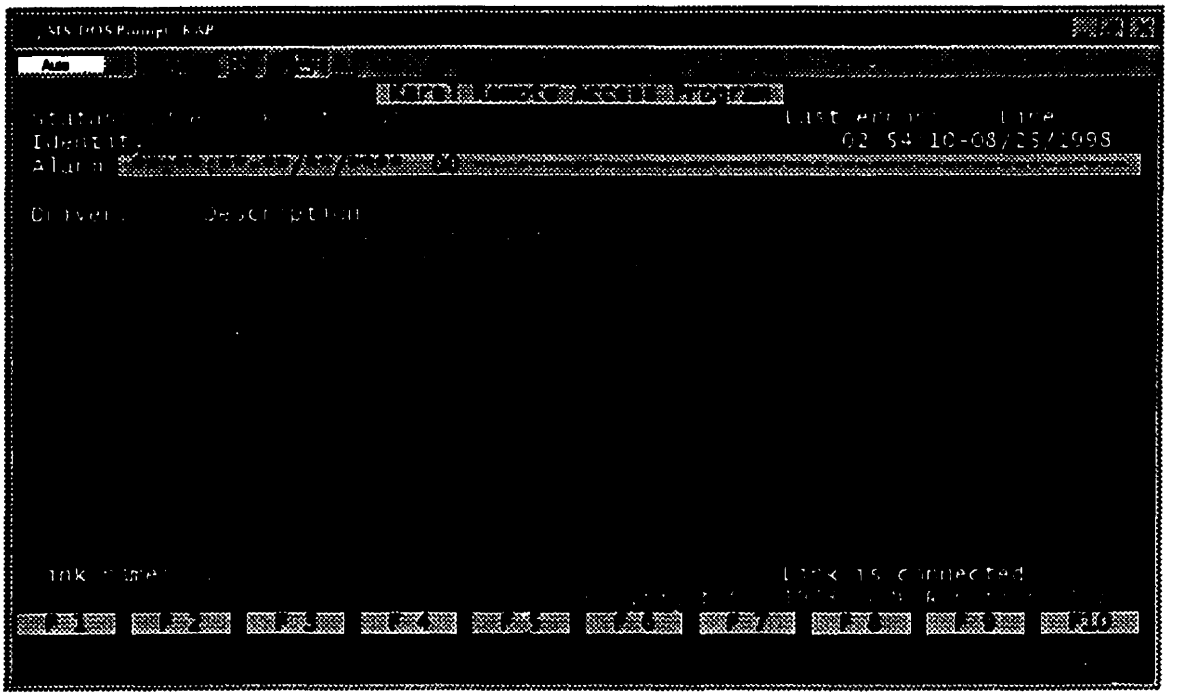

Figure 3-17: KAP User-Interface

In addition, to run KCS, the Karel controller must have DDCMP and NCP protocols running on the RS-232-C serial port. NCP (Network Command Processor) is a different kind of protocol that deals with such functions as file access and file transfer methods [13]. NCP is the entity in the Karel controller that KCS communicates with. DDCMP and NCP are resident in the Karel controller's memory and can be invoked at anytime. Once the controller is set and the null model cable is installed between the PC and controller, GMFCOMM can be started, followed by KAP.

C. Visual Designer

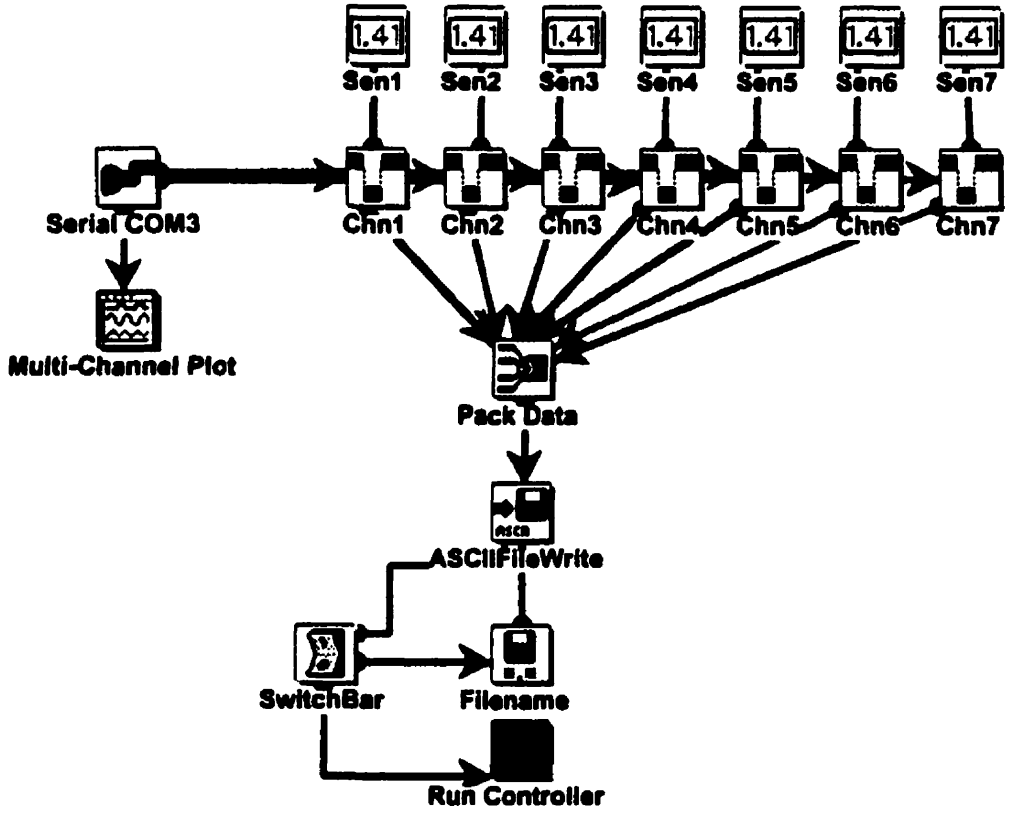

Figure 3-18: Visual Designer's Customized FlowGram (*part_loc.dgm*) for APLS

Visual Designer™ is a PC-based data acquisition, analysis, display and control software. Visual Designer allows to interface the PC to signals from the seven laser sensors via System 10. It is a Windows-based, block-diagram development environment that requires no programming. Visual Designer consists of two sub-programs: DIAGRAM and RUN. In the DIAGRAM program, block diagrams or "FlowGrams" are drawn to generate a custom application. Figure 3-18 is the built custom application for APLS, called *part_loc.dgm*. As it is shown, FlowGrams are built with block functions and interconnections. The block functions are first selected and positioned in appropriate locations and, then interconnected using "wires" to define the application. Once the FlowGram is created, DIAGRAM processes the FlowGram into an executable form called "FlowCode". The RUN program, which can be invoked from DIAGRAM, loads and executes this FlowCode.

Described in the following are the Visual Designer block functions that together provide the custom application for APLS. Referring to the FlowGram in Figure 3-16, the *Serial COM3* block allows communication with the external System 10 via the PC's COM3 serial port. System 10 provides the converted digital data from the seven laser sensors as its input. The output of the *Serial COM3* is a multi-channel buffer containing the readings from all seven sensors. The *Multi-Channel Plot* block displays each channel to an oscilloscope screen. The next are the *Chn1* to *Chn7* blocks which receive the multi-channel buffer as their inputs. Each block extracts a segment, single-channel buffer, out of the input buffer and places it in the output buffer. The single-channel buffer contains the reading from one of the sensors. For example, the *Chn1* block extracts Sensor 1's reading, the *Chn2* block extracts Sensor 2's reading, and is continued until the *Chn7* block extracts Sensor 7's reading. Each *Sen* block (*Sen1* to *Sen7*) displays the corresponding readings as a single numeric value in a digital panel. The *Pack* block then combines the seven single-channel buffers into a multi-channel output buffer again. Then, the *ASCIIFileWrite* block saves the data to a file in ASCII. The filename is generated by the *Filename* block. When the RUN program is invoked for the FlowGram, Figure 3-19 is obtained as a result. Figure 3-19 is the dialogue box used during the part localization.

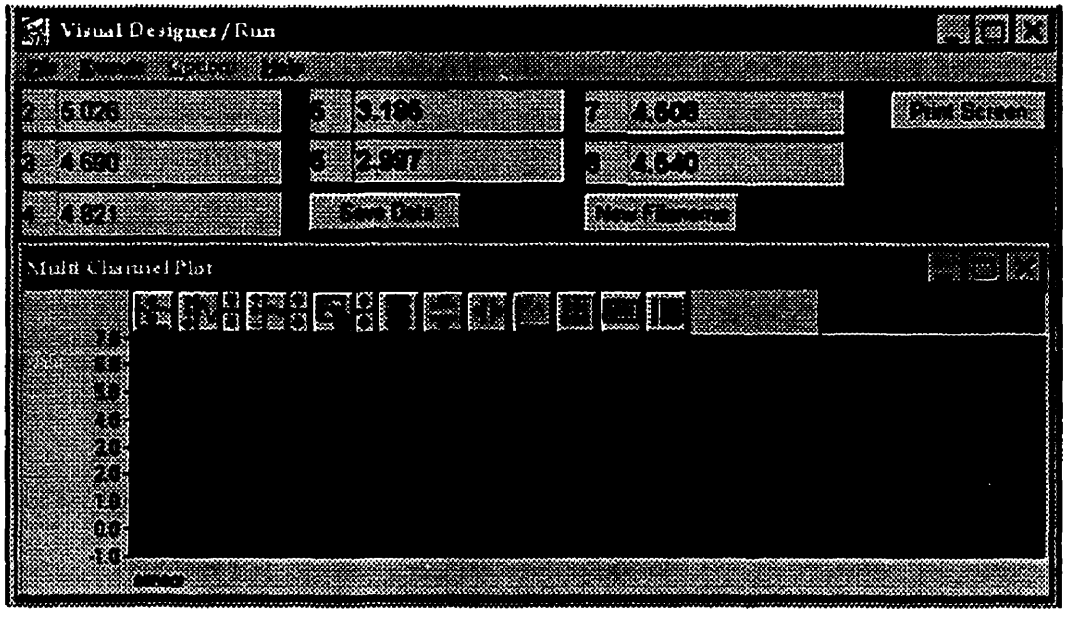

Figure 3-19: Visual Designer's RUN Program for APLS

D. MATLAB

The part localization algorithm is developed in MATLAB™. See Chapter 2 for the details of the algorithm. MATLAB by The MathWorks™ is a standard tool in the control research community, so no further explanation will be given here. In APSL, MATLAB handles all calculations necessary in finding location of the fender. There are mainly two functions that are implemented in MATLAB *.m* script files: *part_cal.m* and *part_loc.m*. These functions are called in the MATLAB workspace.

Part_cal.m is the function needed in the Mapping stage (see Section 2.1.2) of the algorithm. It solves for the inverse calibration matrix, C_{INV}, and the forward calibration matrix, C_{FOR}. The necessary calibration data, both the fender's location and sensor measurements, are acquired by Borland C/C++ (*part_cal.c*) and Visual Designer (*part_loc.dgm*) respectively. The calibration data are saved in *.dat* data files and loaded

into the *part_cal.m* function, when the function is called. The obtained calibration matrices are also saved in data files as *cinv.dat* and *cfor.dat* for part_loc.m.

Part_loc.m is the function used in the Localization stage (See section 2.1.3). This function receives, as its input, the current sensor readings in the *.dat* files from Visual Designer (*part_loc.dgm*) and the above calibration matrices, *cinv.dat* and *cfor.dat*. The purpose of the *part_loc.m* function is to determine the location misalignment of the fender from the input. Once the misalignment is found, the information is saved in a *.dat* file for to be retrieved by Borland C/C++ (*part_loc.c*). *Part_loc.c*, then, sends an adjusted location of the robot to the Karel controller.

3.4.2 Software Modules in Karel Controller

The Karel programming language is used to program the robot for the applications required by APLS. The Karel language is developed specifically for robot applications on the Karel controller, but is similar to other basic programming languages. The Karel language program source codes are stored in *.kl* script (ASCII) files. There are two source codes developed for APLS: *part_cal.kl* and *part_loc.kl*. To run these programs in the controller, they must be translated into "p-code" for the controller to understand. The Karel Translator changes the source codes from *.kl* files into the p-code for *.pc* files. Hence the two source codes become *part_cal.pc* and *part_loc.pc* respectively. Both the controller and PC have the translator installed in them. Because of slow processing time and translating speed of the controller, the APSL's programs are developed in the PC using Microsoft WordPad (Version 1.0). Then, after translation, the *.pc* files are transferred to the Karel controller using Karel Communications Software (KCS) which were described previously.

Part_cal.pc and *part_loc.pc* contain routines that allow sending or receiving of character or numeric data from the PC. Included in them are also routines that check for correct transmission or reception of data. *Part_cal.pc* controls the movement of the robot

during the process of collecting the calibration data; whereas *part_loc.pc* is used during the actual process of localizing the misaligned fender.

3.5 Experimental Set-Up

One of the main goals of this thesis is to investigate the feasibility of implementing the part localization algorithm on a commercial robot. The Fanuc S-110 and the developed APLS were used for the testing and verification of the algorithm. Recall that the algorithm was presented thoroughly in Chapter 2.

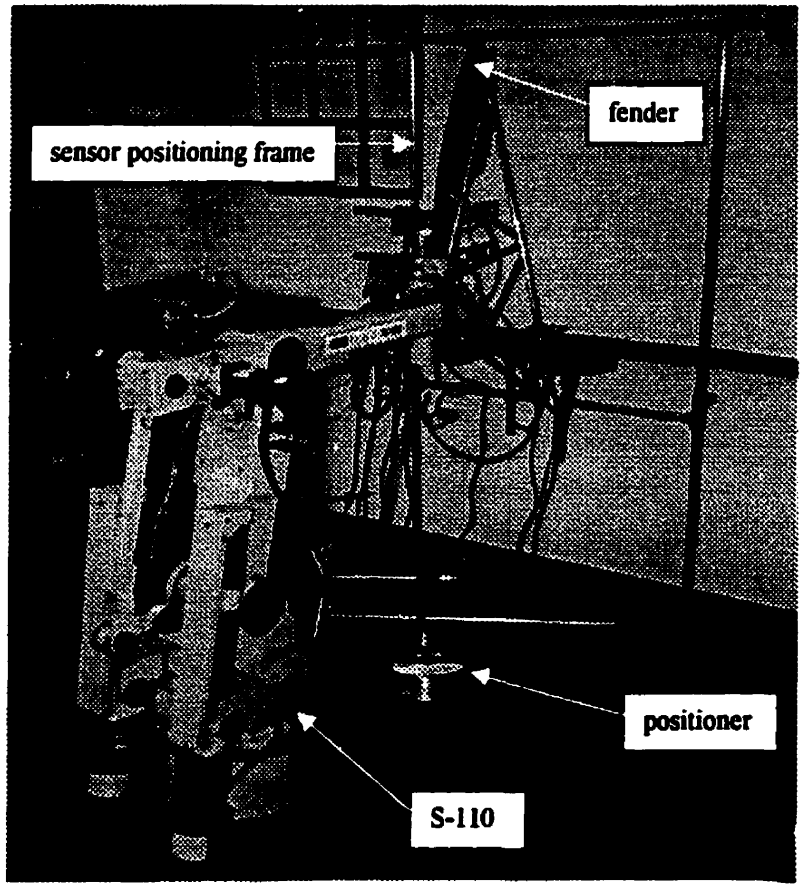

Figure 3-20: View of Typical Experiment and Setup

Figure 3-20 shows a picture of a typical experimental setup. For the purpose of performing localization experiments, a positioner was designed and built in the LNSC. The function of the positioner in Figure 3-20 is to hold the fender firmly in place while the robot grasps it. As pointed out in Section 1.2, the employment of this low precision positioner is necessary to ensure that the fender is placed within some close neighborhood of a planned location. This planned location is where the fender should be positioned for a successful assembly.

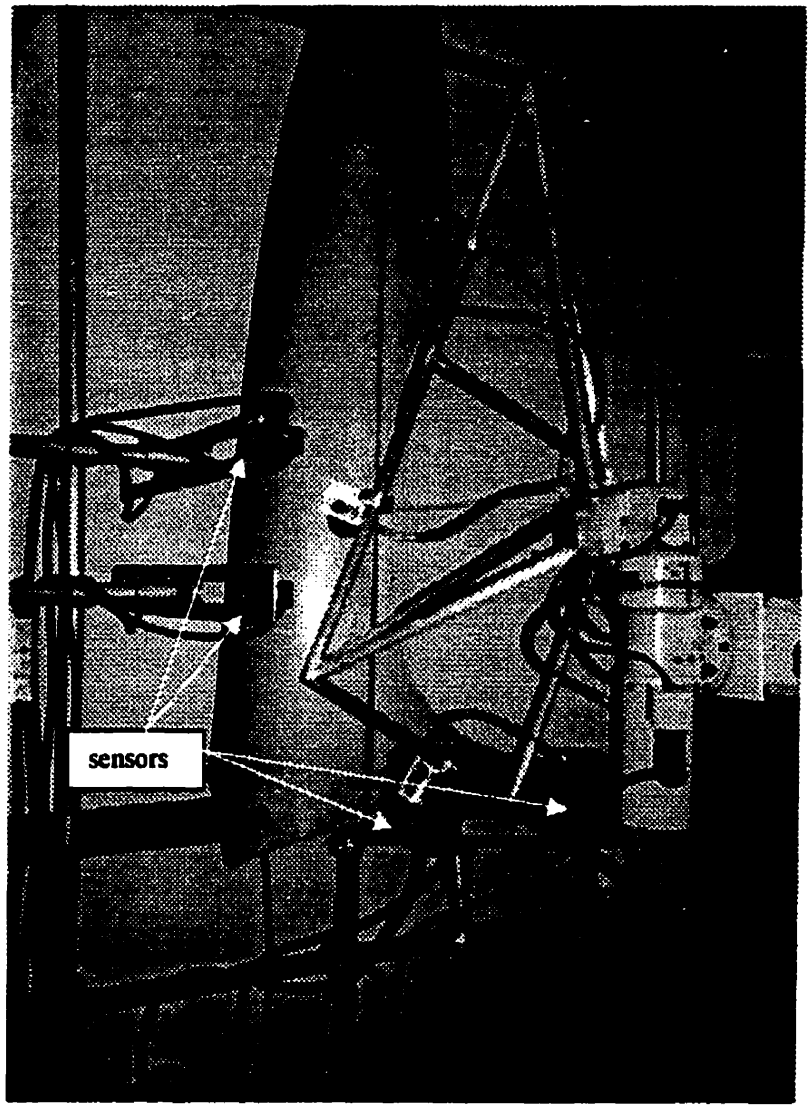

Figure 3-21: View of Sensors Taking Measurements of Fender

Figure 3-21 provides a close up view of the sensors taking measurements of the misaligned fender, grasped by the robot. To simplify the localization experiment, the robot's user-defined coordinate frame was set equal to the part (fender) coordinate frame, which was previously illustrated in Figure 2-1. The advantage of working in the same coordinate frames is that the perturbations or misalignments of the fender can be created artificially for the experiments. The results of these experiments are presented in Chapter 4. Lastly, Figure 3-22 shows a typical Windows based user-interface for the localization experiments. All the components of the user-interface were explained in previous sections.

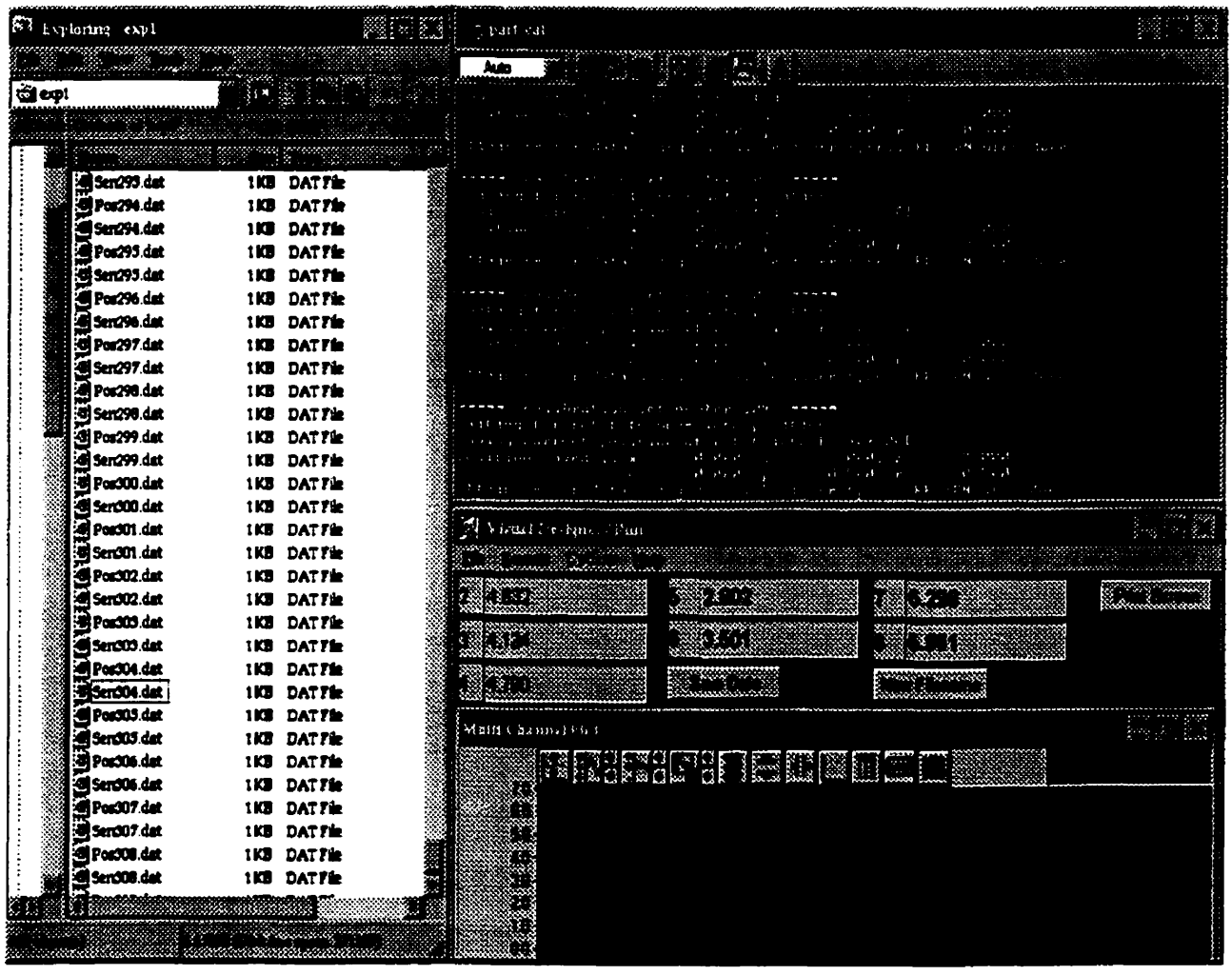

Figure 3-22: User-Interface for Part Localization Experiments

3.6 Summary

In this chapter, the development of Automatic Part Localization System (APLS) was described. APLS is a demonstration system that enables the use of the proposed part localization concept. This chapter gives a detailed explanation of various hardware and software components that form the system. Additional setup required to perform part localization experiments are also given. It is hoped that the documents presented in here will be sufficient to guide others in future investigations.

Chapter 4

Experimental Results

4.1 Introduction

In this chapter, experimental results are presented using the part localization algorithm developed in Chapter 2. The description of Automatic Part Localization System (APLS) and the experimental setup were presented in the previous chapter. The main goal of these experiments is to investigate the feasibility of implementing the algorithm on a commercial robot. The performances of using inverse and forward mappings in localizing a fender are also studied. See Chapter 2 for explanations of the inverse and forward mappings. In this chapter, the localization of the fender using the inverse mapping is called inverse localization, whereas the other is called forward localization if the forward mapping is used. Recall from Chapter 1 that the target performance or the acceptable estimated location error is within $\pm 0.5mm$ for the x, y, and z coordinates, and $\pm 0.07°$ for the w, p, and r angular coordinates. The error is the difference between the commanded and estimated location of the fender. Following the development of the algorithm in Chapter 2, three types of experiments are performed:

1. 3 DOF (x, y, r) edge localization experiments
2. 3 DOF (z, w, p) surface localization experiments

3. 6 DOF localization experiments

For each type, the results of the inverse localization are presented first. Secondly, a comparison is made to the results of the forward localization. Note that the sensor weights in the forward localization can be assigned to many different values. In this thesis, however, only a selected combination of the sensor weights is presented for each type of forward localization experiment. Detailed explanations are made on what each of these results represents. Lastly, concluding remarks are given for the overall experimental results, including noise considerations.

4.2 Edge Localization Performance Analysis

The experimental results for 3 DOF edge localization are presented for a single calibration data set. In the edge localization experiments, the misalignments are given to the x, y, and r coordinates. In order to measure the misalignments in these coordinate axes, edge sensors ($s4$, $s5$, $s6$, $s7$) are used. Figure 2-4, in Chapter 2, displays a diagram of the coordinate axes and the sensor vectors with respect to the fender. Three points at $30mm$ intervals are used for the x and y coordinates, and five points at 0.5° intervals are used for the r coordinate to form the calibration set. By taking all combinations of these, a 45 calibration-point data set is created, which evenly span the fender's possible misalignment range. Therefore, the given misalignment range of the fender are within $\pm 30mm$ for the x and y coordinates, and $\pm 0.5°$ for the r coordinate at the localization stage in Step 3 (see Chapter 2). The performance evaluation data and plots in the following sections correspond to results from 50 experiments in localizing the misaligned fender – using the edge localization.

4.2.1 Inverse Localization Performance

The edge localization performance using the inverse mapping is given in Table 4-1. It shows the error statistics in estimating the correct location of the fender. The data clearly shows that the performance is excellent. The mean values for all three errors are well within the target values of $\pm 0.5mm$ and $\pm 0.07°$.

	Inverse Localization Error		
	x (mm)	*y (mm)*	*r (°)*
Mean	-0.10	0.06	-0.010
STD	0.11	0.06	0.007
Max	0.13	0.19	0.003
Min	-0.27	-0.06	-0.026

Table 4-1: Inverse Localization Error Statistics for 3 DOF Edge Localization

Figure 4-1 shows a plot of the x coordinate error as a function of the misalignment along the x coordinate. Each point (o) on the plot is a single experiment performed to find the fender's misalignments. Notice the different values of the x coordinate error at the zero x-misalignment location. The differences, which are as much as $0.25mm$ apart, are caused by different magnitudes of misalignments in the y or w coordinate directions. It implies that misalignments in one coordinate direction have an effect on the magnitude of the error on the other. Figure 4-2 is the plot of the y coordinate error versus the misalignment along the y coordinate and Figure 4-3 is for the r coordinate error versus the misalignment along the r coordinate. The three plots show that all 50 experiments have localized the fender within the target range.

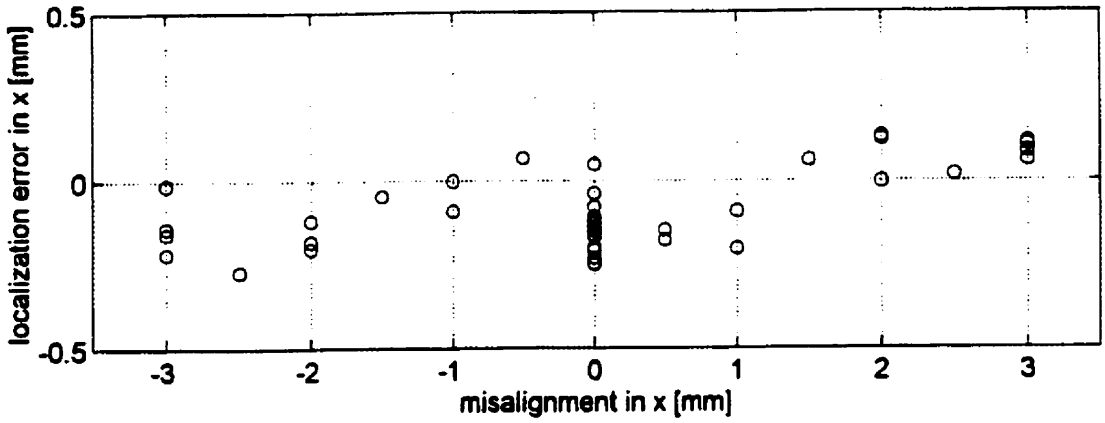

Figure 4-1: 3 DOF Edge Localization (Inverse) – x

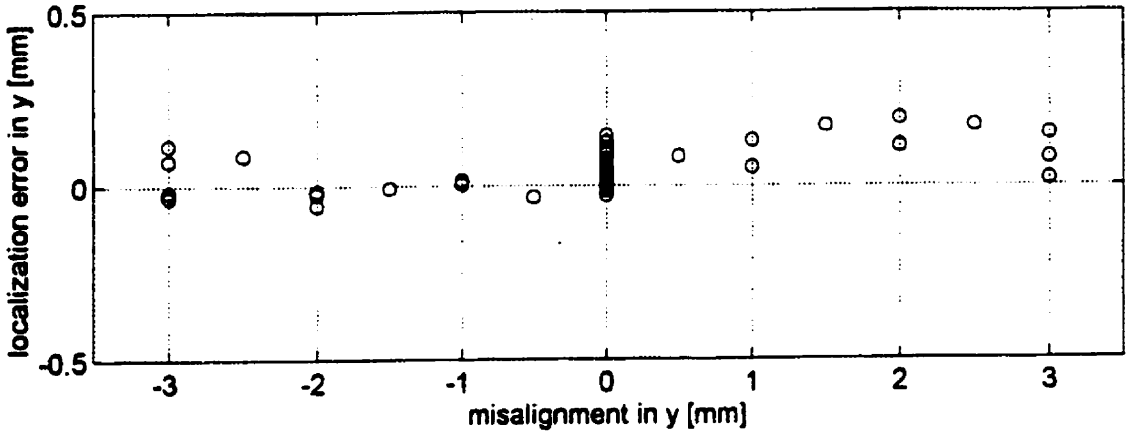

Figure 4-2: 3 DOF Edge Localization (Inverse) – y

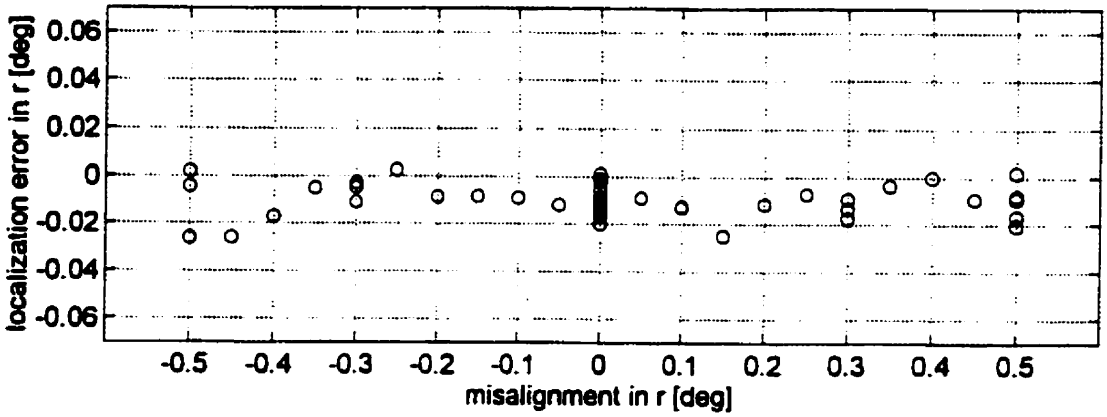

Figure 4-3: 3 DOF Edge Localization (Inverse) – r

4.2.2 Forward Localization Performance

Table 4-2 displays error statistics for the forward edge localization of the misaligned fender. Recall from Chapter 2 that the forward localization is an iterative procedure – the calculation of the fender's location is repeated for a preset number of iterations. The results presented here were obtained using 500 iterations and the following scalar sensitivity weights (λ_i) for the four edge sensors:

$$\lambda_4 = 10 \text{ for Sensor 4 } (s4)$$

$$\lambda_5 = 10 \text{ for Sensor 5 } (s5)$$

$$\lambda_6 = 1 \text{ for Sensor 6 } (s6)$$

$$\lambda_7 = 10 \text{ for Sensor 7 } (s7)$$

The details on the usage of these weights are explained in Section 2.3, as Equations (2.35) and (2.37). The table indicates that the performance of the forward localization algorithm is almost identical to that of the inverse localization algorithm. Only the mean value for the x coordinate error is slightly improved. Other values for the sensitivity weights or the number of iterations generally produced results for these experiments which are worse. The plots, in Figures 4-4, 4-5, and 4-6, for the forward localization are very similar to the corresponding plots for the inverse. Again, each point (o) on the plots is one of the 50 experiments performed to find the fender's misalignments, using the edge localization.

	Forward Localization Error		
	x (mm)	*y (mm)*	*R (°)*
Mean	-0.09	0.06	-0.010
STD	0.11	0.06	0.007
Max	0.15	0.18	0.008
Min	-0.26	-0.06	-0.026

Table 4-2: Forward Localization Error Statistics for 3 DOF Edge Localization

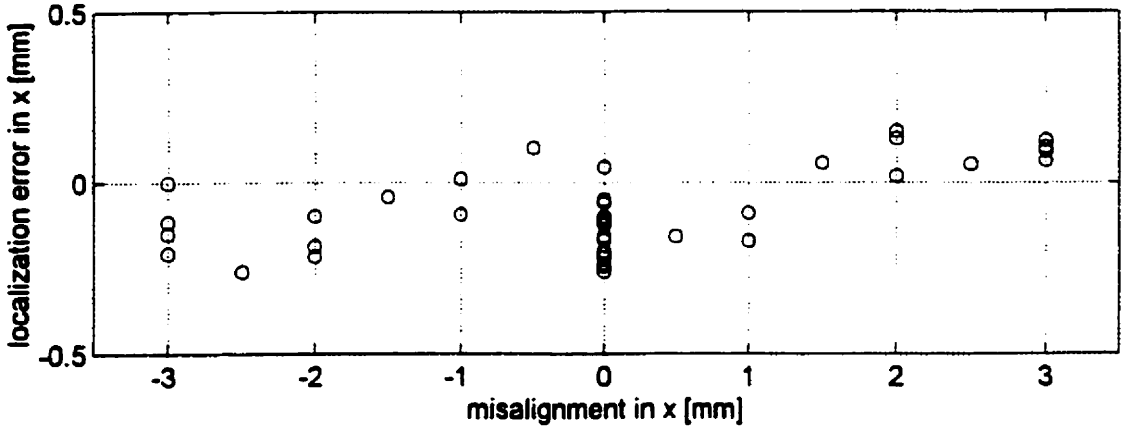

Figure 4-4: 3 DOF Edge Localization (Forward) – x

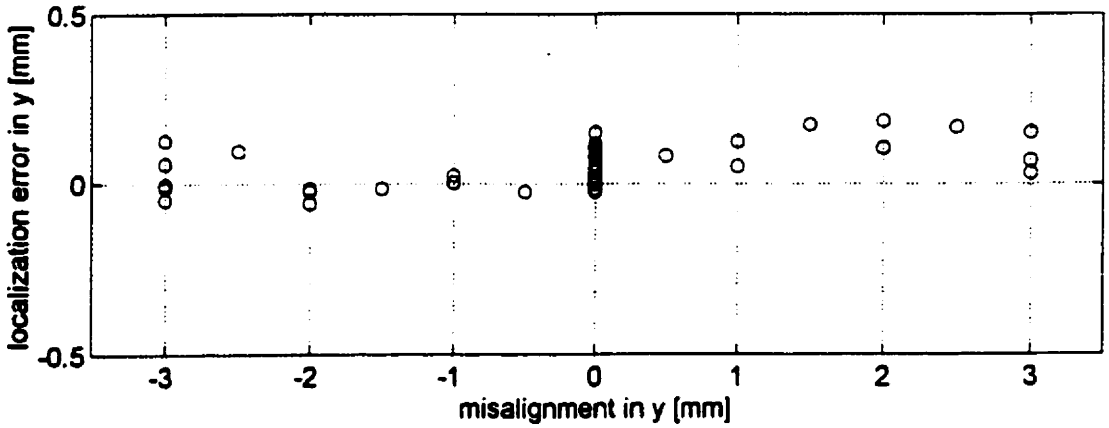

Figure 4-5: 3 DOF Edge Localization (Forward) – y

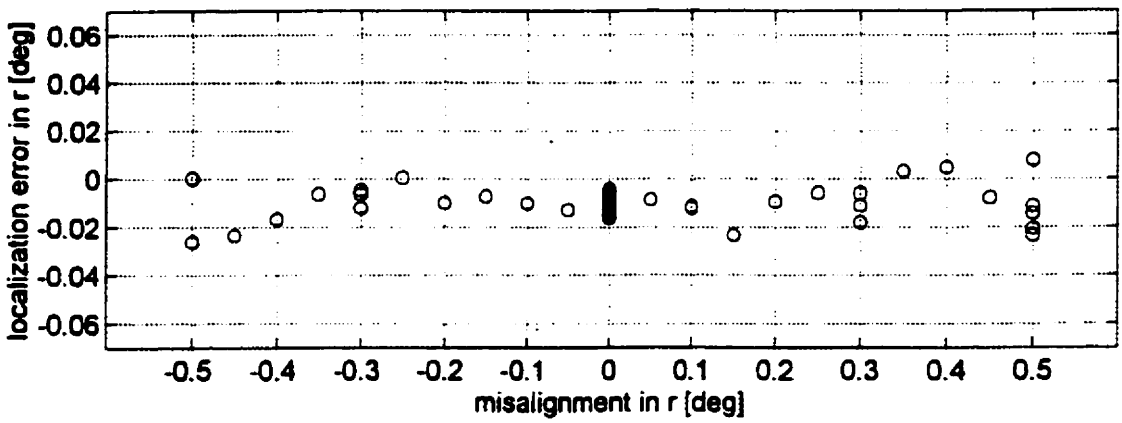

Figure 4-6: 3 DOF Edge Localization (Forward) – r

Figures 4-4, 4-5, and 4-6 do not clearly show the correlation between the error in one coordinate and misalignments along other coordinates. To avoid repetition, the correlation between the misalignments in different coordinates is presented using 6 DOF localization experiments (Section 4.5).

4.3 Surface Localization Performance Analysis

The experimental results for 3 DOF surface localization are presented for a single calibration data set. Recall from Chapter 2 that the misalignments are present in z, w, and p coordinate axes in the surface localization experiments. In order to measure the misalignments about the z, w, and p coordinates, proximity sensors ($s1$, $s2$, $s3$) are used. Figure 2-3 shows the representation of these coordinate axes and sensor vectors with respect to the fender in a schematic diagram. Three points at $30mm$ intervals are used for the z coordinate, and five points at $0.5°$ intervals are used for the w and p coordinates to form the calibration set. By taking all combinations of these perturbations, the set contains 75 calibration points that evenly span the fender's possible misalignment range. The misalignment range of the fender is within $\pm 30mm$ for the z coordinate, and $\pm 0.5°$ for the w and p coordinates at the localization stage. The following performance evaluation data and plots correspond to 45 experiments in which the fender is localized – using the surface localization.

4.3.1 Inverse Localization Performance

The results of the surface localization performance using the inverse mapping is displayed in Table 4-3. The table shows the error statistics in estimating the correct location of the fender. Notice that all three errors are well within the target values. However, the maximum and minimum error values are much greater than that of the edge localization.

	Inverse Localization Error		
	z (mm)	*w (°)*	*p (°)*
Mean	-0.04	-0.009	-0.009
STD	0.17	0.025	0.023
Max	0.22	0.045	0.023
Min	-0.47	-0.066	-0.062

Table 4-3: Inverse Localization Error Statistics for 3 DOF Surface Localization

Figure 4-7 shows the plot of the z coordinate error as a function of the misalignment along the z coordinate. As in the inverse localization, each point (o) on the plot is one of the 45 experiments in finding the fender's location. Figure 4-8 is the plot for the w coordinate error versus the misalignment along the w coordinate and Figure 4-9 is for the p error versus the misalignment along the p coordinate. The three plots show that all 45 experiments have localized the fender within the target range.

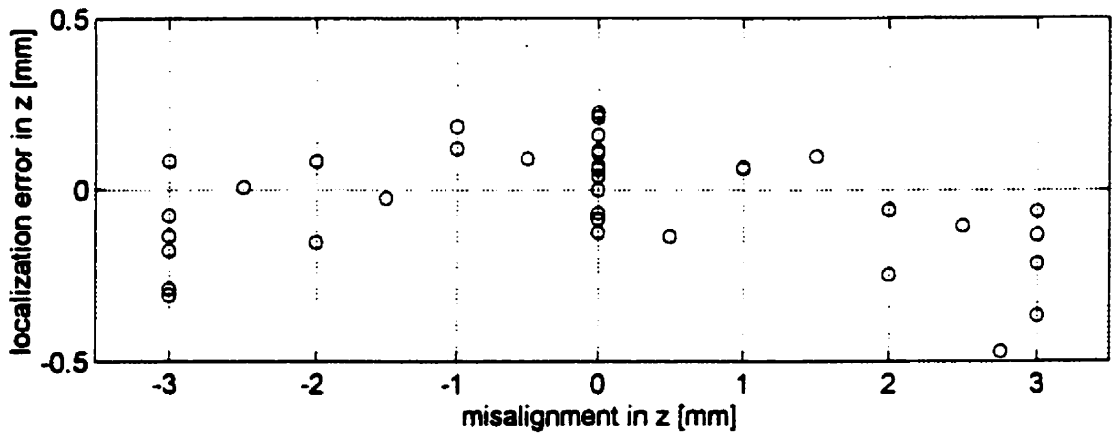

Figure 4-7: 3 DOF Surface Localization (Inverse) – z

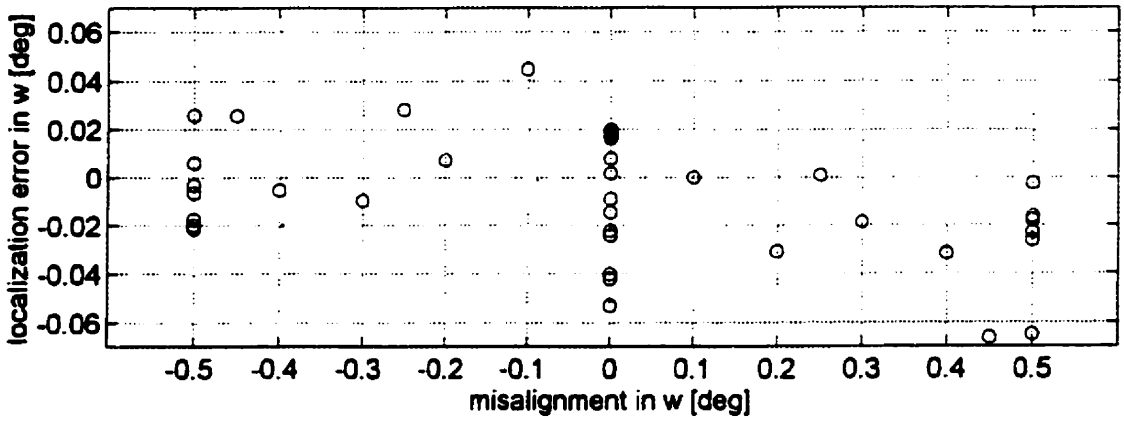

Figure 4-8: 3 DOF Surface Localization (Inverse) – *w*

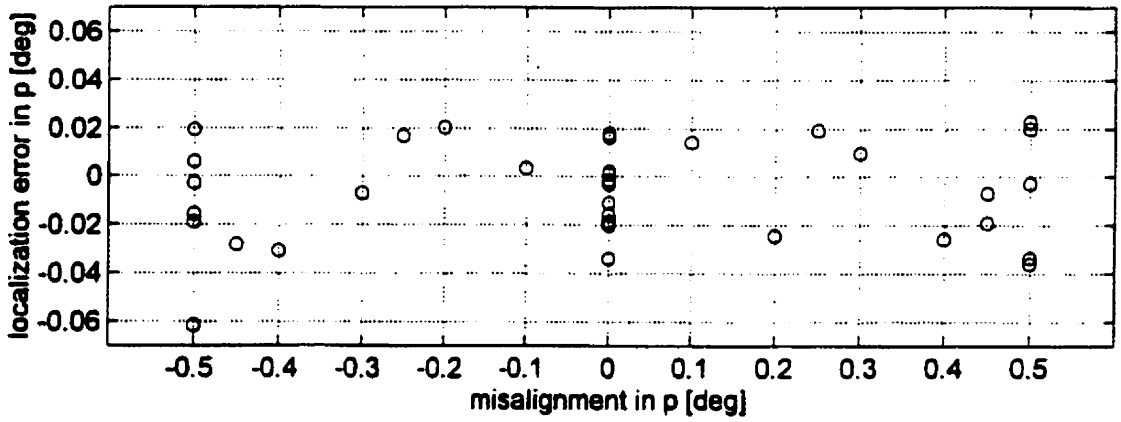

Figure 4-9: 3 DOF Surface Localization (Inverse) – *p*

Comparing to the plots previously presented in the edge localization, the points (o) are more scattered along the error axes in above plots. Even though all points are within the target values, one or two points in each plot are close to being out of the bounds. The main reason for this is that the surface localization uses fewer number of sensors, i.e. three, than the edge localization, which uses four, to localize the same number of degrees of freedom (3 DOF). The reduced number of measurements has led to greater inaccuracy in estimating the fender's location. In addition, the complexity in the shape of the fender has also contributed to the inaccuracy. The greater complexity in the

shape of the fender's surface than its edges has led to difficulty in mapping the surface more effectively.

4.3.2 Forward Localization Performance

Table 4-4 gives the error statistics for the forward surface localization of the misaligned fender. Recall that in the forward localization, the calculation of the fender's location is repeated for a preset number of iterations. The experimental results are obtained using 500 iterations and the following scalar sensitivity weights (λ_i) for the three proximity sensors:

$$\lambda_1 = 1 \text{ for Sensor 1 } (s1)$$

$$\lambda_2 = 10 \text{ for Sensor 2 } (s2)$$

$$\lambda_3 = 1 \text{ for Sensor 3 } (s3)$$

The details on the usage of the above sensitivity weights in the surface localization algorithm are previously explained in Section 2.2, as Equations (2.27) and (2.28). Table 4-4 indicates that the forward localization's performance is superior for the w and p coordinates than that of the inverse localization. However, the mean value for the z coordinate error has increased significantly, even though its standard deviation and maximum and minimum values have been improved. These results indicate that there are coupled trade-offs in the performance between the coordinates. The performances were improved in the w and p coordinates, whereas the performance in the z coordinate has suffered. Other values for the sensitivity weights or the number of iterations have produced different results. For example, when the sensitive weights (λ_1, λ_2, λ_3) were set to 1 with 100 iterations, the mean errors were 0.050*mm*, 0.013 °, and 0.003 ° for the z, w, and p coordinates respectively. These results also show the trade-off between the coordinates: the performance of the p coordinate improved compared to that of the inverse, whereas the w coordinate's performance fell. However, finding the optimum values for the sensitivity weights or the number of iterations are not in the scope of this thesis; hence, they are not described here in detail.

	Forward Localization Error		
	z (mm)	*w (°)*	*p (°)*
Mean	-0.09	-0.007	-0.007
STD	0.12	0.025	0.021
Max	0.08	0.044	0.031
Min	-0.46	-0.070	-0.048

Table 4-4: Forward Localization Error Statistics for 3 DOF Surface Localization

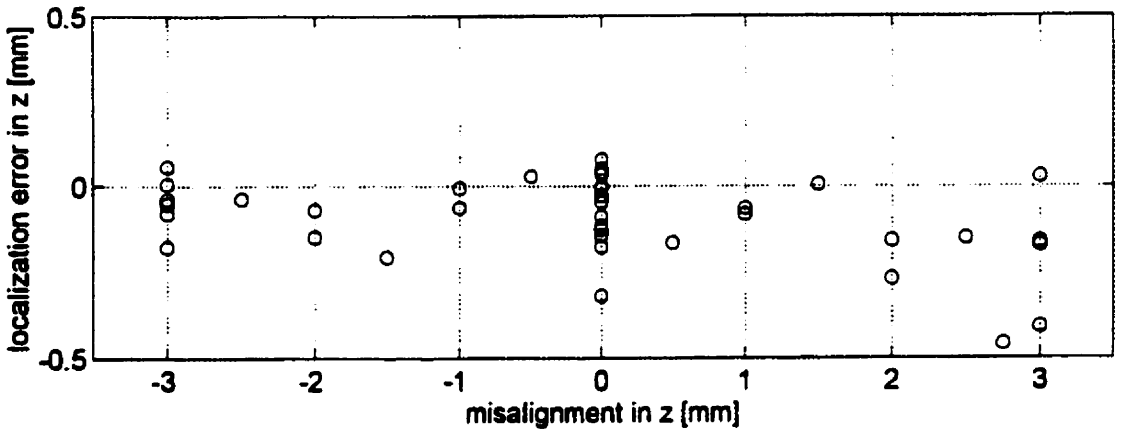

Figure 4-10: 3 DOF Surface Localization (Forward) – *z*

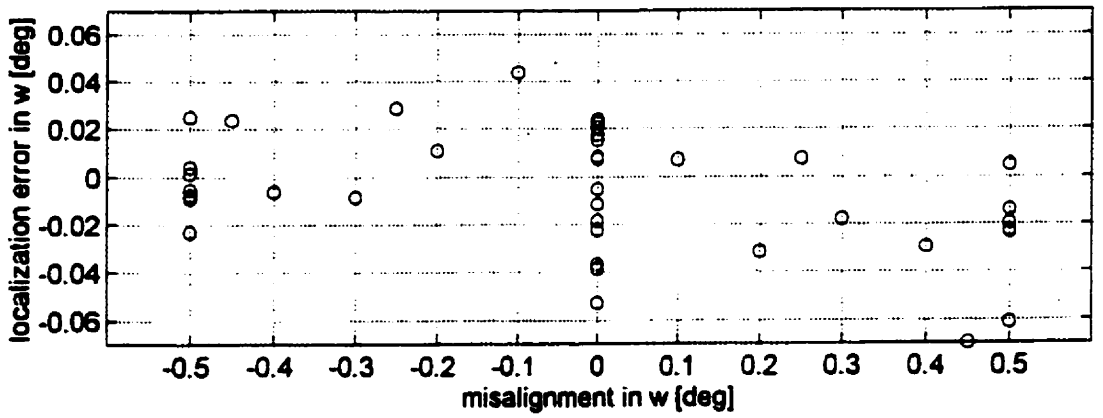

Figure 4-11: 3 DOF Surface Localization (Forward) – *w*

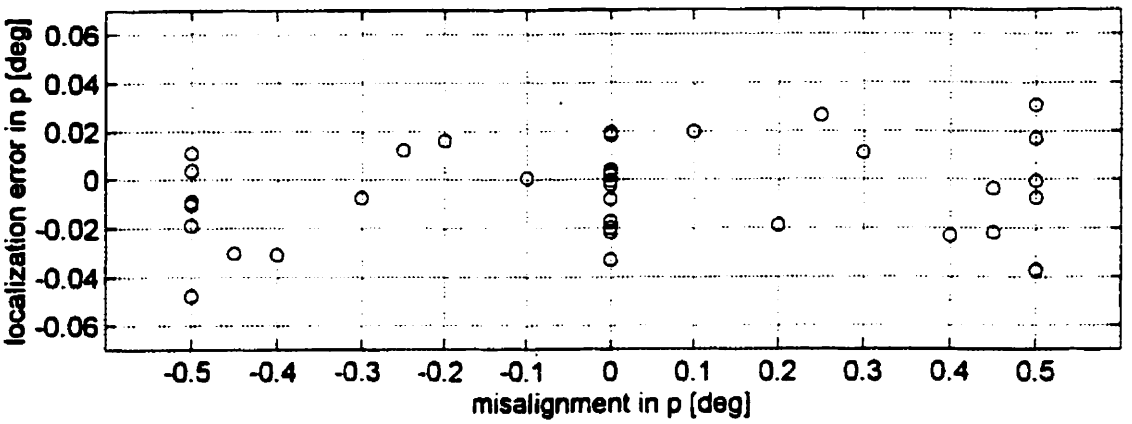

Figure 4-12: 3 DOF Surface Localization (Forward) – p

The plots (Figures 4-10, 4-11, 4-12) for the forward localization are very similar to the corresponding plots for the inverse, as for the case of the edge localization. Take note that the points on Figures 4-10 have shifted downwards slightly compared to the points on Figure 4-7 of the inverse. This shifting has led to the increase of the z coordinate's mean error from -0.04*mm* to -0.09*mm*.

4.4 6 DOF Localization Performance Analysis

The experimental results for 6 DOF localization are presented for a single calibration data set that contains a total of 729 calibration points. In this experiment, misalignments are given to all coordinate axes: three translations (x, y, z) and three rotations (w, p, r). In order to measure the misalignments in these coordinates, both the proximity sensors ($s1$, $s2$, $s3$) and edge sensors ($s4$, $s5$, $s6$, $s7$) are used. Figure 2-1, in Chapter 2, shows the representation of the coordinate axes and the sensor vectors with respect to the fender for this experiment. Three points at 20*mm* intervals are used for the x, y, and z coordinates, and three points at 0.5° intervals were used for the w, p, and r to form the calibration set. By taking all combinations of these perturbations, the 729 point calibration set is formed, which evenly span the fender's possible misalignment

range. The misalignment range is $\pm 20mm$ for the x, y, and z coordinates, and $\pm 0.5°$ for the w, p, and r coordinates in the 6 DOF localization. The following performance evaluation data and plots correspond to results from 129 experiments conducted in localizing the misaligned fender

4.4.1 Inverse Localization Performance

The results of using 6 DOF inverse localization algorithm are shown in Table 4-5. The table shows the error statistics in estimating the localization of the misaligned fender. As can be seen from this data, the performance is good. The mean values for the x, y, and z coordinate errors are well within the target value of $\pm 0.5mm$. The mean values for the w, p, and r coordinate errors are also within the target value of $\pm 0.07°$. The maximum and minimum values also do not exceed the target values.

	Inverse Localization Error					
	x (mm)	*y (mm)*	*z (mm)*	*w (°)*	*p (°)*	*r (°)*
Mean	0.19	-0.01	-0.24	-0.038	-0.031	0.020
STD	0.14	0.11	0.10	0.020	0.023	0.015
Max	0.49	0.39	0.12	0.032	0.059	0.060
Min	-0.25	-0.25	-0.50	-0.069	-0.069	-0.047

Table 4-5: Inverse Localization Error Statistics for 6 DOF Localization

Figure 4-13 presents the x coordinate error as a function of the misalignment along the x coordinate, when no misalignments are present to other coordinate axes. Similarly, Figures 4-14 to 4-18 illustrate the coordinate errors from y to r coordinates respectively. From these plots, it is evident that each coordinates error is a strong function of its misalignment. For example, in Figure 4-14, the y coordinate error is linearly

dependent on the y coordinate misalignment. The y coordinate error increases as the y coordinate misalignment changes from $-0.2mm$ to $+0.2mm$. It is also possible to describe the relationship between the localization error and the misalignment for every coordinate in detail as [26] has done. In the 6 DOF localization, however, finding the relationships becomes insignificant. Again considering the y coordinate, when misalignments are also present in the other coordinates, each has an effect to the y localization error. Hence the relationship between the y coordinate error and the y misalignment will no longer be linear.

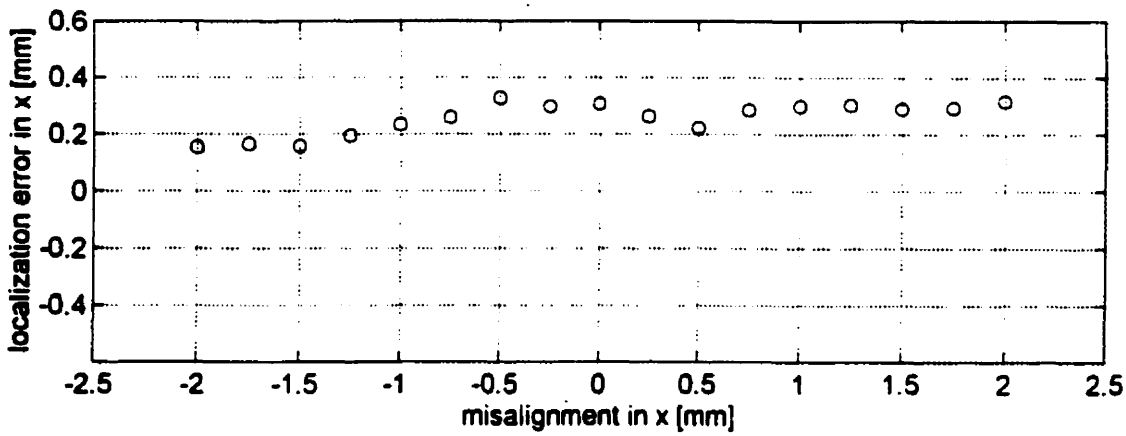

Figure 4-13: 6 DOF Inverse Localization – x, no misalignment in other coordinates

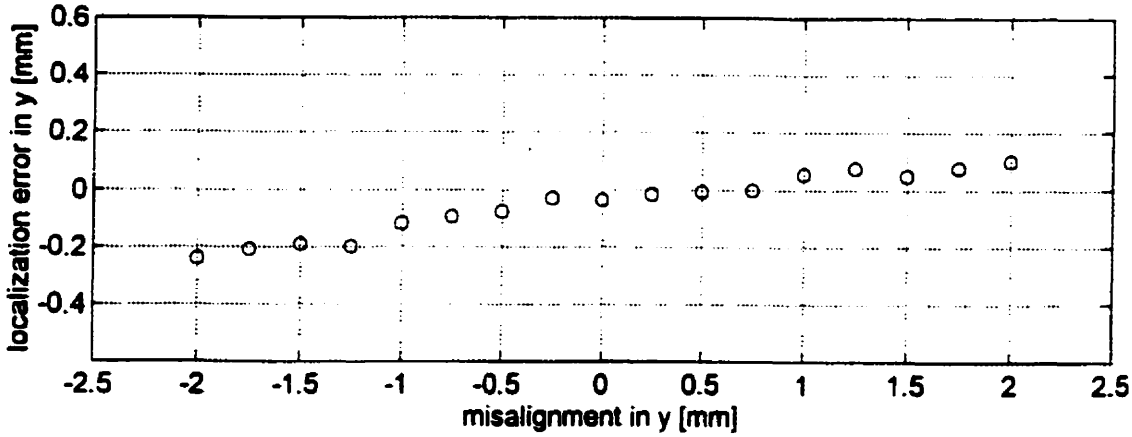

Figure 4-14: 6 DOF Inverse Localization – y, no misalignment in other coordinates

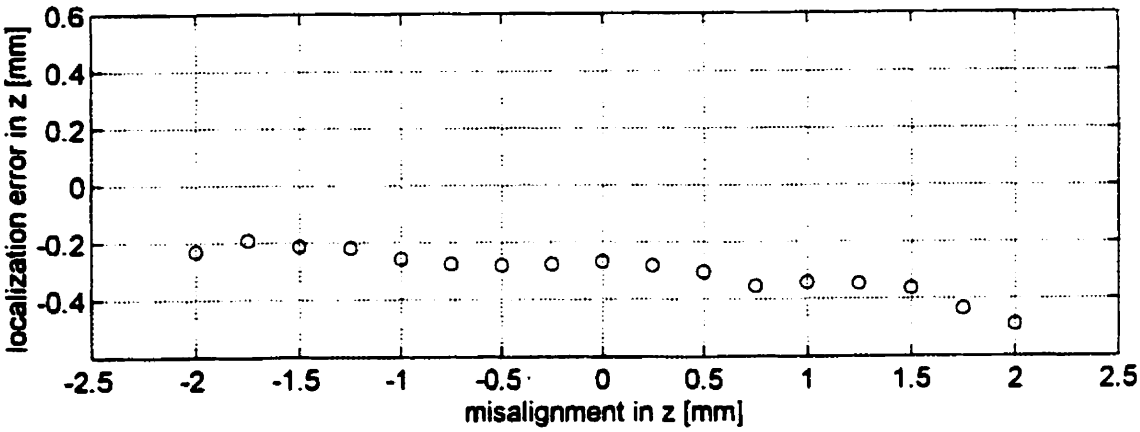

Figure 4-15: 6 DOF Inverse Localization – z, no misalignment in other coordinates

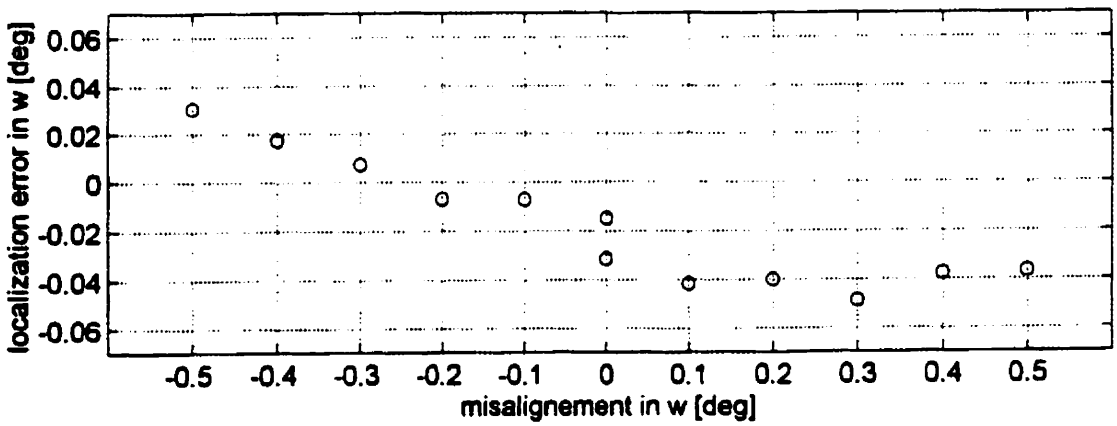

Figure 4-16: 6 DOF Inverse Localization – w, no misalignment in other coordinates

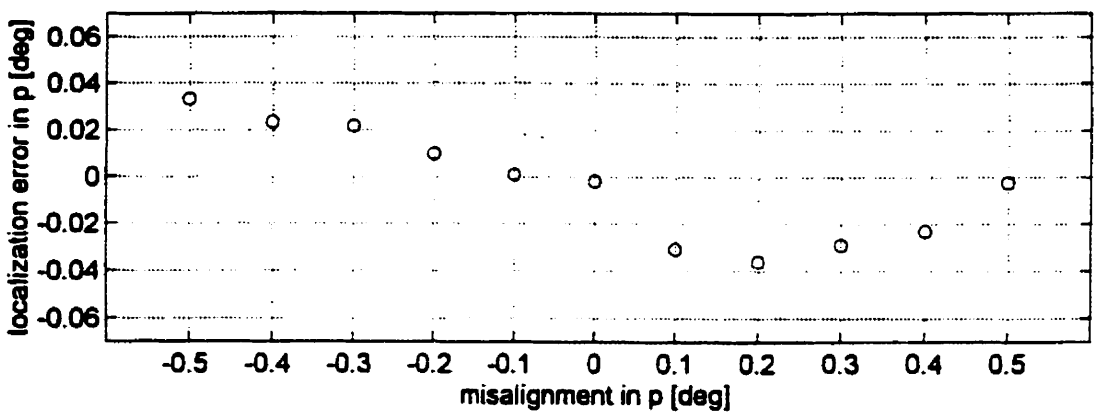

Figure 4-17: 6 DOF Inverse Localization – p, no misalignment in other coordinates

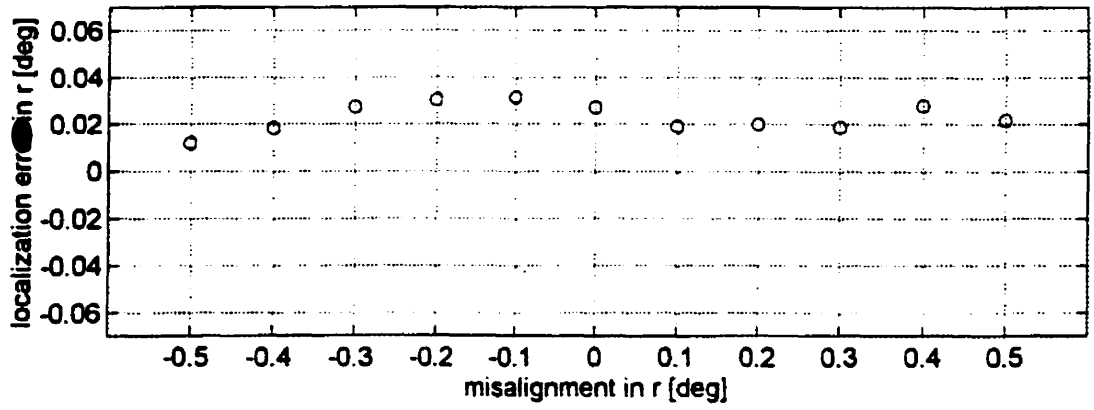

Figure 4-18: 6 DOF Inverse Localization – *r*, no misalignment in other coordinates

Figures 4-19, 4-20, and 4-21 also show the localization errors in the *x*, *y*, and *z* coordinates respectively. However, these plots are obtained by allowing misalignments in all translational coordinate axes and maintaining zero misalignment in the rotational coordinate axes. Conversely, the errors for the *w*, *p*, and *r* coordinates in the next three plots (Figures 4-22, 4-23, 4-24) are plotted with misalignment present for all rotational coordinates and no misalignment in the translational coordinates. As previously mentioned, the effect of misalignments from other coordinates starts to show in Figures 4-19 to 4-24. Even though only three coordinates are involved in each plot, the location of the points (o) in these plots has become less predictable than in the previous six plots.

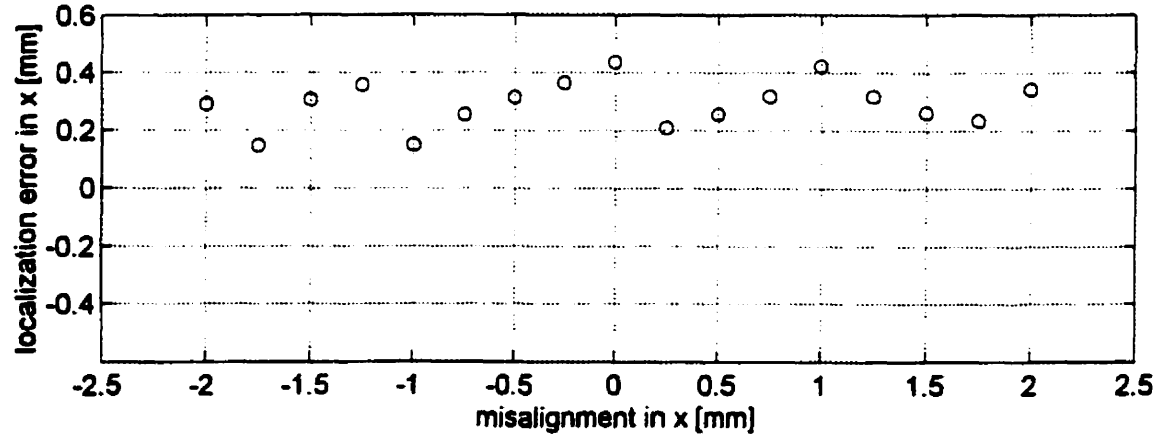

Figure 4-19: 6 DOF Inverse Localization–*x*, misalignment in all translational coordinates

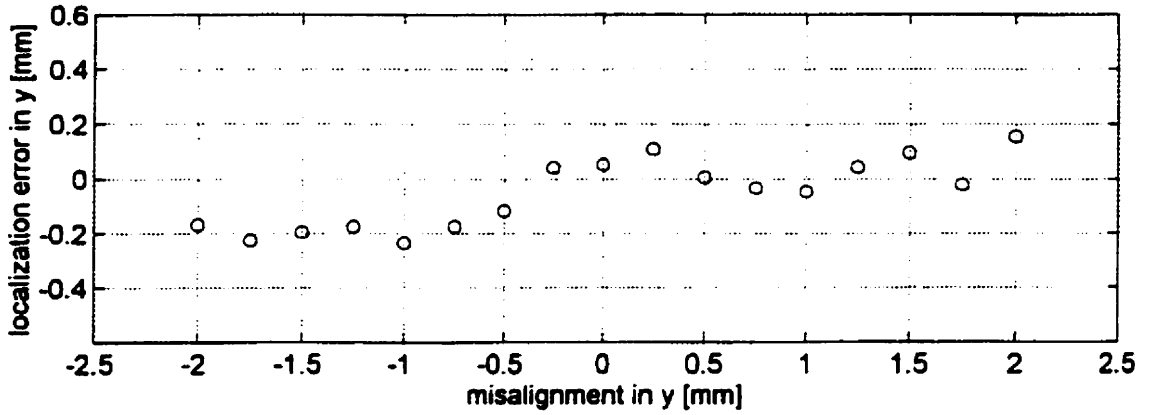

Figure 4-20: 6 DOF Inverse Localization–y, misalignment in all translational coordinates

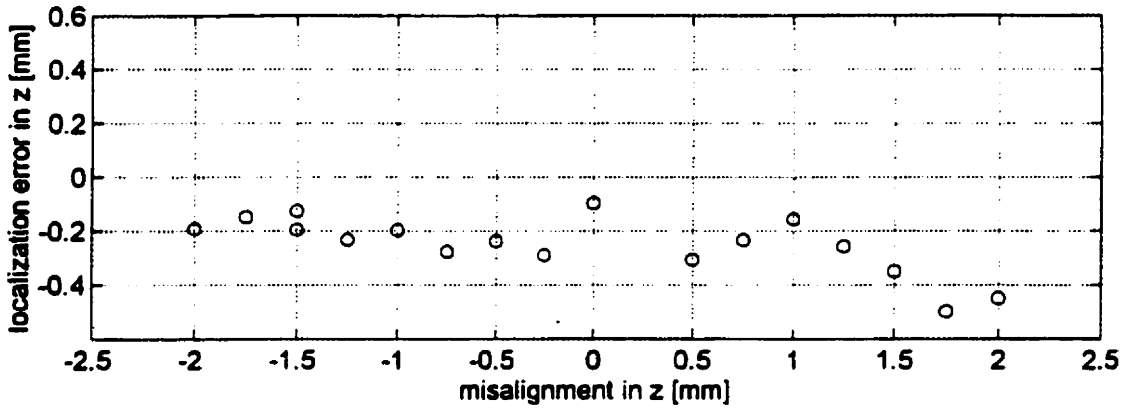

Figure 4-21: 6 DOF Inverse Localization–z, misalignment in all translational coordinates

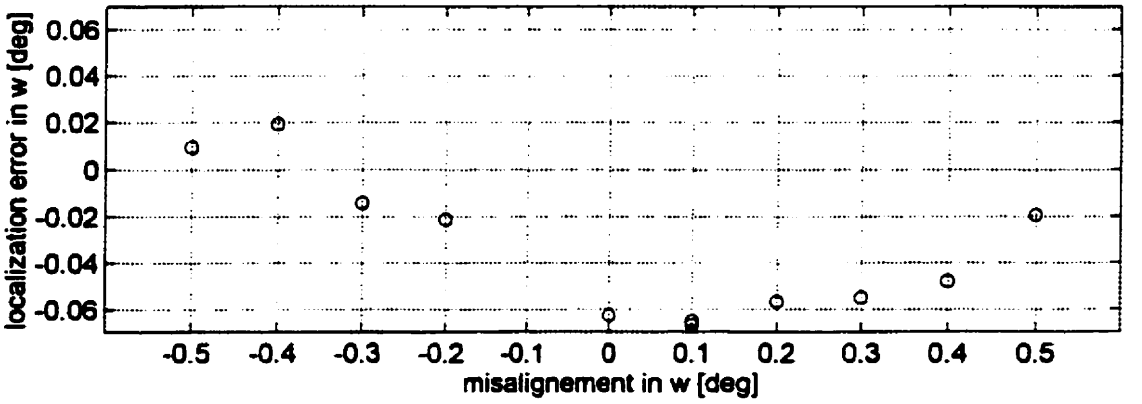

Figure 4-22: 6 DOF Inverse Localization – w, misalignment in all rotational coordinates

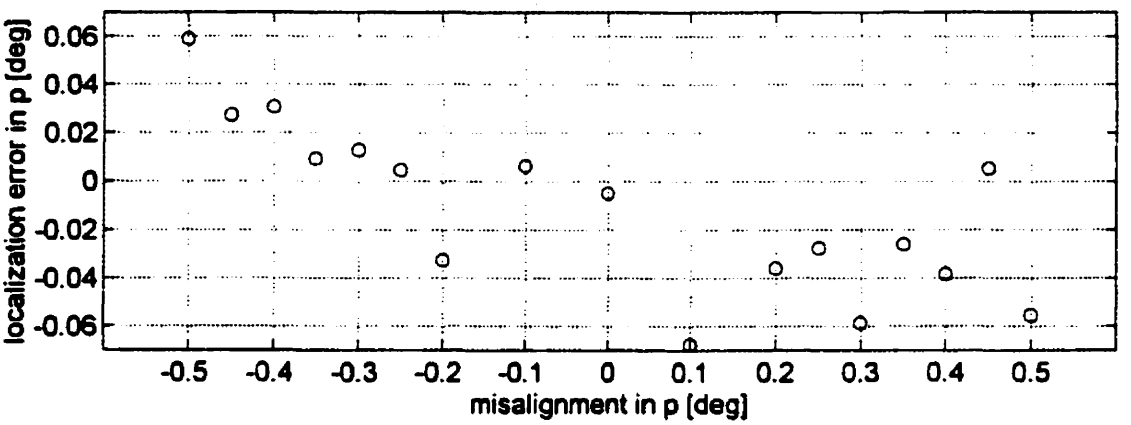

Figure 4-23: 6 DOF Inverse Localization – p, misalignment in all rotational coordinates

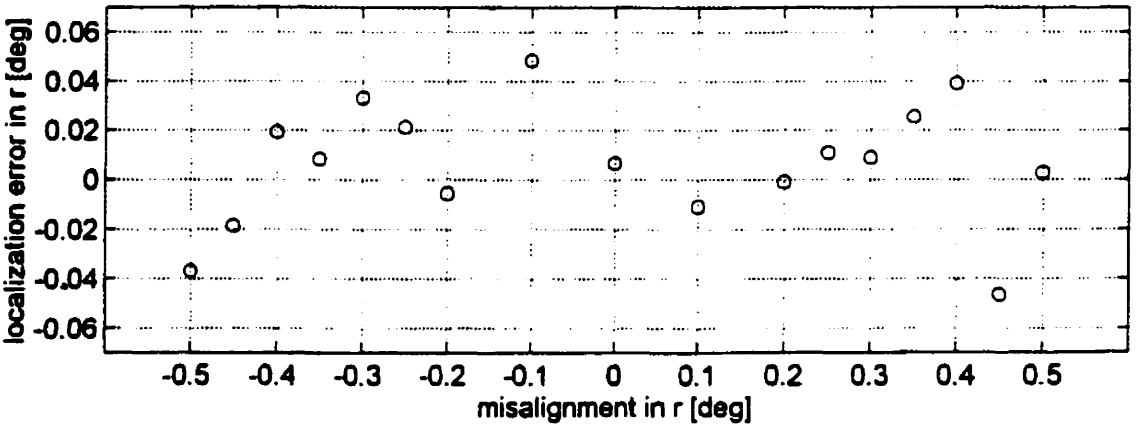

Figure 4-24: 6 DOF Inverse Localization – r, misalignment in all rotational coordinates

Figures 4-25 to 4-30 display the localization errors for the six coordinates with misalignment present in all coordinates simultaneously. Note that the points on the plots are widely scattered and generally show no trend at all. As expected, the values for the localization errors became more difficult to predict.

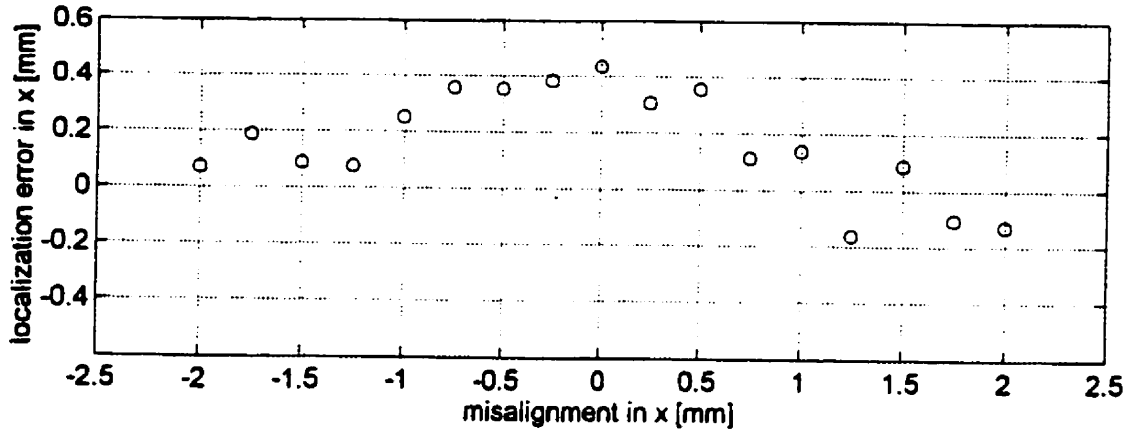

Figure 4-25: 6 DOF Inverse Localization – x, misalignment in all coordinates

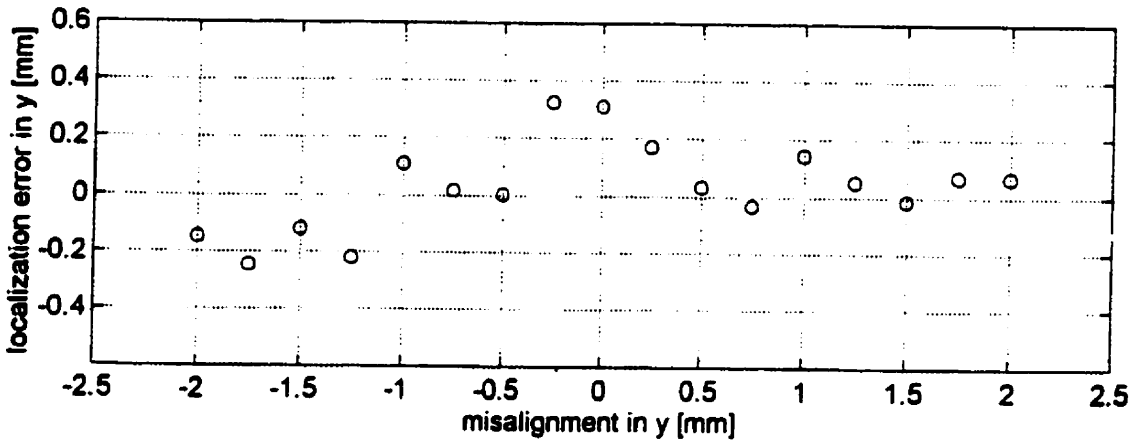

Figure 4-26: 6 DOF Inverse Localization – y, misalignment in all coordinates

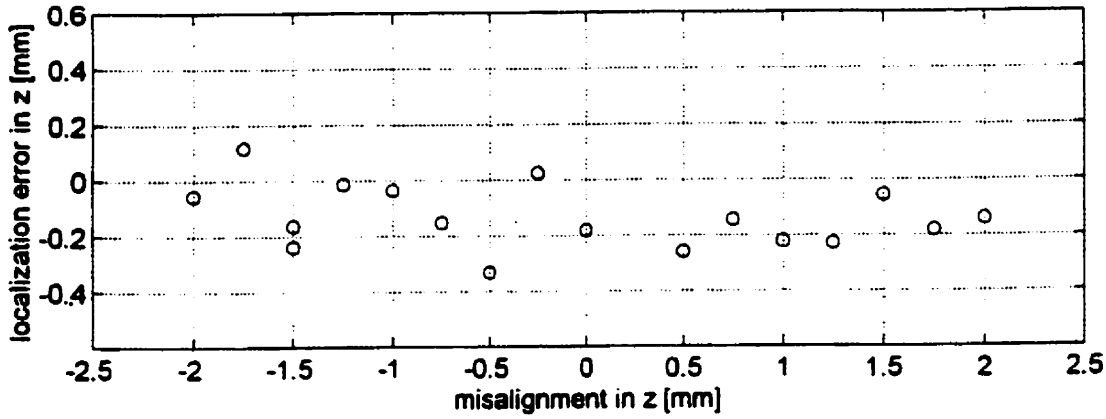

Figure 4-27: 6 DOF Inverse Localization – z, misalignment in all coordinates

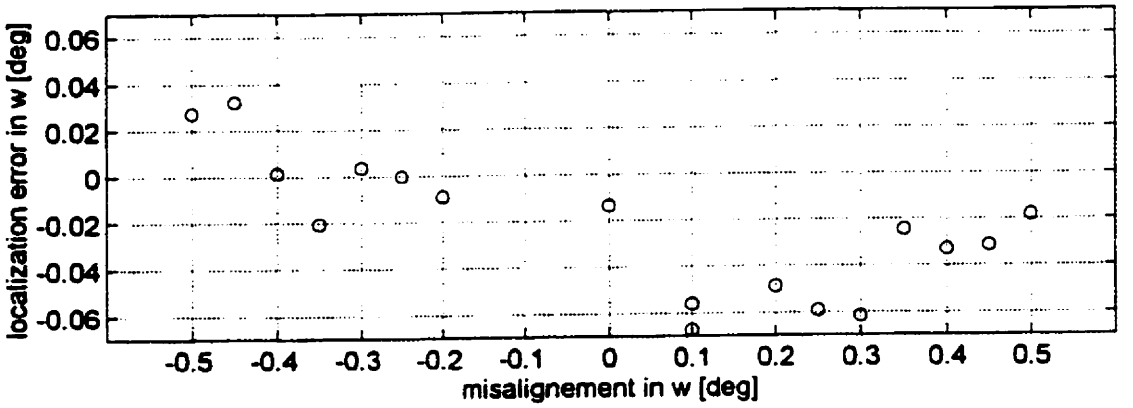

Figure 4-28: 6 DOF Inverse Localization – *w*, misalignment in all coordinates

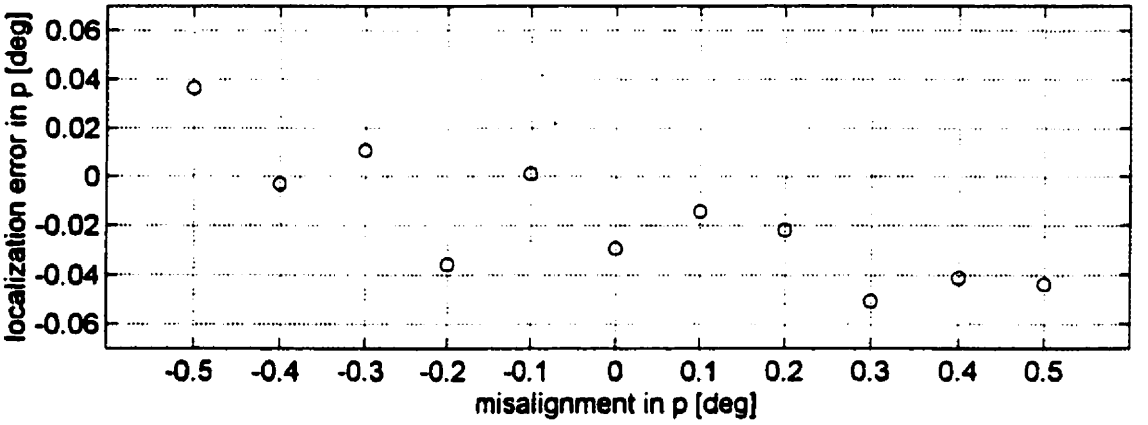

Figure 4-29: 6 DOF Inverse Localization – *p*, misalignment in all coordinates

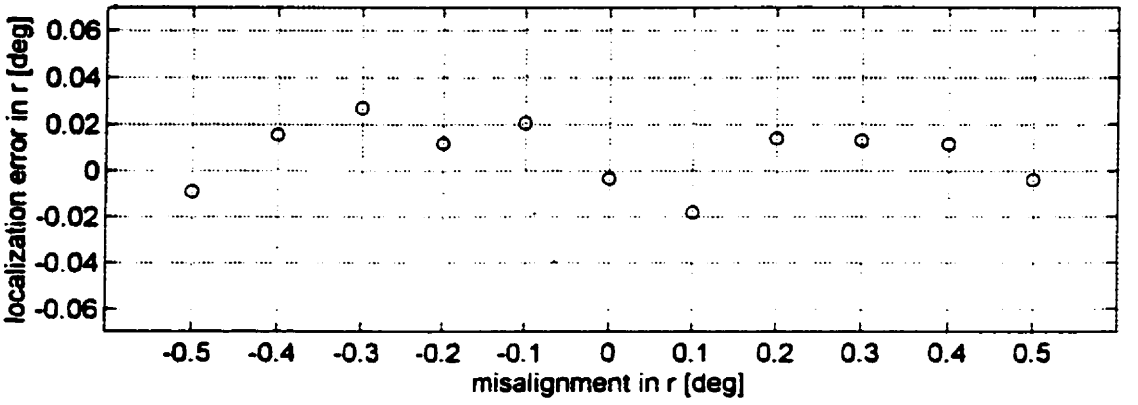

Figure 4-30: 6 DOF Inverse Localization – *r*, misalignment in all coordinates

Figures 4-13 to 4-30 clearly indicate that all 6 DOF-inverse experiments (o) have localized the fender within the target range. However, as Figures 4-25 to 4-30 show, when misalignments are present in other coordinates simultaneously the localization performance is not as good as that of the edge localization, but close to the surface localization. The reasons that contribute to the exceptional performance of the edge localization were explained in Section 4.3.1 of this chapter.

4.4.2 Forward Localization Performance

The 6 DOF localization performance using the forward mapping is shown in Table 4-6. The results presented here were obtained using 500 iterations and the following scalar sensitivity weights (λ_i) for the seven sensors:

$$\lambda_1 = 10 \text{ for Sensor 1 } (s1)$$

$$\lambda_2 = 10 \text{ for Sensor 2 } (s2)$$

$$\lambda_3 = 10 \text{ for Sensor 3 } (s3)$$

$$\lambda_4 = 1 \text{ for Sensor 4 } (s4)$$

$$\lambda_5 = 1 \text{ for Sensor 5 } (s5)$$

$$\lambda_6 = 1 \text{ for Sensor 6 } (s6)$$

$$\lambda_7 = 1 \text{ for Sensor 7 } (s7)$$

Recall from Chapter 2 that the forward calculation of the fender's misalignment is repeated for a preset number of iterations – which was chosen as 500. A detailed explanation on these sensitivity weights is also given Chapter 2; see Equations (2.45) and (2.46). Only the proximity sensors were assigned the value of 10 for their sensitivity weights (λ_1, λ_2, λ_3), the edge sensors were left unweighted. Since the proximity sensors are used to measure the surface-detection related coordinates (z, w, p), the performance would generally be better along these coordinates, than in the case of inverse localization.

Table 4-6 displays the comparable error statistics in localizing the misaligned fender with that of the inverse localization presented in the previous section. By comparing to the results in Table 4-5 of the inverse localization, the performance is seen

to be similar for the x and y coordinate errors. On the other hand, the z, w, p, and r coordinates show improvements over the results of the inverse localization. For example, the r coordinate error is about half that of the inverse result in Table 4-5, 0.020° to 0.009°. However, while the mean values are excellent, the standard deviations and the values for the maximum and minimum errors have increased. Some of the maximum and minimum error values are even outside the target range. As expected, the weighting of the proximity sensors has led to the decrease in the mean localization error for the z, w, and p coordinates.

	Forward Localization Error					
	X (mm)	*y (mm)*	*z (mm)*	*w (°)*	*p (°)*	*r (°)*
Mean	0.19	0.02	-0.16	-0.033	-0.027	0.009
STD	0.16	0.18	0.12	0.022	0.024	0.025
Max	0.62	0.52	0.25	0.035	0.059	0.089
Min	-0.41	-0.35	-0.43	-0.076	-0.071	-0.072

Table 4-6: Forward Localization Error Statistics for 6 DOF Localization

Figures 4-31 to 4-36 show the forward localization errors in each coordinate with misalignments present in all coordinates simultaneously. Each point (o) on the plot is a single experiment performed to find the fender's misalignments. A total of 17, out of the 129, 6 DOF localization experiments are shown in the plots. Only in these experiments were intentionally given misaligned in all coordinates.

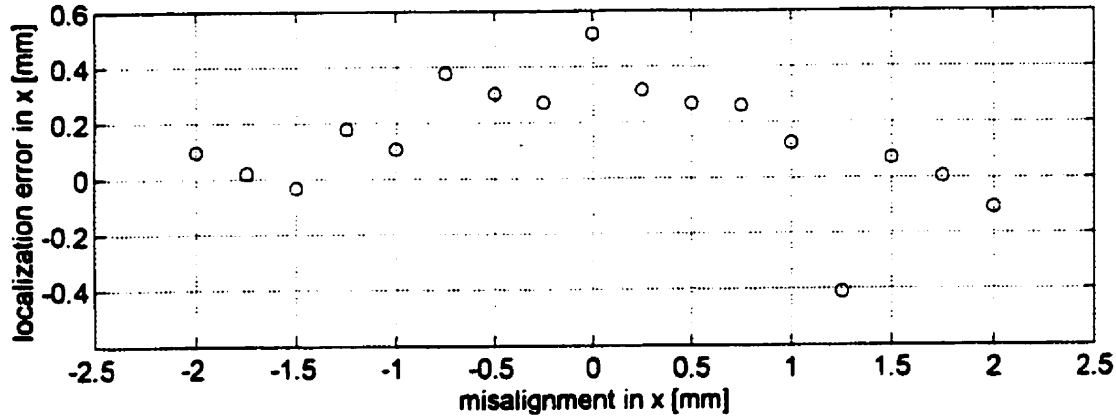

Figure 4-31: 6 DOF Forward Localization – x, misalignment in all axes

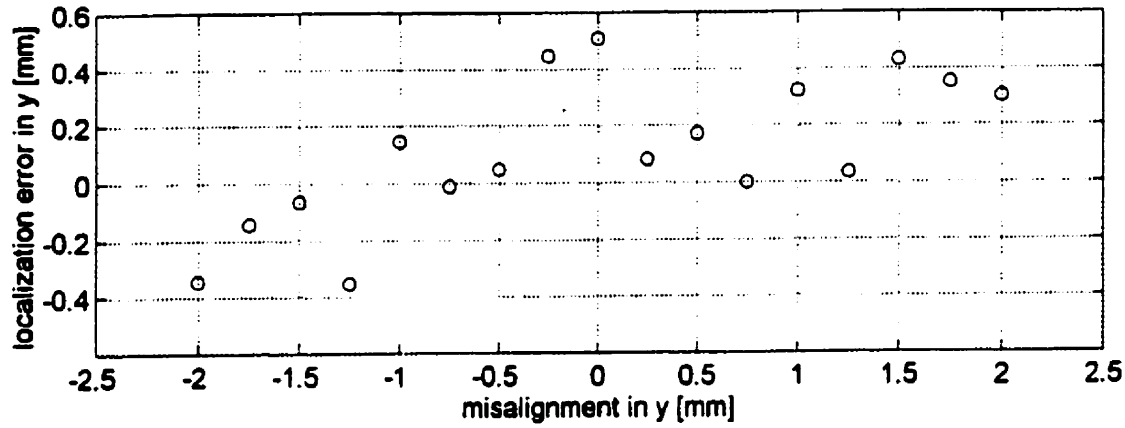

Figure 4-32: 6 DOF Forward Localization – y, misalignment in all axes

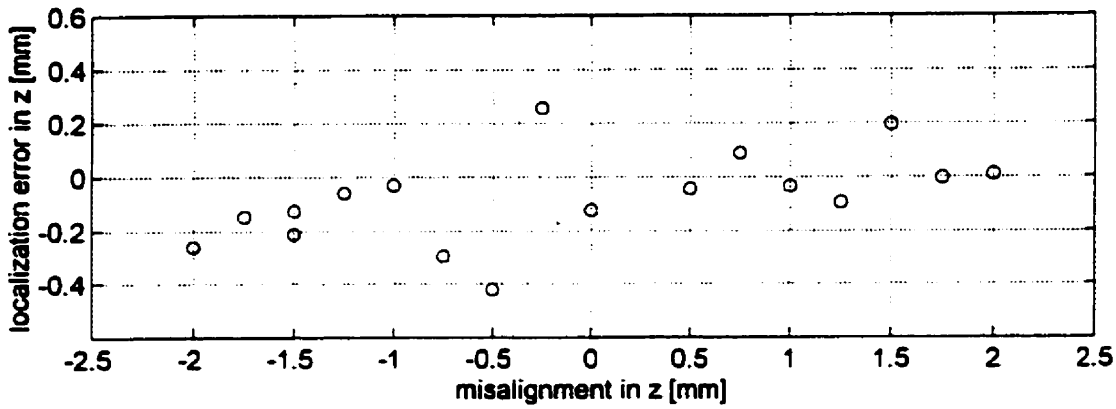

Figure 4-33: 6 DOF Forward Localization – z, misalignment in all axes

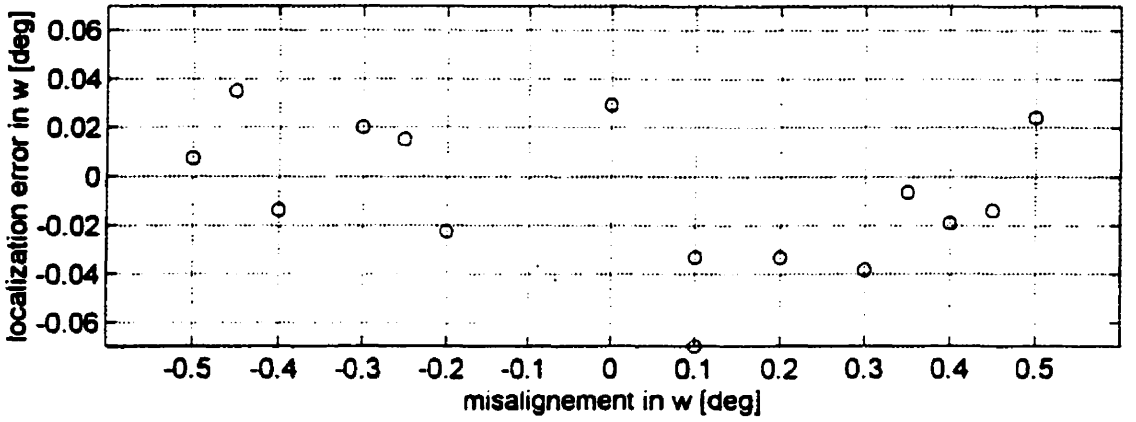

Figure 4-34: 6 DOF Forward Localization – *w*, misalignment in all axes

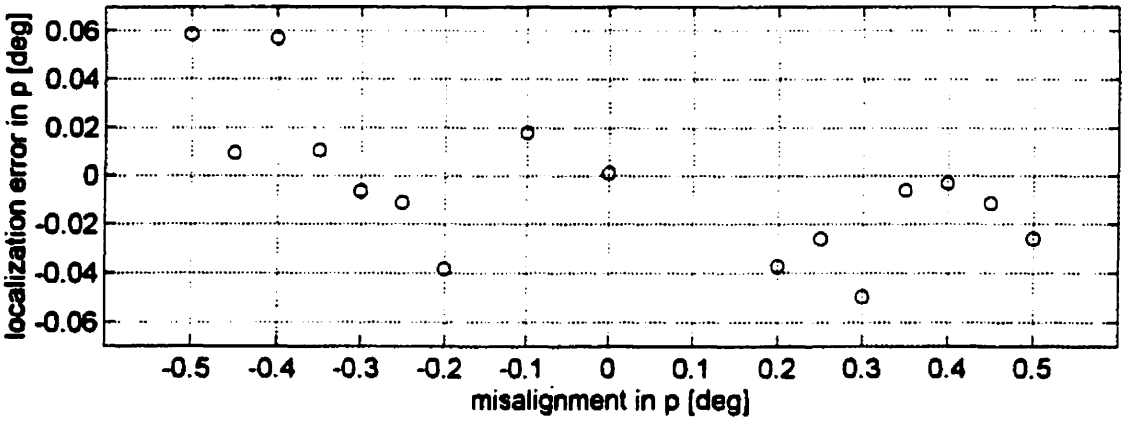

Figure 4-35: 6 DOF Forward Localization – *p*, misalignment in all axes

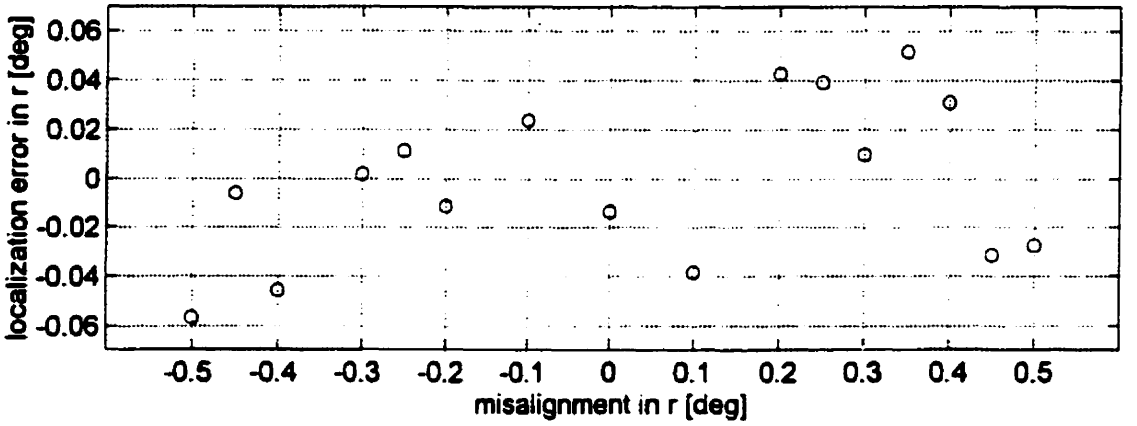

Figure 4-36: 6 DOF Forward Localization – *r*, misalignment in all axes

The above plots show that all 17 experiments have successfully localized the fender within the target range for all six coordinates.

4.5 Remarks

From the experimental results presented in the previous sections, the following remarks can be made:

- Misalignment in one coordinate direction has an effect on the localization errors of the other coordinates
- The localization performance is highly related to the number of sensors used. A reduced number of sensor measurements leads to greater localization error.
- The complexity in the shape of part also affects the localization performance. Murray et al. [27] suggested that the localization performance is largely determined by how well the mapping from the sensor measurements to part misalignments can be learned from the limited calibration data set. This implies is that both fewer number of sensor measurements and the complexity of the part's shape could affect the accurate mapping.
- The performance of localization can also be improved by increasing the number of points in the calibration sets [27] and by pointing the sensors to less curved locations on the fender. It is also important that sensors are positioned well apart of each other and perpendicular to the measuring surfaces of edges of the fender.
- Comparing to the inverse localization results, the forward localization results show coupled trade-offs in the performance between the coordinates involved. By adjusting the sensitivity weights, the performance of selected coordinates can be improved, while the other coordinates suffer.
- In case of the 6 DOF experiments, the computation time of the fender's misalignment was approximately 0.5 seconds for the inverse localization and 3 seconds for the forward localization with 500 iterations. This computation time

does not account for the robot motion – which was timed at 6 seconds from the pick-up of the fender to the taking of measurements under the sensor frame.

• As described in Chapter 3, the resolution of the proximity sensor is $0.02mm$ or less at a response time of $40ms$. The edge sensor has the resolution of $0.01mm$ or less and a response time of $0.5ms$. The pumping sound from the vacuum generator also contributes to the noise in the sensor measurements: less than $0.01mm$ which is not very significant. Another source of the error could come from the S-110 robot which has repeatability as big as $\pm 0.2mm$ in the translational coordinates at its maximum extension and speed.

4.6 Summary

Experimental results are presented for the part localization algorithm developed in Chapter 2. The results from the experiments clearly demonstrate the feasibility of implementing the part localization algorithm on a commercial robot. The proposed algorithm meets the target performance: finding the misalignment of the fender within the localization errors of $\pm 0.5mm$ for the x, y, and z coordinates, and $\pm 0.07°$ for the w, p, and r angular coordinates.

Chapter 5

Concluding Remarks

5.1 Summary and Conclusions

The primary goal of this research, which is to develop and implement a real-time automatic part localization system on a commercial robot (Fanuc S-110) in a fixtureless assembly scheme, was successfully achieved. In this thesis, the problem of part localization was studied in the context of an application to the assembly of car body fenders. First, an effective part localization method – called the direct calibration method [27] – was chosen, after an extensive literature review. This method was the most suitable approach available in localizing sheet metal parts, such as a car body fender. The formulation of the algorithm for the localization of the fender was described. In addition, the modification of this algorithm to three types of the misalignment of the fender are presented in detail. The three types of misalignments studied in this thesis are: 3 DOF surface misalignment (z, w, p coordinates), 3 DOF edge misalignment (x, y, r coordinates), and 6 DOF misalignment (all coordinates).

Secondly, the development of Automatic Part Localization System (APLS) in the Laboratory for Nonlinear Controls System was outlined. APLS is a demonstration system

which was built to prove the feasibility of the proposed part localization concept. It comprises various hardware and software components. Besides the Fanuc industrial robot, the sensor system is the core feature of APLS, used for detecting and measuring the misalignments of the fender grasped by the robot. A total of seven laser beam sensors are used in the system: three proximity sensors and four edge sensors. The proximity sensors are used to measure the fender's surface misalignment coordinates, whereas the edge sensors measure the edge misalignment coordinates. Other hardware components include a vacuum-cup gripper which grasps the fender, a sensor positioning frame which enables robust measurement positions for the sensors, an external data acquisition mainframe and a host PC which is a Pentium 166 MHz. The APLS's software modules were also documented in detail. The modules were developed to perform the PC-to-robot interface, PC-to-sensor system interface, and most importantly, to implement the localization algorithm.

The part localization algorithm and APLS were tested and verified through experiments. The experimental results clearly demonstrate that the proposed method meets the target performance of this investigation: finding the misalignment of the fender within the localization errors of $\pm 0.5mm$ for the x, y, and z coordinates, and $\pm 0.07°$ for the w, p, and r angular coordinates. To the best of our knowledge, the experiments results presented in this thesis are currently one of very few successful 6 DOF localization of a complex shaped part (i.e., car body fender). The results reported by Murray and Polhammer [27] was only obtained from simulation results for 3 DOF localization. The results of this investigation show a significant promise in stepping towards improving the high-speed automatic assembly by eliminating dedicated fixtures.

5.2 Recommendations for Future Work

Based on the work reported in this thesis, the following topics could be studied in future works:

1. The present research could be extended by solving the part localization algorithm with neural networks. Murray and Pohlhammer presented the neural network approach of solving their 3 DOF localization problem in [26]. It would be interesting to study the performance of neural network localization algorithm for the 6 DOF localization of complex shaped part such as the fender.

2. Experiments could be conducted to study the effect of misalignment in one coordinate to the localization error of other coordinates.

3. Experiments could be conducted to find the optimum values for the sensitivity weights and the number of iteration in the forward localization.

4. As [27] suggested, this thesis could be extended by studying the error due to the iterative inversion step of the forward mapping and the error due to the mapping of the least squares techniques.

5. Experiments could be carried out for multiple number of fenders (of the same type) to study the effect of part-to-part variations in fender's geometry to the localization performance. Murray and Pohlhammer [27] showed that superior performance is achieved with the forward localization, over the inverse, in the presence of the part-to-part variations in the geometry.

Bibliography

[1] Aromat Corporation, *NaiS Laser Beam Sensor UZL110 Operating Instructions*, Automation Controls Group, Matsushita Electric Works Ltd.

[2] Aromat Corporation, *NaiS LM100/200 Laser Analog Sensors: Laser Measurement for a wide range of application.*, Automation Controls Group, Matsushita Electric Works Ltd.

[3] Aromat Corporation, *NaiS LM100, LM200 Operating Instructions.*, Automation Controls Group, Matsushita Electric Works Ltd.

[4] Ayyadevara, V. R. and Sturges, R. H., "Analysis of Stability of Sheet Metal Parts for Automated Handling", *Proc. of the Japan-USA Symposium of Flexible Automation 1996*, pp. 891-893, 1996.

[5] Bolles, R. C. and Horaud, P., "3DPO: A Three-Dimensional Part Orientation System", Int'l J. Robotics Research, pp. 3-26, Fall 1986.

[6] Cambell, J., *C Programmer's Guide to Serial Communications*, Indianapolis, Ind.: Sams Pub., 1994.

[7] Daytronic Corporation, System 10 Guidebook, Miamisburg, OH, 1986.

[8] Faugeras, O. D. and Herbert, M., "The Representation, Recognition, and Location of 3-D Objects", Int'l J. of Robotics Research, Vol. 5(3), pp. 27-52, 1986.

[9] Gadagkar, H. P., Trivedi, M. M., and Lassiter, N., "Versatile Multimodal System for Surface Profile Measurements Using a Wrist-Mounted Laser Device", *Sensor Fusion V*, pp. 466-474, 1992.

[10] Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic Press, 1981.

[11] GMF Robotics Corporation, *Karel-PC Communications Software Reference Manual*, Ver. 1.0, Auburn Hills, Michigan, 1988.

[12] GMF Robotics Corporation, *Karel S-100 Description Manual*, Auburn Hills, Michigan, 1986.

[13] GMF Robotics Corporation, *Serial Communication Reference Manual*, Auburn Hills, Michigan, 1987.

[14] Grimson, W. E. and Lozano-Perez, T., "Model-Based Recognition and Localization from Sparse Range or Tactile Data", *Int'l J. Robotics Research*, pp. 3-35, Fall 1984.

[15] Gunnarsson, K. and Prinz, F. B., "CAD Model-Based Localization of Parts in Manufacuring", *Computer*, Vol. 20(8), pp. 66-74, 1987.

[16] He, D., Hujic, D., Mills, J. K., and Benhabib, B., "Moving-Object Recognition Using Premarking and Active Vision", *Proc. IEEE Int. Conf. on Robotics and Automation*, Minneapolis, MN, pp. 1980-1985, May 1996.

[17] Horaud, R. and Dornaika, F., "Hand-Eye Calibration", *Int'l. J. of Robotics Research*, Vol. 14(3), pp. 195-210, 1995.

[18] Hoska, D. R., "FLAM: What It Is. How to Achieve It", *Manufacturing Engineering*, pp. 49-54, April 1988.

[19] Ing, J. L., *Dynamic and Control of Fixtureless Assembly Using Robotic Manipulators*, M.A.Sc. Thesis, Department of Mechanical and Industrial Engineering, University of Toronto, 1994.

[20] Intelligent Instrumentation, Inc., *PC Based Data Acquisition Solutions*, 8th ed., U.S.A., 1995.

[21] Lawson, C. L. and Hanson, R. J., *Solving Least Squares Problems*, Englewood Cliffs, NJ: Prentice-Hall, 1974.

[22] Li., X. M., Yeung, M., and Li, Z. X., "An Algebraic Algorithm for Workpiece Localization", *Proc. IEEE Int. Conf. on Robotics and Automation*, Minneapolis, MN, May 1996.

[23] Liu, K., Jiang, R., Zhang, Q., Cheng, Y., and Yang, J., "Calibration of Multiple Sensors by a Planar Calibration Object", *Sensor Fusion V*, pp. 499-509, 1992.

[24] Mills, J. K. and Ing, J., "Robotic Fixtureless Assembly of Sheet Metal Parts Using Dynamic Finite Element Models: Modelling and Simulation", *Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, May 21-27, 1995.

[25] Menq, C. H., Yau, H., and Lai, G., "Automated Precision Measurement of Surface Profile in CAD-Directed Inspection", *IEEE Transactions on Robotics and Automation*, Vol. 8(2), pp. 268-278, 1992.

[26] Murray, W. R., Heg, C. T., and Pohlhammer, C. M., "Iterative Inversion of a Neural Network for Estimating the Location of a Planar Object", *1993 World Congress on Neural Networks*, Vol. 3, pp. 188-191, Portland, OR: INNS Press, 1993.

[27] Murray, W. and Pohlhammer, C., "Robust Estimation of the Location of a Planar Body: Results from Simulations", *Dynamic Systems and Control*, ASME, Vol. 55(2), pp.1103 - 1109, 1994.

[28] Neter, J., Wasserman, W., and Kunter, M. H., *Applied Linear Regression Models*, Irwin, Inc., 1989.

[29] Nguyen, W., *Fixtureless Assembly: Control of Multiple Robotic Manipulators Handling Flexible Payloads*, M.A.Sc. Thesis, Department of Mechanical and Industrial Engineering, University of Toronto, 1995.

[30] Nilsson, A. and Homberg, P., "Combining a Stable 2-D Vision Camera and an Ultrasonic Range Detector for 3-D Position Estimation", *IEEE Transactions on Instrumentation and Measurement*, Vol. 43(2), pp. 272-276, 1994.

[31] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed., Cambridge University Press., 1992.

[32] Sahoo, K. C. and Menq, C. H., "Localization of 3D Objects Using Surface Representation and Tactile Sensing", *Proc. Symp. Computer Aided Design Manufacturing. of Dies and Molds, ASME Winter Ann. Meet.*, Chicago, pp.105-118, 1988.

[33] Shiu, Y. C. and Ahmad, S., "Calibration of Wrist-Mounted Robotic Sensors by Solving Homogeneous Transform Equations of the Form $AX = XB$", *IEEE Transactions on Robotics and Automation*, Vol. 5(1), pp. 16-29, 1989.

[34] Tran, P.M., *Fanuc S110-R End-Effector Robotic Gripper Design*, B.A.Sc. Thesis, Department of Mechanical Engineering, University of Toronto, 1997.

[35] Tsai, R. Y. and Lenz, R. K., "A New Technique for Fully Autonomous and Efficient 3D Robotic Hand/Eye Calibration", *IEEE Transactions on Robotics and Automation*, Vol. 5(3), pp. 345-358, 1989.

[36] Wallack, A. S. and Canny, J. F., "Object Recognition and Localization from Scanning Beam Sensors", *IEEE Int'l Conf. on Robotics and Automation*, pp. 247-254, 1995.

[37] Zhou, T., *An Application of Part Localization to Robotic Fixtureless Assembly of Sheet Metal Parts*, Technical Report, Department of Mechanical and Industrial Engineering, University of Toronto, 1996.

Appendix A

Singular Value Decomposition

To find the least squares solution to an overdetermined set of linear equations – meaning there are more equations than unknowns – a technique called singular value decomposition (SVD) can be used. See [21] or [31] for the relevant theory on SVD. According to [31], SVD produces a solution that is the best approximation in the least squares sense.

From Chapter 2, consider Equation (2.7) of the inverse localization,

$$L_{CAL} \doteq \bar{S}_{CAL} \cdot C_{INV}$$

which is to be solved for the inverse calibration matrix,

$$C_{INV} = \bar{S}_{CAL}^{-1} \cdot L_{CAL}.$$

The SVD theorem, outlined by [31], can be applied here as the following: the $M \times N$ matrix $\bar{S}_{CAL}$ whose number of rows M is greater that or equal to its number of columns N, can be written as the product of an $M \times N$ column-orthogonal matrix U_I, an $N \times N$ diagonal matrix W_I with positive or zero elements (called singular values), and the transpose of an $N \times N$ orthogonal matrix V_I. Then, the SVD of $\bar{S}_{CAL}$ can be written as

$$\bar{S}_{CAL} = U_I \cdot W_I \cdot V_I^T \qquad \text{(A.1)}$$

where:

$$W_I = \left[diag\left(w_j \right) \right] = \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_N \end{bmatrix}. \tag{A.2}$$

From Equation (A.1), now the inverse of $\bar{S}_{CAL}$ is

$$\bar{S}_{CAL}^{-1} = V \cdot W_I^{-1} \cdot U_I^T \tag{A.3}$$

where:

$$W_I^{-1} = \left[diag\left(\frac{1}{w_j} \right) \right] = \begin{bmatrix} 1/w_1 & & \\ & \ddots & \\ & & 1/w_N \end{bmatrix}. \tag{A.4}$$

Therefore, the solution C_{INV} is obtained from

$$C_{INV} = V_I \cdot W_I^{-1} \cdot U_I^T \cdot L_{CAL}. \tag{A.5}$$

Similarly, the solution to the forward calibration matrix given by Equation (2.14),

$$C_{FOR} = \bar{L}_{CAL}^{-1} \cdot S_{CAL},$$

can be calculated from

$$C_{FOR} = V_F \cdot W_F^{-1} \cdot U_F^T \cdot S_{CAL}. \tag{A.6}$$

Appendix B

Sample Experimental Data

The following is the experimental data obtained for the *6 DOF localization experiments* presented in Section 4.4:

- C_{INV}:

```
 1.358609de+001 -2.5353787e+001 -3.9687309e+001 -1.1155202e-001 -1.5114794e-001  1.9763526e+000
-2.2214472e+000  7.6232622e+000  4.9705011e+000  1.6150215e-001 -1.1165528e-002 -3.8657549e-001
 3.0976698e+000 -2.8931913e+000  3.4751048e+000  4.5550696e-001 -4.0239876e-001 -4.5839913e-001
 8.4894772e-002 -5.3212284e-001  3.3944304e+000 -5.2009652e-001  4.2453590e-002  1.3412568e-001
-7.4172424e+000 -1.1738660e+000 -5.2908359e+000 -7.1296869e-001  2.1661940e-001  1.1023730e+000
 7.2282248e+000 -1.5140416e+000  4.4701878e+000  5.7241272e-001 -2.2211310e-001 -1.2762814e+000
 4.1655557e+000  2.3859892e+000  9.7495628e-001  4.8117396e-001 -2.0984470e-004 -1.5670395e-001
-5.1270830e+000  1.1310854e+000 -2.6455269e+000 -6.0360381e-001  2.2231382e-001  1.2415901e-001
-3.8705348e-001  7.4302812e-002  8.1679864e-001  1.7737585e-001  4.2560424e-002 -1.0131423e-001
 1.7462550e+000 -8.1358967e-001 -2.5215281e+000 -4.6542048e-001 -1.2451537e-001  3.6312577e-001
-3.3785330e-001  1.7499680e-002  3.8802964e-002  1.2578917e-001  4.8536757e-002 -4.9877483e-002
-7.2939876e-001  1.4766115e+000  9.4041027e-001  1.3835821e-001  1.2734101e-001 -2.5082826e-001
 8.4089995e-001 -1.5391542e+000 -1.4079840e+000 -2.2664293e-001 -1.5817014e-001  3.1509059e-001
 7.3373692e-001 -7.2908054e-001 -5.9082242e-001 -1.5443733e-001 -1.0789893e-001  1.7807246e-001
-9.0580611e-001  5.2335238e-001  1.2820850e+000  2.2843004e-001  1.0387220e-001 -2.1001347e-001
-1.7254782e+000  1.5127999e+000  2.2701700e+000  3.7827696e-001  2.0646028e-001 -3.6014051e-001
 1.8492806e+000 -1.8554116e+000 -1.5504345e+000 -3.3420569e-001 -2.9298015e-001  3.3500023e-001
-5.1643358e-001 -1.5243093e+000 -4.2879215e-001 -6.1768773e-002 -3.0278110e-002  1.1611617e-001
 3.8264999e-001  9.9354161e-001  4.5088332e-001  5.7882093e-002 -3.6533593e-002 -1.2888315e-001
-8.5565843e-001  1.1256068e+000  7.3300969e-001  1.7187677e-001  1.3507094e-001 -1.9613455e-001
 9.4667402e-001 -9.0850357e-001 -1.5385500e+000 -2.5562214e-001 -1.1569675e-001  2.2954576e-001
-1.1643909e+000  7.4413184e-001  3.8016247e-001  1.1826121e-001  1.6417216e-001 -1.2146505e-001
 2.0579354e+000 -9.7323972e-003  2.9593925e-001  1.9297474e-002 -1.3561165e-001 -1.3189058e-002
-2.5713410e+000  6.6980809e-001  1.5341618e-001  5.0112554e-002  2.2667193e-001 -1.1873246e-002
-4.7943046e-001 -1.9821579e-001 -3.0458799e-001 -7.4390118e-002 -2.3232086e-002  6.1942148e-002
 6.3959901e-001  1.7836323e-001  4.2486507e-001  9.2034020e-002 -3.8273206e-003 -6.8406781e-002
-7.6607026e-001  1.3077314e+000 -5.6176017e-001 -1.3684012e-001  1.3502048e-002  1.1018392e-001
 2.5333069e+000 -1.2557596e+000  9.9123925e-001  2.5206412e-001 -1.3611719e-001 -1.9992888e-001
 1.2153548e+000 -9.2427630e-001  1.4000737e-001  8.2909328e-002 -6.2912318e-002 -1.3505395e-002
-1.1816072e+000  1.2341414e+000 -4.5700462e-001 -8.3132792e-002  6.0330893e-002  5.8165507e-002
-1.9971249e+000  2.1048468e+000 -1.1834373e-001 -8.2880856e-002  1.3789953e-001  9.2106380e-002
-1.6360166e+000  9.0452735e-001 -4.2849516e-001 -9.0659177e-002  7.4573899e-002  4.2055978e-003
 1.7241375e+000 -1.1828202e+000  1.6353581e-001  7.0431904e-002 -8.3188146e-002 -4.4476338e-002
-5.0527001e-001  1.3654653e-001  2.3429365e-002 -1.0930526e-003  3.7781445e-002 -1.5659295e-002
 1.1066256e+000 -1.5713360e-001  2.5704108e-002 -1.4424437e-002 -8.1198562e-002  5.7167037e-002
-5.9484091e-001  1.5325037e-001  1.4565948e-001  2.8913128e-002  1.5078734e-002 -1.6537851e-002
```

- C_{FOR} :

```
 6.1248698e+000  4.7196805e+000  4.8177289e+000  4.4896624e+000  4.4847442e+000  4.5175021e+000  4.6438115e+000
-1.2241411e-001 -3.4205769e-002  5.6898858e-002 -3.2280871e-001 -3.2770279e-001  4.1425266e-002 -2.1781868e-002
-1.4561796e-002 -1.0755823e-003 -5.3623526e-003 -6.8424740e-003  2.3064705e-003  4.2042998e-001  3.1643708e-001
 1.3972318e-001  1.8951701e-001  1.8448127e-001 -3.3387742e-002 -3.4726451e-002 -6.4475366e-002  4.5946341e-002
 4.0194677e-001 -1.3154796e+000 -1.1202831e+000 -1.8830500e-001 -1.3314358e-001 -1.5648393e+000 -2.2653450e+000
-1.5288521e+000 -5.4322414e-001  1.2442543e+000 -2.3083123e+000 -2.3250581e+000  3.2513969e-001  1.1205119e-002
 3.9256406e-001 -1.3147545e-001  4.9591443e-001  8.3078965e-001 -3.1919708e-001  8.5393027e-001 -1.1930858e+000
 5.1045263e-003  3.1933105e-003 -2.6417293e-004 -7.2470924e-003 -7.7176268e-003  6.0803576e-003 -7.2580545e-004
-2.1122833e-003 -1.1579908e-005 -8.3434446e-004 -2.2222714e-003 -2.8712688e-004 -6.2458650e-003  1.3186913e-003
-1.0189191e-003 -2.1358083e-003 -1.4886654e-003 -2.5532900e-003 -3.5996269e-003 -1.3508033e-003 -2.6296112e-003
-1.3374941e-002  3.0308875e-003  8.3744149e-003 -3.5357828e-002 -2.4434826e-002  8.3816081e-003  7.9783213e-003
 7.4613607e-003  6.0061495e-003  1.1502765e-003 -1.0822859e-001 -1.0809912e-001 -2.0544476e-003 -6.9999262e-003
-1.7705188e-002  1.5120394e-002  9.6583655e-003  4.1259456e-002 -4.4894702e-002 -1.4880738e-002  5.0801731e-003
-4.7160833e-003 -5.9215360e-003 -5.0875549e-003  1.0654076e-004  4.9582784e-004 -1.0598794e-002 -1.1224875e-002
-6.6671749e-004  1.3985602e-003  1.8316306e-003  3.4731113e-004  3.7722941e-004  5.1434755e-003  5.0033818e-005
 7.9477342e-003  3.7823023e-003 -6.3357816e-003  1.9018163e-002  1.4802811e-002  8.8447703e-002  1.6207764e-001
-1.2594625e-003 -5.1378379e-004 -5.8494036e-003 -7.4934715e-003  5.2712633e-003 -3.3021777e-002 -6.6097584e-004
 1.4379833e-002 -2.8720187e-003 -9.0919545e-003  1.5243472e-002  1.1472564e-002 -6.5160322e-002  8.4818383e-002
 1.5910155e-003  5.2914270e-003  6.1099759e-003 -1.1434592e-003 -1.8560240e-003  1.0894775e-003  3.6285205e-003
 1.1756376e-002  1.6341542e-001 -5.5178804e-003  1.1734212e-002  1.1358366e-002 -2.2632544e-002  8.4788771e-003
-5.0637613e-004 -1.4600204e-002 -2.1099404e-002 -3.9749530e-003 -4.6176256e-003  1.1379458e-002 -1.0151717e-002
 4.4199564e-003 -1.9059693e-003 -1.3135164e-002  1.9502731e-002  1.1870712e-002  2.6143696e-003  2.8831981e-003
-1.7232153e-001 -1.5510008e-002  9.5463319e-002 -6.0295348e-002 -9.6560582e-002  1.2533287e-001 -3.2601775e-001
-2.8880757e-003 -6.9942475e-003  3.3372923e-002 -9.8828583e-002 -8.8369004e-002  7.4482170e-002 -3.2825232e-002
-4.7415539e-003  8.4016550e-003  5.2306089e-002  8.8668089e-002  1.1689992e-002  3.0657966e-001 -6.4244637e-001
 4.4517976e-002  1.5329498e-002 -3.3598409e-002 -3.0419658e-001 -3.0270873e-001 -1.0437083e-001 -9.3968364e-002
-1.3320175e-002  6.7203283e-002  3.4064281e-002  2.8886277e-001 -1.6744308e-001 -1.5371536e-001 -3.5862269e-002
-3.1482024e-002  6.1551721e-002  2.5784307e-002 -6.9233620e-002 -2.4042063e-002  8.3205288e-003 -5.9820215e-002
```

- **Resulting localization error for each coordinate from 129 *inverse localization experiments*:**

x (mm)	*y (mm)*	*z (mm)*	*w (°)*	*p (°)*	*r (°)*
3.1698793e-001	9.8624223e-002	-2.0562255e-001	-4.7962345e-002	-1.3984472e-002	2.2259673e-003
2.9408646e-001	4.1199832e-002	-2.3554153e-001	-4.6262170e-002	-2.8800439e-002	4.5802703e-003
2.8992035e-001	1.6206210e-002	-2.6557113e-001	-4.8863909e-002	-2.9216149e-002	6.9853701e-003
3.0281064e-001	-1.9474529e-002	-2.9323033e-001	-5.4332109e-002	-3.5399247e-002	1.2426797e-002
2.9969981e-001	-4.9702914e-003	-2.7783080e-001	-5.0260234e-002	-3.9898909e-002	1.9204446e-002
2.8533044e-001	-3.9731942e-003	-2.7407097e-001	-4.8560503e-002	-4.1137827e-002	1.8633665e-002
2.2224136e-001	-1.1304979e-002	-2.7159339e-001	-4.6835717e-002	-3.7925930e-002	2.0783584e-002
2.6142962e-001	-2.9196858e-002	-3.0112643e-001	-4.7636356e-002	-3.6994332e-002	2.1837711e-002
3.0645440e-001	-3.7142466e-002	-2.9561753e-001	-4.7166467e-002	-3.9030414e-002	2.3185719e-002
2.9738342e-001	-2.8579611e-002	-2.8938852e-001	-4.4255257e-002	-3.9085158e-002	2.5278945e-002
3.2810572e-001	-6.0704484e-002	-3.2351122e-001	-5.0480985e-002	-4.3355916e-002	2.7846934e-002
2.5817303e-001	-6.6889078e-002	-3.3614412e-001	-5.1325015e-002	-3.9769246e-002	3.0607937e-002
2.3289893e-001	-2.5584415e-002	-2.8344934e-001	-4.2372152e-002	-3.8224219e-002	3.2908011e-002
1.9450621e-001	-2.8796165e-002	-3.0098988e-001	-4.2952099e-002	-3.8210348e-002	3.5899541e-002
1.5793999e-001	-6.2018328e-002	-3.5097934e-001	-4.9183008e-002	-3.8101879e-002	3.9977091e-002
1.6500221e-001	-5.2238323e-002	-3.4761978e-001	-4.7645277e-002	-4.1301442e-002	4.2639460e-002
1.5363604e-001	-6.6407157e-002	-3.7514191e-001	-4.9896320e-002	-4.4855291e-002	4.5628658e-002
1.9647041e-001	1.0062841e-001	-1.9051775e-001	-5.9156742e-002	-2.9166482e-002	2.4957926e-002
1.4412646e-001	7.5687226e-002	-1.9392678e-001	-5.5083932e-002	-3.3436167e-002	2.5616285e-002
2.1638468e-001	4.7008162e-002	-2.7764071e-001	-5.8332178e-002	-4.5324777e-002	2.3407503e-002
1.3436391e-001	7.3966749e-002	-2.3776701e-001	-5.1297652e-002	-3.7050683e-002	2.4375216e-002
1.1649377e-001	5.3764576e-002	-2.3930798e-001	-4.7318114e-002	-3.8741996e-002	2.4301603e-002
1.7624873e-001	-3.3626565e-003	-2.9162783e-001	-5.0664768e-002	-5.1464230e-002	2.4788024e-002
1.2144664e-001	-9.3608354e-003	-2.6539365e-001	-3.7529604e-002	-4.9262426e-002	2.5257807e-002
8.2729287e-002	-1.7880880e-002	-2.5276212e-001	-3.4227136e-002	-4.6762877e-002	2.6500618e-002
9.4673046e-002	-3.8612604e-002	-2.6681632e-001	-3.4676354e-002	-5.0828342e-002	2.6906403e-002
5.1970842e-002	-3.2611934e-002	-2.6296962e-001	-3.6711446e-002	-4.5603743e-002	2.8403651e-002
9.7049295e-002	-7.8377276e-002	-2.8134482e-001	-3.9471470e-002	-5.1959968e-002	2.8398882e-002
1.0279135e-001	-9.5150852e-002	-2.9163887e-001	-3.7811561e-002	-5.3231878e-002	2.7994317e-002
9.9947391e-002	-1.1899087e-001	-3.1225738e-001	-3.6743141e-002	-5.3819426e-002	2.8601324e-002
1.4190790e-001	-1.9989425e-001	-2.8111371e-001	-4.0121522e-002	-6.1103602e-002	2.8508608e-002
5.2083751e-002	-1.9342763e-001	-2.2775359e-001	-3.2975740e-002	-5.0431381e-002	2.9199953e-002
8.5307308e-002	-2.0974511e-001	-1.9728699e-001	-2.8151625e-002	-5.7580236e-002	2.8586556e-002
2.3416128e-001	-2.4158335e-001	-2.2745453e-001	-3.1083487e-002	-6.9038514e-002	2.6615504e-002
1.7858822e-001	-3.9800924e-002	-4.8887490e-001	-6.6037185e-002	-2.6812527e-002	2.7592971e-002
1.1018992e-001	-4.4449478e-002	-4.3551286e-001	-5.1061750e-002	-3.3678604e-002	2.8341565e-002
1.0252222e-001	-4.8491770e-003	-3.6507834e-001	-4.3072698e-002	-3.3844344e-002	2.8040604e-002
1.1353130e-001	-5.5118187e-003	-3.4773106e-001	-4.5614681e-002	-3.3905222e-002	2.7837103e-002

```
 1.2225186e-001  7.6266191e-003 -3.4453138e-001 -4.3886546e-002 -3.9242896e-002  2.6724837e-002
 1.2819446e-001 -3.7488613e-003 -3.5451416e-001 -4.8176788e-002 -4.3135321e-002  2.7107170e-002
 1.3641829e-001 -3.3479978e-002 -3.0657932e-001 -4.4400292e-002 -4.1454656e-002  2.6360004e-002
 1.1977443e-001 -3.0646145e-002 -2.8339280e-001 -4.1295891e-002 -3.9226301e-002  2.5381863e-002
 1.0892195e-001 -1.2386993e-002 -2.6894650e-001 -4.2776484e-002 -4.1495630e-002  2.5971183e-002
 1.1375744e-001  2.8558775e-002 -2.7661676e-001 -4.2417658e-002 -4.3913257e-002  2.4939409e-002
 1.4380323e-001  3.1135493e-002 -2.8249405e-001 -4.6393469e-002 -4.3618191e-002  2.5154678e-002
 1.4554748e-001  2.5609425e-002 -2.7549812e-001 -4.6190886e-002 -4.2956439e-002  2.4879724e-002
 1.4386519e-001 -4.9689220e-003 -2.5798907e-001 -4.2930772e-002 -4.3277560e-002  2.5881112e-002
 1.3282988e-001  2.0951678e-002 -2.2022735e-001 -3.6972097e-002 -4.4529717e-002  2.5258335e-002
 1.6033933e-001  2.3096293e-002 -2.1332424e-001 -3.4449293e-002 -4.3137861e-002  2.5195286e-002
 1.5568961e-001  2.5710728e-002 -1.9252277e-001 -3.0607730e-002 -4.0857229e-002  2.6100430e-002
 1.9628107e-001  1.3466134e-002 -2.3109219e-001 -3.1585833e-002 -3.9694781e-002  2.5992192e-002
 2.2435541e-001 -4.3468569e-002 -1.7174253e-001 -3.6689063e-002 -6.1142240e-002  1.6399811e-002
 1.0295919e-001 -6.0928431e-002 -1.8516042e-001 -3.7607686e-002 -3.6812847e-002  3.0085413e-002
 3.6653395e-001 -1.2291376e-001 -3.2049554e-001 -4.8736568e-002 -6.2089100e-002  1.8893246e-002
 1.7852645e-001 -6.7929889e-002 -2.5907107e-001 -4.0094613e-002 -4.2936584e-002  2.3123068e-002
 1.8785278e-001 -1.0403514e-001 -2.9107060e-001 -4.1830915e-002 -4.7506484e-002  2.0384527e-002
 8.6806754e-002 -1.1882866e-001 -2.4812638e-001 -1.5114168e-002 -4.9906846e-002  2.4781665e-002
 7.5590886e-002 -1.2394558e-002 -2.1831858e-001 -7.3691125e-003 -5.1607777e-002  2.2065727e-002
 1.0845556e-001  7.1979125e-003 -2.1702388e-001 -7.0527625e-003 -5.0254669e-002  1.6762346e-002
 4.7728003e-002  1.2902040e-001 -1.4554707e-001  7.0846974e-003 -3.8609397e-002  1.5630104e-002
 1.0518519e-001  2.4299899e-001 -8.0947757e-002  1.6986450e-002 -4.4310146e-002  1.2294993e-002
-5.2647402e-002  3.0321169e-001  3.2649595e-002  3.0390054e-002 -2.1542144e-002  8.2667916e-003
-4.9699006e-003  1.3688030e-001 -2.4657770e-001 -4.5102896e-002 -2.3860553e-003 -3.0182500e-003
 2.2290126e-001  4.1913481e-002 -2.6964023e-001 -4.8302695e-002 -2.3326048e-002  3.3536574e-003
 2.8050484e-001  4.6949007e-002 -2.3827239e-001 -4.0652065e-002 -2.9411064e-002  5.6572445e-003
 3.2500120e-001  2.6348092e-002 -2.4853467e-001 -4.1532810e-002 -3.6459896e-002  1.4303940e-002
 2.9135791e-001  2.5818478e-002 -2.3818955e-001 -3.9620255e-002 -3.1032231e-002  2.5554469e-002
 1.8566781e-001  1.3620528e-002 -2.7488162e-001 -4.7249324e-002 -1.9746152e-003  2.5473917e-002
 1.3622344e-001 -2.3925815e-003 -2.9983016e-001 -5.2879380e-002  6.2368977e-004  2.6065372e-002
 5.2953153e-002  4.9589899e-003 -2.8663677e-001 -5.1780513e-002  9.7816086e-003  2.3954405e-002
-7.0028253e-002 -9.2151285e-004 -3.3219584e-001 -5.5964308e-002  2.1450442e-002  2.1327028e-002
-1.3144686e-001  4.5053235e-002 -3.1585754e-001 -5.3540185e-002  2.3153198e-002  1.2710074e-002
-1.7906300e-001  4.9869977e-002 -3.4960039e-001 -6.1133038e-002  3.3095626e-002  7.5207657e-004
 4.9004721e-001 -2.0125040e-001 -3.8962108e-001 -6.9316793e-002 -6.1526378e-002  2.1559057e-002
 4.2967195e-001 -1.2183610e-001 -3.1998582e-001 -5.9274447e-002 -5.5689117e-002  2.7647756e-002
 4.0203972e-001 -9.2141489e-002 -3.1992745e-001 -5.3664794e-002 -4.0479693e-002  1.8618149e-002
 3.6804686e-001 -7.3535431e-002 -3.2349240e-001 -5.2864400e-002 -3.6774577e-002  1.9733942e-002
 3.2886308e-001 -2.9649812e-002 -2.7821446e-001 -4.5664419e-002 -3.2928134e-002  1.8773803e-002
 2.7957339e-001 -1.2190435e-002 -2.8909277e-001 -4.3220741e-002 -2.3946444e-002  2.6995159e-002
 3.0122662e-001  5.0340219e-003 -2.4342562e-001 -4.1556468e-002 -2.7809065e-002  3.1367540e-002
 3.2772298e-001 -8.2074439e-004 -2.6438222e-001 -4.2761947e-002 -3.1623172e-002  3.0287965e-002
 3.0583878e-001  1.1780237e-002 -2.2195348e-001 -4.1011302e-002 -2.6627189e-002  2.7325778e-002
 2.6374192e-001  5.2748970e-002 -1.4652796e-001 -3.0953252e-002 -2.0921975e-002  1.7970895e-002
 2.5197553e-001  3.1489895e-002 -1.1500391e-001 -3.6821129e-002 -1.6559210e-002  1.1584218e-002
 3.4089034e-001 -1.6975544e-001 -1.5770141e-001 -1.3289601e-002 -5.2034305e-002  6.5538940e-003
 2.3287108e-001  1.0681872e-001 -1.2673820e-001 -4.2273379e-002  5.3116504e-004  5.8266794e-003
 2.6114425e-001  3.9341401e-002 -1.5028998e-001 -2.6737868e-002 -2.7215186e-002  1.4859577e-002
 3.1634406e-001  5.2064164e-002 -2.9103013e-001 -5.5231943e-002 -2.2702242e-002  8.8599592e-003
 4.2056147e-001 -2.2618412e-001 -1.9368513e-001 -2.4567175e-002 -6.0595518e-002  3.3271535e-002
 3.1553894e-001  1.5368945e-001 -9.7648174e-002 -4.4330432e-002 -2.1662954e-002  1.7189833e-002
 2.5512414e-001 -1.1861075e-001 -2.3298467e-001 -3.0398935e-002 -4.4946652e-002  2.5131157e-002
 2.0768153e-001 -1.9615607e-001 -2.5730802e-001 -1.4620067e-002 -3.9049589e-002  2.2293673e-002
 4.3516969e-001  9.3063294e-002 -2.3421764e-001 -4.6069245e-002 -3.4142496e-002  2.3497020e-002
 3.6228205e-001  5.6627043e-003 -3.0907513e-001 -3.6015208e-002 -3.8310396e-002  2.5489186e-002
 3.1231656e-001 -3.4694373e-002 -1.9672723e-001 -3.1576856e-002 -3.8826272e-002  2.5547094e-002
 2.5327141e-001  4.2209420e-002 -1.9892307e-001 -3.9818674e-002 -2.6108481e-002  2.7224252e-002
 1.4840643e-001 -2.3477968e-001 -3.5059254e-001 -1.9266304e-002 -3.6659069e-002  3.7279542e-002
 3.5582884e-001 -1.7615438e-001 -4.9845596e-001 -4.0770459e-002 -5.1757577e-002  3.5988900e-002
 3.0399728e-001 -2.1524354e-002 -2.7681626e-001 -5.9988479e-002 -3.2490323e-002  3.6371883e-002
 1.4505097e-001 -1.7585167e-001 -2.3854276e-001 -1.2793647e-002 -5.7478148e-002  4.2587868e-002
 2.8856070e-001 -4.7588479e-002 -4.5013630e-001 -6.0391790e-002 -3.5763369e-002  5.9650689e-002
 1.1520735e-001  1.6134980e-001 -2.1085326e-001 -2.1553946e-002 -4.3854351e-002 -3.6858739e-003
-2.5192517e-001  7.5442562e-002 -2.8738423e-001 -5.6860832e-002  3.6479555e-002  1.1598538e-002
 2.7199813e-001  3.8508751e-001 -3.8741853e-003  9.4415830e-003 -4.1560685e-002 -1.8137986e-002
-1.0888983e-001  1.4656658e-001 -1.7866892e-001 -1.4387729e-002 -3.2844757e-003 -9.4778180e-003
 4.3410496e-001 -5.7031042e-002 -3.1462068e-001 -6.2702540e-002 -5.0806948e-002  1.1414186e-002
 8.9392129e-002 -1.6572144e-002 -3.8261321e-001 -6.6921388e-002  1.0606696e-002  1.4001042e-002
 2.0621151e-001 -9.9537204e-003 -3.2676691e-002 -1.9519481e-002 -2.1949003e-002 -4.1146300e-003
 2.0419768e-001 -1.1838770e-001 -2.9806867e-001 -4.8130838e-002 -3.6000628e-002  1.2972872e-002
 1.7709898e-001  2.6286778e-001  2.2590086e-003  1.9105632e-002 -1.4369288e-002  2.0408337e-002
 2.3390253e-001 -3.3842149e-003 -2.6609590e-001 -6.5322656e-002  8.6341353e-004  1.5588331e-002
 3.7735753e-001 -4.1674179e-003 -2.2267329e-001 -5.5019736e-002 -2.9370178e-002  2.6745791e-002
-1.3048922e-001 -1.4883804e-001 -3.2402338e-001 -9.2307460e-003 -5.5763831e-002  6.4330864e-003
-1.0755957e-001  1.6952772e-001 -1.6575343e-001 -4.8704393e-002  5.8959760e-002 -5.6700614e-003
 8.6832995e-002  3.2346105e-001  1.1716746e-001  2.7262254e-002 -3.8409443e-002 -1.0980370e-002
-1.6234182e-001  3.1002639e-001  2.3494291e-002  3.4094815e-003  3.0559421e-002 -3.7086854e-002
 1.3707443e-001 -2.4762747e-001 -5.5930095e-002 -1.3665697e-002 -5.8801576e-002  3.9083200e-002
 1.1295760e-001  6.7259033e-002 -1.8173346e-001 -6.7207552e-002  1.2621537e-002 -1.3211372e-003
 3.5488511e-001  2.0408166e-003 -1.5375300e-002 -1.8096466e-002 -3.6195311e-002  2.6240863e-003
 3.0511213e-001 -1.2009512e-001 -2.2952727e-001 -3.1202950e-002 -3.2781924e-002  8.7934650e-003
 4.3495558e-001 -1.5047384e-002 -1.4650346e-001  1.2594262e-003 -6.7796512e-002  4.8209154e-002
 3.8304351e-001  3.1834668e-002 -2.5958563e-001 -5.5905251e-002  6.1854212e-003  1.9212441e-002
```

3.5321883e-001	-3.3608211e-002	-2.4092155e-001	-6.1604568e-002	-5.0900412e-003	3.3189282e-002
3.5798065e-001	5.1165959e-002	-3.6725442e-002	-3.1642827e-002	-2.7701161e-002	2.1094391e-002
2.5005281e-001	1.3674766e-001	-5.6519520e-002	3.2226886e-002	-2.6128471e-002	8.1847259e-003
7.4789606e-002	1.6178282e-002	-1.8247994e-001	-2.4570292e-002	5.1722469e-003	-1.8665231e-002
8.6489450e-002	6.9280749e-002	-1.5411682e-001	-2.0709736e-002	4.6344525e-003	1.0749026e-002
1.9651816e-001	-2.2237382e-001	-3.3306903e-001	-5.9193092e-002	-4.0287546e-003	2.5464416e-002
6.9694520e-002	1.4210279e-001	-1.4132025e-001	-3.0826969e-004	2.7247167e-002	-4.6629301e-002

- **Resulting localization error for each coordinate from 129 *forward localization experiments*:**

x (mm)	*y (mm)*	*z (mm)*	*w (°)*	*p (°)*	*r (°)*
3.1698793e-001	9.8624223e-002	-2.0562555e-001	-4.7962345e-002	-1.3984472e-002	2.2259673e-003
2.9408646e-001	4.1199832e-002	-2.3554153e-001	-4.6262170e-002	-2.8800439e-002	4.5802703e-003
2.8992035e-001	1.6206210e-002	-2.6557113e-001	-4.8863909e-002	-2.9216149e-002	6.9853701e-003
3.0281064e-001	-1.9474529e-002	-2.9323033e-001	-5.4332109e-002	-3.5399247e-002	1.2426797e-002
2.9969981e-001	-4.9702914e-003	-2.7783080e-001	-5.0260234e-002	-3.9898909e-002	1.9204446e-002
2.8533044e-001	-3.9731942e-003	-2.7407097e-001	-4.8560503e-002	-4.1137827e-002	1.8633665e-002
2.2224136e-001	-1.1304979e-002	-2.7159339e-001	-4.6835717e-002	-3.7925930e-002	2.0783584e-002
2.6142962e-001	-2.9196858e-002	-3.0112643e-001	-4.7636356e-002	-3.6994332e-002	2.1837711e-002
3.0645440e-001	-3.7142466e-002	-2.9561753e-001	-4.7166467e-002	-3.9030414e-002	2.3185719e-002
2.9738342e-001	-2.8579611e-002	-2.8938852e-001	-4.4255257e-002	-3.9085158e-002	2.5278945e-002
3.2810572e-001	-6.0704484e-002	-3.2351122e-001	-5.0480985e-002	-4.3355916e-002	2.7846934e-002
2.5817303e-001	-6.6889078e-002	-3.3614412e-001	-5.1325015e-002	-3.9769246e-002	3.0607937e-002
2.3289893e-001	-2.5584415e-002	-2.8344934e-001	-4.2372152e-002	-3.8224219e-002	3.2908011e-002
1.9450621e-001	-2.8796165e-002	-3.0098988e-001	-4.2952099e-002	-3.8210348e-002	3.5899541e-002
1.5793999e-001	-6.2018328e-002	-3.5097934e-001	-4.9183008e-002	-3.8101879e-002	3.9977091e-002
1.6500221e-001	-5.2238323e-002	-3.4761978e-001	-4.7645277e-002	-4.1301442e-002	4.2639460e-002
1.5363604e-001	-6.6407157e-002	-3.7514191e-001	-4.9896320e-002	-4.4855291e-002	4.5628658e-002
1.9647041e-001	1.0062841e-001	-1.9051775e-001	-5.9156742e-002	-2.9166482e-002	2.4957926e-002
1.4412646e-001	7.5687226e-002	-1.9392678e-001	-5.5083932e-002	-3.3436167e-002	2.5616285e-002
2.1638468e-001	4.7008162e-002	-2.7764071e-001	-5.8332178e-002	-4.5324777e-002	2.3407503e-002
1.3436391e-001	7.3966749e-002	-2.3776701e-001	-5.1297652e-002	-3.7050683e-002	2.4375216e-002
1.1649377e-001	5.3764576e-002	-2.3930798e-001	-4.7318114e-002	-3.8741996e-002	2.4301603e-002
1.7624873e-001	-3.3626565e-003	-2.9162783e-001	-5.0664768e-002	-5.1464230e-002	2.4788024e-002
1.2144664e-001	-9.3608354e-003	-2.6539365e-001	-3.7529604e-002	-4.9262426e-002	2.5257807e-002
8.2729287e-002	-1.7880880e-002	-2.5276212e-001	-3.4227136e-002	-4.6762877e-002	2.6500618e-002
9.4673046e-002	-3.8612604e-002	-2.6681632e-001	-3.4676354e-002	-5.0828342e-002	2.6906403e-002
5.1970842e-002	-3.2611934e-002	-2.6296962e-001	-3.6711446e-002	-4.5603743e-002	2.8403651e-002
9.7049295e-002	-7.8377276e-002	-2.8134482e-001	-3.9471470e-002	-5.1959968e-002	2.8398882e-002
1.0279135e-001	-9.5150852e-002	-2.9163887e-001	-3.7811561e-002	-5.3231878e-002	2.7994317e-002
9.9947391e-002	-1.1899087e-001	-2.7225738e-001	-3.6743141e-002	-5.3819426e-002	2.8601324e-002
1.4190790e-001	-1.9989425e-001	-2.8111371e-001	-4.0121522e-002	-6.1103602e-002	2.8508608e-002
5.2083751e-002	-1.9342763e-001	-2.2775359e-001	-3.2975740e-002	-5.0431381e-002	2.9199953e-002
8.5307308e-002	-2.0974511e-001	-1.9728699e-001	-2.8151625e-002	-5.7580236e-002	2.8586556e-002
2.3416128e-001	-2.4158335e-001	-2.2745453e-001	-3.1083487e-002	-6.9038514e-002	2.6615504e-002
1.7858822e-001	-3.9800924e-002	-4.8887490e-001	-6.6037185e-002	-2.6812527e-002	2.7592971e-002
1.1018992e-001	-4.4449478e-002	-4.3551286e-001	-5.1061750e-002	-3.3678604e-002	2.8341565e-002
1.0252222e-001	-4.8491770e-003	-3.6507834e-001	-4.3072698e-002	-3.3844344e-002	2.8040604e-002
1.1353130e-001	-5.5118187e-003	-3.4773106e-001	-4.5614681e-002	-3.3905222e-002	2.7837103e-002
1.2225186e-001	7.6266191e-003	-3.4453138e-001	-4.3886546e-002	-3.9242896e-002	2.6724837e-002
1.2819446e-001	-3.7488613e-003	-3.5451416e-001	-4.8176788e-002	-4.3135321e-002	2.7107170e-002
1.3641829e-001	-3.3479978e-002	-3.0657932e-001	-4.4400292e-002	-4.1454656e-002	2.6360004e-002
1.1977443e-001	-3.0646145e-002	-2.8339280e-001	-4.1295891e-002	-3.9226301e-002	2.5381863e-002
1.0892195e-001	-1.2386993e-002	-2.6894650e-001	-4.2776484e-002	-4.1495630e-002	2.5971183e-002
1.1375744e-001	2.8558775e-002	-2.7861676e-001	-4.2417658e-002	-4.3913257e-002	2.4939409e-002
1.4380323e-001	3.1135493e-002	-2.8249405e-001	-4.6393469e-002	-4.3618191e-002	2.5154678e-002
1.4554748e-001	2.5609425e-002	-2.7549812e-001	-4.6190886e-002	-4.2956439e-002	2.4879724e-002
1.4386519e-001	-4.9689220e-003	-2.5798907e-001	-4.2930772e-002	-4.3277560e-002	2.5881112e-002
1.3282988e-001	2.0951678e-002	-2.2022735e-001	-3.6972097e-002	-4.4529717e-002	2.5258335e-002
1.6033933e-001	2.3096293e-002	-2.1332424e-001	-3.4449293e-002	-4.3137861e-002	2.5195286e-002
1.5568961e-001	2.5710728e-002	-1.9252277e-001	-3.0607730e-002	-4.0857229e-002	2.6100430e-002
1.9628107e-001	1.3466134e-002	-2.3109219e-001	-3.1585833e-002	-3.9694781e-002	2.5992192e-002
2.2435541e-001	-4.3468569e-002	-1.7174253e-001	-3.6689063e-002	-6.1142240e-002	1.6399811e-002
1.0295919e-001	-6.0928431e-002	-1.8516042e-001	-3.7607686e-002	-3.6812847e-002	3.0085413e-002
3.6653395e-001	-1.2291376e-001	-3.2049554e-001	-4.8736568e-002	-6.2089100e-002	1.8893246e-002
1.7852645e-001	-6.7929889e-002	-2.3907107e-001	-4.0094613e-002	-4.2936584e-002	2.3123068e-002
1.8785278e-001	-1.0403514e-001	-2.9107060e-001	-4.1830915e-002	-4.7506484e-002	2.0384527e-002
8.6806754e-002	-1.1882866e-001	-2.4812638e-001	-4.5114168e-002	-4.9906846e-002	2.4781665e-002
7.5590886e-002	-1.2394558e-002	-2.1831858e-001	-7.3691125e-003	-5.1607777e-002	2.2065727e-002
1.0845556e-001	7.1979125e-003	-2.1702388e-001	-7.3527625e-003	-5.0254669e-002	1.9762346e-002
4.7728003e-002	1.2902040e-001	-1.4554707e-001	7.3846974e-003	-3.8609397e-002	1.5630104e-002
1.0518519e-001	2.4299899e-001	-8.0947757e-002	1.6986450e-002	-4.4310146e-002	1.2294993e-002
-5.2647402e-002	3.0321169e-001	3.2649595e-002	3.0390054e-002	-2.1542144e-002	8.2667916e-003
-4.9699006e-003	1.3688030e-001	-2.4657770e-001	-4.5102896e-002	-2.3860553e-003	-3.0182500e-003

2.2290126e-001	4.1913481e-002	-2.6964023e-001	-4.8302695e-002	-2.3326048e-002	3.3536574e-003
2.8050484e-001	4.6949007e-002	-2.3827239e-001	-4.0652065e-002	-2.9411064e-002	5.6572445e-003
3.2500120e-001	2.6348092e-002	-2.4853467e-001	-4.1532810e-002	-3.6459896e-002	1.4303940e-002
2.9135791e-001	2.5818478e-002	-2.3818955e-001	-3.9620255e-002	-3.1032231e-002	2.5554469e-002
1.8566781e-001	1.3620528e-002	-2.7488162e-001	-4.7249324e-002	-1.9746152e-003	2.5473917e-002
1.3622344e-001	-2.3925815e-003	-2.9983016e-001	-5.2879380e-002	6.2368977e-004	2.6065372e-002
5.2953153e-002	4.9589899e-003	-2.8663677e-001	-5.1780513e-002	9.7816086e-003	2.3954405e-002
-7.0028253e-002	-9.2151285e-004	-3.3219584e-001	-5.5964308e-002	2.1450442e-002	2.1327028e-002
-1.3144686e-001	4.5053235e-002	-3.1585754e-001	-5.3540185e-002	2.3153198e-002	1.2710074e-002
-1.7906300e-001	4.9869977e-002	-3.4960039e-001	-6.1133038e-002	3.3095626e-002	7.5207657e-004
4.9004721e-001	-2.0125040e-001	-3.8962108e-001	-6.9316793e-002	-6.1526378e-002	2.1559057e-002
4.2967195e-001	-1.2183610e-001	-3.1998582e-001	-5.9274447e-002	-5.5689117e-002	2.7647756e-002
4.0203972e-001	-9.2141489e-002	-3.1992745e-001	-5.3664794e-002	-4.0479693e-002	1.8618149e-002
3.6804686e-001	-7.3535431e-002	-3.2349240e-001	-5.2864400e-002	-3.6774577e-002	1.9733942e-002
3.2886308e-001	-2.9649812e-002	-2.7821446e-001	-4.5664419e-002	-3.2928134e-002	1.8773803e-002
2.7957339e-001	-1.2190435e-002	-2.8909277e-001	-4.3220741e-002	-2.3946444e-002	2.6995159e-002
3.0122662e-001	5.0340219e-003	-2.4342562e-001	-4.1556468e-002	-2.7809065e-002	3.1367540e-002
3.2772298e-001	-8.2074439e-004	-2.6438222e-001	-4.2761947e-002	-3.1623172e-002	3.0287965e-002
3.0583878e-001	1.1780237e-002	-2.2195348e-001	-4.1011302e-002	-2.6627189e-002	2.7325778e-002
2.6374192e-001	5.2748970e-002	-1.4652796e-001	-3.0953252e-002	-2.0921975e-002	1.7970895e-002
2.5197553e-001	3.1489895e-002	-1.1500391e-001	-3.6821129e-002	-1.6559210e-002	1.1584218e-002
3.4089034e-001	-1.6975544e-001	-1.5770141e-001	-1.3289601e-002	-5.2034305e-002	6.5538940e-003
2.3287108e-001	1.0681872e-001	-1.2673820e-001	-4.2273379e-002	5.3116504e-004	5.8266794e-003
2.6114425e-001	3.9341401e-002	-1.5028998e-001	-2.6737868e-002	-2.7215186e-002	1.4859577e-002
3.1634406e-001	5.2064164e-002	-2.9103013e-001	-5.5231943e-002	-2.2702242e-002	8.8599592e-003
4.2056147e-001	-2.2618412e-001	-1.9368513e-001	-2.4567175e-002	-6.0595518e-002	3.8271535e-002
3.1553894e-001	1.5368945e-001	-9.7648174e-002	-4.4330432e-002	-2.1662954e-002	1.7189833e-002
2.5512414e-001	-1.1861075e-001	-2.3298467e-001	-3.0398935e-002	-4.4946652e-002	2.5131157e-002
2.0768153e-001	-1.9615607e-001	-2.5730802e-001	-1.4620067e-002	-3.9049589e-002	2.2293673e-002
4.3516969e-001	9.3063294e-002	-2.3421764e-001	-4.6069245e-002	-3.4142496e-002	2.3497020e-002
3.6228205e-001	5.6627043e-003	-3.0907513e-001	-3.6015208e-002	-3.8310396e-002	2.5489186e-002
3.1231656e-001	-3.4694373e-002	-1.9672723e-001	-3.1576856e-002	-3.8826272e-002	2.5547094e-002
2.5327141e-001	4.2209420e-002	-1.9892307e-001	-3.9818674e-002	-2.6108481e-002	2.7224252e-002
1.4840643e-001	-2.3477968e-001	-3.5059254e-001	-1.9266304e-002	-3.6659069e-002	3.7279542e-002
3.5582884e-001	-1.7615438e-001	-4.9845596e-001	-4.0770459e-002	-5.1757577e-002	3.5988900e-002
3.0399728e-001	-2.1524354e-002	-2.7681626e-001	-5.9988479e-002	-3.2490323e-002	3.6371883e-002
1.4505097e-001	-1.7585167e-001	-2.3854276e-001	-1.2793647e-002	-5.7478148e-002	4.2587868e-002
2.8856070e-001	-4.7588479e-002	-4.5013630e-001	-6.0391790e-002	-3.5763369e-002	5.9650689e-002
1.1520735e-001	1.6134980e-001	-2.1085326e-001	-2.1553946e-002	-4.3854351e-002	-3.6858739e-003
-2.5192517e-001	7.5442562e-002	-2.8738423e-001	-5.6860832e-002	3.6479555e-002	1.1598538e-002
2.7199813e-001	3.8508751e-001	-3.8741853e-003	9.4415830e-003	-4.1560685e-002	-1.8137986e-002
-1.0888983e-001	1.4656658e-001	-1.7866892e-001	-1.4387729e-002	-3.2844757e-003	-9.4778180e-003
4.3410496e-001	-5.7031042e-002	-3.1462068e-001	-6.2702540e-002	-5.0806948e-002	1.1414186e-002
8.9392129e-002	-1.6572144e-002	-3.8261321e-001	-6.6921388e-002	1.0606696e-002	1.4001042e-002
2.0621151e-001	-9.9537204e-003	-3.2676691e-002	-1.9519481e-002	-2.1949003e-002	-4.1146300e-003
2.0419768e-001	-1.1838770e-001	-2.9806867e-001	-4.8130838e-002	-3.6000628e-002	1.2972872e-002
1.7709898e-001	2.6286778e-001	2.2590086e-003	1.9105632e-002	-1.4369288e-002	2.0408337e-002
2.3390253e-001	-3.3842149e-003	-2.6609590e-001	-6.5322656e-002	8.6341353e-004	1.5588331e-002
3.7735753e-001	-4.1674179e-003	-2.2267329e-001	-5.5019736e-002	-2.9370178e-002	2.6745791e-002
-1.3048922e-001	-1.4883804e-001	-2.2402338e-001	-9.2307460e-003	-5.5763831e-002	6.4330864e-003
-1.0755957e-001	1.6952772e-001	-1.6575343e-001	-4.8704393e-002	5.8959760e-002	-5.6700614e-003
8.6832995e-002	3.2346105e-001	1.1716746e-001	2.7262254e-002	-3.8409443e-002	-1.0980370e-002
-1.6234182e-001	3.1002639e-001	2.3494291e-002	3.4094815e-003	3.0559421e-002	-3.7086854e-002
1.3707443e-001	-2.4762747e-001	-5.5930095e-002	-1.3665697e-002	-5.8801576e-002	3.9083200e-002
1.1295760e-001	6.7259033e-002	-1.8173346e-001	-6.7207552e-002	1.2621537e-002	-1.3211372e-003
3.5488511e-001	2.0408166e-003	-1.5375300e-002	-1.8096466e-002	-3.6195311e-002	2.6240863e-003
3.0511213e-001	-1.2009512e-001	-2.2952727e-001	-3.3202950e-002	-3.2781924e-002	8.7934650e-003
4.3495558e-001	-1.5047384e-002	-1.4650346e-001	1.2594262e-003	-6.7796512e-002	4.8209154e-002
3.8304351e-001	3.1834668e-002	-2.5958563e-001	-5.5905251e-002	6.1854212e-003	1.9212441e-002
3.5321883e-001	-3.3608211e-002	-2.4092155e-001	-6.1604568e-002	-5.0900412e-003	3.3189282e-002
3.5798065e-001	5.1165959e-002	-3.6725442e-002	-3.1642827e-002	-2.7701161e-002	2.1094391e-002
2.5005281e-001	1.0674766e-001	-5.6519520e-002	3.2226686e-002	-2.6128471e-002	8.1847259e-003
7.4789606e-002	1.6178282e-002	-1.8247994e-001	-2.4570292e-002	5.1722469e-003	-1.8665231e-002
8.6489450e-002	6.9280749e-002	-1.5411682e-001	-2.0709736e-002	4.6344525e-003	1.0749026e-002
1.8651816e-001	-2.2237382e-001	-3.3306903e-001	-5.9193092e-002	9.0287546e-003	2.5464916e-002
6.9694520e-002	1.4210279e-001	-1.4132025e-001	-3.0826969e-004	2.7247167e-002	-4.6629301e-002