DEVELOPMENT OF A POSITION SENSITIVE DEVICE AND MULTI-POSITION ALIGNMENT CONTROL SYSTEM FOR AUTOMATED INDUSTRIAL ROBOT CALIBRATION

By

Erick Nieves-Rivera

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering - Doctor of Philosophy

2013

ABSTRACT

DEVELOPMENT OF A POSITION SENSITIVE DEVICE AND MULTI-POSITION ALIGNMENT CONTROL SYSTEM FOR AUTOMATED INDUSTRIAL ROBOT CALIBRATION

By

Erick Nieves-Rivera

This dissertation proposes a novel calibration system capable of automatically calibrating industrial robots. Often, inaccurate assumptions about real links parameters or even small offsets exist in the individual robot joints that lead to errors in the internal kinematic model equation and, as a consequence, affect the accuracy of a robotic system. To solve this problem, the proposed approach introduces a completely new technique for industrial robot calibration. The proposed system consists of an industrial robot manipulator, a camera, a laser fixture attached to the robot tool center point (TCP), a PC-based interface, and a new position sensitive calibration device (PSCD). This wireless calibration device is comprised of two fixed position sensitive detectors (PSDs) tilted with an angle between them to reflect the laser line from one PSD to the other. Such a device is capable of feeding back the movement information needed to localize the TCP frame relative to the device frame. The new calibration approach is not only able to compute the joint offset parameters of the robot but is also capable of simultaneously calibrating the robot's workpiece relationship. It was also designed to be faster, simpler and cheaper than any other methods. Throughout this dissertation, the newly developed calibration device, the principle of our calibration system and the control approach needed to achieve automation of the entire system are presented and discussed. Finally, the feasibility of the overall calibration system including device hardware, software and calibration algorithms was demonstrated with experimental results.

To my entire family who has always shown me love and support in every possible way.

ACKNOWLEDGMENTS

First and foremost, I would like to express my respect, appreciation and gratitude to my advisor Dr. Ning Xi for guiding this dissertation research from its inception to its conclusion. His guidance, feedback, and high expectations have shaped this dissertation research.

In addition, I want to thank my dissertation committee members: Dr. Hassan Khalil, Dr. Guoming Zhu, and Dr. Xiaobo Tan. I greatly appreciate their valuable feedback and insights throughout the whole dissertation process.

I would also like to thank Dr. Percy Pierre and Dr. Barbara O'Kelly for recruiting me to the Alfred P. Sloan program at Michigan State University and giving me the opportunity of a lifetime. Both of you took a chance on me and changed the rest of my life for the better.

Many others provided help and encouragement along the way. I wish to thank current and past Robotics and Automation Lab members: Yunyi Jia, and Dr. Yong Liu helped set the foundation for ideas that would evolve into this dissertation. Likewise, Jianguo Zhao, Hongzhi Chen, Ruiguo Yang, Bo Song, Chi Zhang, Liangliang Chen, Cheng Yu, John Gregory, Andres Ramirez, Jorge Cintron, and Nelson Sepulveda provided technical and motivational support during key stages of this dissertation.

My heartfelt thanks goes to my entire family, especially to my parents Virginio Nieves, Ramonita Rivera, and Saraí Berberena as well as to my brother and sister Elvin Nieves and Janise Nieves whose love, encouragement and support have been constant source of strength and inspiration.

Finally, I would like to thank my wife Sayra Reyes and Kyler Nieves for showing me love, understanding and support through this long, draining, and sometimes frustrating process. I would not have finished this dissertation without them, and I appreciate the sacrifices they made so I could focus and complete my research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

In the modern era, the complexity of industrialization has played an important role in developing a strong economy, typically related to technological innovation in manufacturing. Manufacturing, in general, involves the development of large-scale productions utilizing industrial robots to create assembly lines. Generally, industrial robots reach high repeatability levels, and, for repetitive applications, they are able to perform such tasks successfully. Repeatability demonstrates the quality of modern robots and their precise positioning capabilities. However, it is also well-known in the robot industry that industrial robots have high repeatability but low accuracy [1]. Nevertheless, the recent demand for high accuracy applications such as welding tasks, micro assembly operations, and surgery have increased the importance of and interests in robot calibration among researchers over the last few decades. Although there have been significant improvements in terms of accuracy of the newly designed industrial robot models, for such high accuracy applications the accuracy of the robot alone is not enough. While there are several sources of inaccuracies (e.g. thermal expansions, gear errors, structural deformations, or even incorrect knowledge of link and joint parameters), the main source of inaccuracy lies in kinematic model parameter errors. The majority of kinematic parameters (e.g. arm length, link offset, and link twist angles) are related to the structural mechanics of the manipulator. Typically, those parameters will not change by much once the robot is sent from the factory and installed in manufacturing areas. However, some kinematic parameters (e.g. joint offset) might be affected by the assembly or replacement of motors and encoders. According to [2] and [3], around 90% of the inaccuracy in robot positioning is due to errors on assumed initial joint values of the robot. Without an appropriate robot calibration, any robotic system will experience accuracy degradation over time. Because of this, robot calibration has been used to improve position and orientation accuracy of industrial robots by identifying inaccuracies in the kinematic model parameters in order to create a more accurate model that better fits the real robot.

Numerous robot kinematic calibration approaches and complete systems have been designed in industry as well as the academia with promising methodologies to calibrate the external parameters of industrial robots. Some of them collect accurate position data of the robot tool center point (TCP) by using highly precise equipment such as Computer Numerical Controlled (CNC) machines [4], Inclinometers [5], Theodolites [6], Coordinate Measurement Machines (CMMs) [7] and laser tracking systems [8], [9]. Other methods impose physical limitations on the TCP to form a closed kinematic chain. Such methods are required to fix one or more position and orientation constraints to the TCP. This allows the system to generate an equation capable of determining a set of parameters, also known as a self-calibration system. Due to this particular advantage, self-calibration systems are investigated and analyzed more widely than any other method. Furthermore, expensive measuring devices are not required. As discussed in [10] and [11], calibration is achieved by imposing plane constraints on the TCPs positions. In [12], the authors measured the position and poses of a robot by matching the pin of the TCP to an aperture on a dime. Related results can also be found in [13] where the authors used a single end point contact constraint equivalent to a ball joint. The robot then moved to different positions that satisfied such constraints. However, those methods are still problematic due to the need for external physical contact as well as the dependence on their individual manufacturing accuracies. Additionally, vision-based approaches and systems have been developed and used to perform the calibration process. However, those systems suffer from lack of resolution under wide fields of view, and low serving speed due to the low frame rate cameras possess [14], [15]. Because such devices are so expensive or their procedures are time consuming, they are difficult to use extensively in manufacturing plants. For instance, a Laser Tracker System can cost more than $100,000 US dollars. Therefore, it is particularly important to develop and design a method which is both cost-effective and easy to implement, while still being able to achieve a high level of position accuracy.

Among the existing robot calibration methods, our optical approach using Position Sensitive Detectors (PSDs) is one of the best choices since it promises high precision, fast response, and low computational load. A line-based method to calibrate the robot's external parameters [16] as well as an approach to calibrate the robot's joint offsets [17] were proposed. Both methods were developed in our lab. Such methods mainly depend on a PSD device and a single laser pointer which is attached to the TCP of the robot. Based on recorded joint angles and available forward kinematic information of the robot, the system is able to calibrate the robot. Both simulation and experimental results verified the feasibility of these proposed methods and demonstrated that the developed systems could perform robot calibrations not only during assembly but also in user environments. However, these methods are still time consuming and have not been yet integrated, therefore they must be performed separately to achieve both workpiece frame and joint offset calibrations. Moreover, two different PSD devices for each method were needed, one of which was not portable. Furthermore, it has been proven that results depend heavily on robot configurations, and at least three PSDs and two directions of laser beams are needed to calibrate the external parameters of the robot.

This dissertation presents a new dual PSD calibration system for industrial robot calibration. This system can be used not only to calibrate the workpiece frame relationship but also the joint offsets of the robot simultaneously, thus improving speed during the whole calibration process. The design of a new portable position sensitive calibration device (PSCD) is presented. This developed device only requires two PSDs to perform both calibration methods. The method developed also employs a focusable laser pointer attached to the TCP. During the calibration task, the beam from the laser pointer is aimed at the center of the first PSD, i.e. PSD1, with a reflection orientation towards the second PSD, i.e. PSD2. Once these two centers are found, a line that passes between them can also be found. The reflected line is a virtual linear constraint, meaning that the TCP and the laser beam are constrained to move along that reflected line. The PSCD is located in an unknown position with respect to the robot. During the whole calibration process, the procedure of aiming the laser beam at the center of each PSD only repeats twice, so the approach is simpler and less time consuming previous methods. Once the aiming process is completed, the joint angles for that particular pose of the robot are recorded. Based on this recorded data along with the kinematic model of the robot, a calibration algorithm is employed. Mathematically, the proposed method uses a non-linear iterative optimization technique to identify the robot parameters from the data recorded. Using different positions, poses and orientations, the errors are minimized. As a final step for the proposed calibration system, after all parameter errors are found, the robotic system is then compensated internally to make the system more precise and accurate. Both simulations and experiments were implemented on an ABB industrial robot (IRB120) to verify the efficiency of the proposed method as well as the feasibility of the newly developed calibration system.

1.1 Problem Description and Motivation

Robot calibration is the process of enhancing or improving the accuracy of a robotic system by means of modifying the robot's model and control software. In general, a calibration system consists of four different but important steps;

1. *Kinematic Modeling*: First, a mathematical model describing the robot's motion and geometry have to be determined.

2. *Pose Localization*: At this point, an accurate measurement of the position and orientation of the TCP have to be performed in world coordinates.

3. *Relationship Identification*: In this step, the relationships between individual joint angles as well as individual user-defined coordinate frames are found.

4. *Compensation*: Finally, the kinematic model of the robotic system is compensated through the internal control software making the overall system more accurate and ready to perform extremely delicate tasks.

The pose localization step is the most crucial phase toward a successful robot calibration. However, it is at this point where most state of the art systems fail or are far from being user friendly calibration tools.

The topic of robot calibration has become a research field of great importance over the last decades, especially in the field of industrial robotics. The main reason for this is that the field of application was significantly broadened due to an increasing number of fully automated or robot assisted tasks to be performed. Those applications require a significantly higher level of accuracy due to more delicate tasks that need to be fulfilled (e.g. assembly in the semiconductor industry or robot assisted medical surgery). In the past, (industrial) robot calibration had to be performed manually for every single robot under lab conditions in a long and cost-intensive process. Expensive and complex measurement systems had to be operated by highly trained personnel. The result of this process is a set of measurements representing the robot pose in the task space (i.e. world coordinate system) and as joint encoder values. To determine the deviation, the robot pose indicated by the internal joint encoder values has to be compared to the physical pose (i.e. external measurement data). Hence, the errors in the kinematic model of the robot can be computed and therefore compensated. These errors are inevitable and due to varying manufacturing tolerances and other sources of error (e.g. friction and deflection). They have to be compensated in order to achieve sufficient accuracy for the given tasks. Furthermore, for performance, maintenance, or quality assurance reasons, the robots may have to undergo the calibration process in constant time intervals to monitor and compensate aging effects such as wear and tear. In modern production processes, old fashioned procedures like the one mentioned above are no longer suitable. Therefore, a new method has to be found that is less time consuming, more cost effective, and involves less or even no human interaction in the calibration process.

1.2 Calibration Task

There are many traditional calibration systems out there, from the old fashioned but reliable systems that uses physical contact, to the most state of the art systems. Figure 1.1 shows an example (image taken from [18]) of the classical calibration device as still in use today. The robot arm has to be moved manually into a position that allows the metal pin attached to the TCP to touch a second metal pin fixed in the robots task space. This has to be performed multiple times using different joint configurations to achieve the required precision. This method is both time consuming and involves a lot of manual interaction by a trained operator. Other approaches are based on non-physical contact like the ones described in [19] and [20]. These methods used a laser line as a virtual linear constraint on the TCP. Gatla et al. [21] proposed a method of virtual closed kinematic chain. In this method, a laser beam aims at a constant but unknown location on a fixed distant object, creating a virtual closed kinematic chain. However, the point of intersection of the TCP position with the line is judged by the operator.

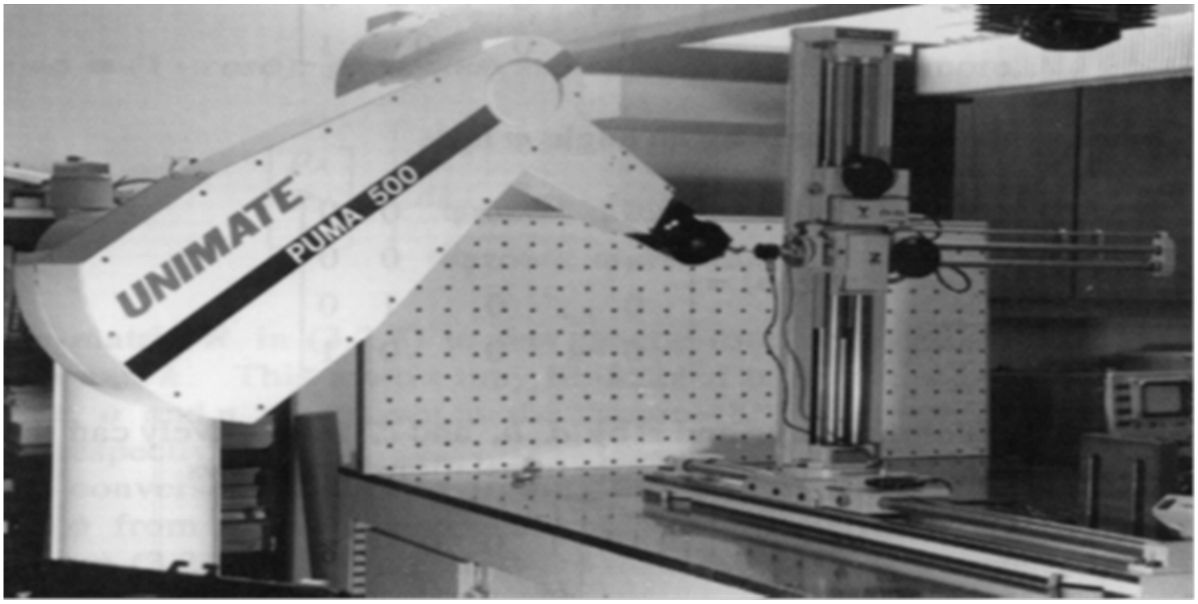


Figure 1.1: Example of the classical calibration system.

Among the most popular state of the art calibration systems is the FARO laser tracker, shown in Figure 1.2 below. Such a device allows to perform robot calibration accurately and is one of the most reliable systems out there able to overcome most of the problems we are currently trying to solve. However, as we mentioned before, we would like to take the calibration task to the next level, creating a simpler system, cost-effective, and more accurate than any other system yet created before, including our own previews system designs.

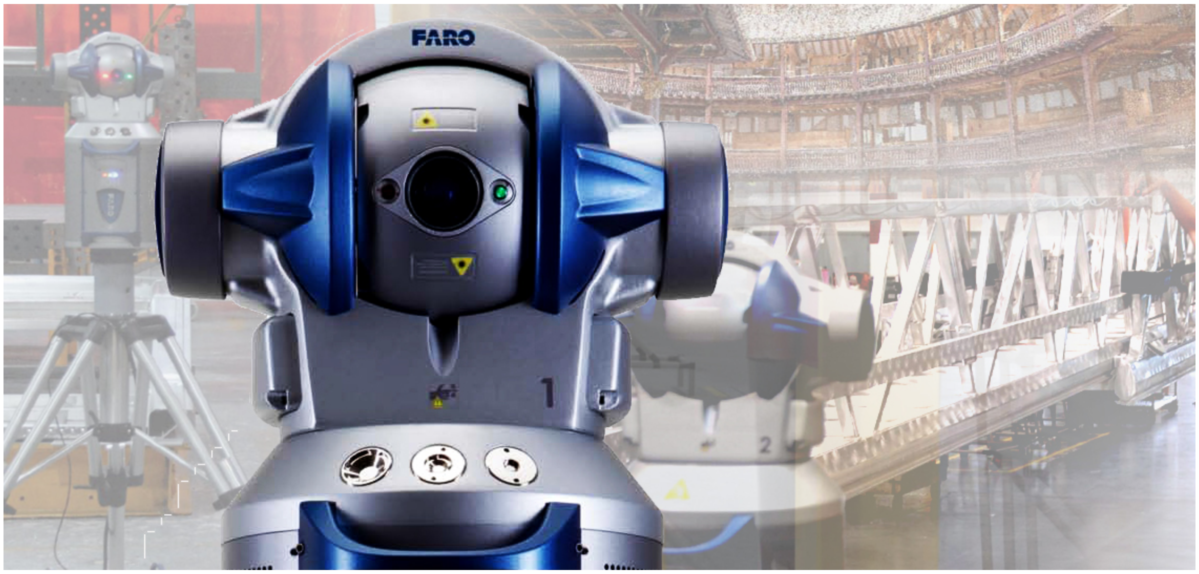


Figure 1.2: FARO laser tracker. Note: For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.

A single PSD device was developed in the past in our labs to overcome those problems, and was successfully implemented, this system is shown in Figure 1.3 below. However, it still lacks speed and the accuracy is not sufficient. Such a system consists of a single PSD mounted on a portable device. The device is arbitrarily located on the robot workspace. The center point of the PSD is supposed to be the single-point constraint. The interface circuit was well designed and the signal tuning board can process the raw output of the laser spot on the PSD surface for two-dimension position feedback. A PC-based controller collects PSD output data without the need of any cables attached, PSD-based positioning servo and calibration algorithms are applied to complete the calibration process. Through the network-based communication between the robot controller and the PC-based controller, the latter can obtain the current robot position information (task space and joint space) from the robot controller and send the control command to the robot controller as well as update the target position in real-time.

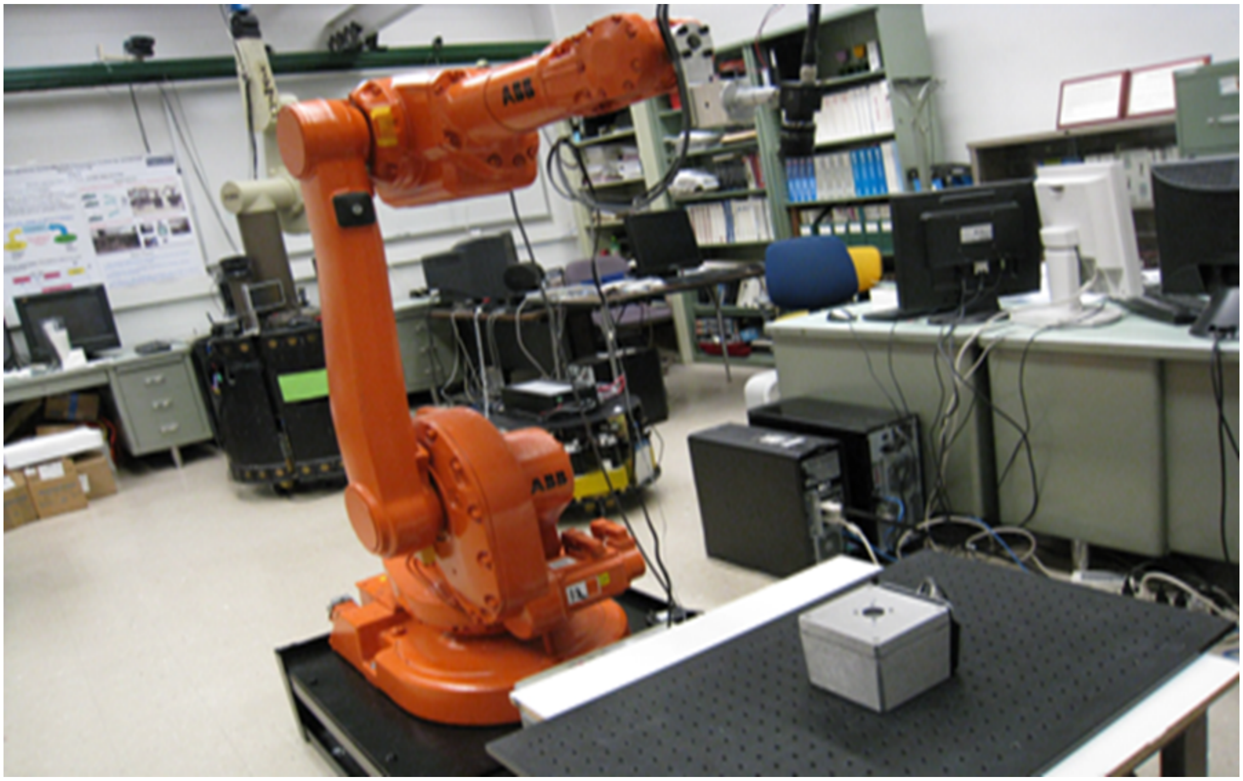


Figure 1.3: Single PSD-based calibration system

In this dissertation an optical approach based on two PSDs (i.e. Position Sensitive Detectors) was chosen to improve the calibration process in respect to time, reliability, and cost aspects that other methods still lack. Another important facet of improvement is to reduce or even eliminate the need for human participation in the process. This is due to the fact that it is both expensive and error-prone. As shown in [2] industrial robots in general feature good repeatability but rather poor accuracy. This is derived from the fact that the major share of error contribution is of non-dynamic nature, such as the joint zero position error. These static errors could be avoided by initially calibrating the robot. For this purpose a system has to be designed and manufactured that allows it to determine the robot's individual positioning inaccuracy.

The idea of the optical system proposed is to obtain a robot specific representation such as joint encoder values for a given set of points in the robots task space. Since these points are already given in an external representation, i.e. world coordinate system (WCS), the robot base frame can be related to the task space, i.e. world frame. Hence, a higher positional accuracy can be achieved using this method. This is based on the fact that the desired pose in the task space can be directly transferred into joint variables in the robots kinematic model. Once this task is performed, the relationship between the robot base and the task space can be established and hence, the calibration operation is performed successfully. Further details of the proposed calibration system will be discussed in Chapter 3.

There are two different main ways to calibrate an industrial robot, both equally useful in the calibration task;

1. Joint Offset Calibration

2. Robot Workpiece Frame Calibration

Ideally one would like to have both methods performed at the same time in order to improve reliability on the calibration system, however the later in not always possible, which is the case of our previous method presented in Figure 1.3.

1.2.1 Joint Offset Calibration

Joint Offset Calibration is the process of calculating the individual variations or error contribution of each robot's joint, so that they can be compensated later in the controller internal kinematic model. Figure 1.4 shows a schematic of a classical joint offset calibration where the values of δ are unknown. For this particular schematic, the offsets in joint one and two are computed and inserted in the original model, so that the transformation matrix between the base frame and the TCP frame is more accurate than before.

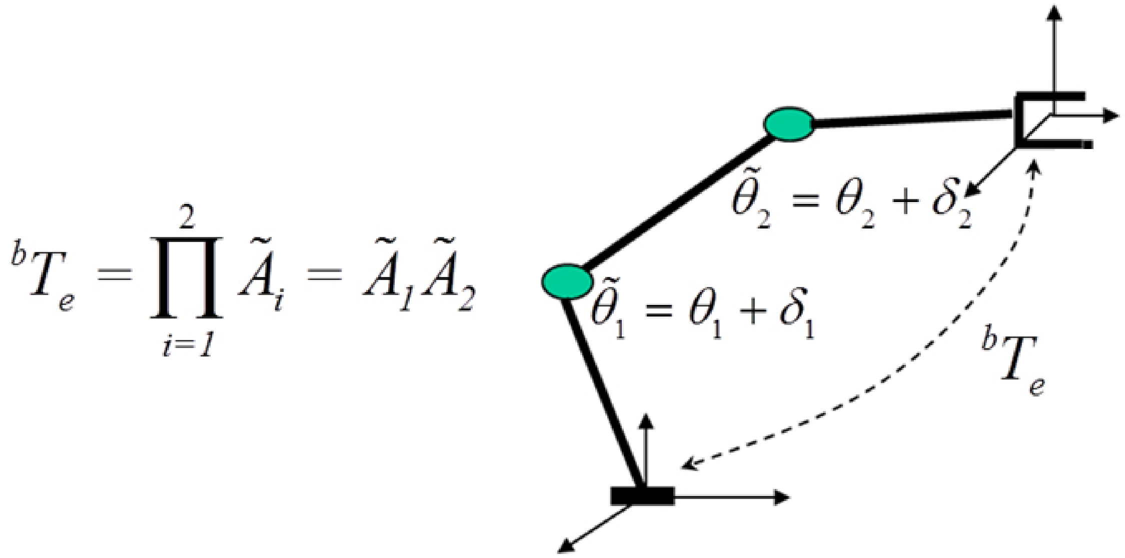

${}^{b}T_{e} = \prod_{i=1}^{2} \tilde{A}_{i} = \tilde{A}_{1}\tilde{A}_{2}$
$\tilde{\theta}_2 = \theta_2 + \delta_2$
$\tilde{\theta}_1 = \theta_1 + \delta_1$
${}^{b}T_{e}$

Figure 1.4: Schematic of a traditional joint offset calibration

1.2.2 Robot Workpiece Frame Calibration

Robot Workpiece Frame Calibration is the process of calculating the relationship between the robot base frame and the robot workpiece frame usually in the form of a transformation matrix, so that the entire kinematic model can be compensated later. Figure 1.5 shows a schematic of a classical workpiece frame calibration where the value of the matrix ${}^{w}T_{b}$ is unknown. For this particular schematic, the unknown matrix is computed and inserted on the model so that the transformation matrix between the base frame and the TCP frame is more accurate than before.

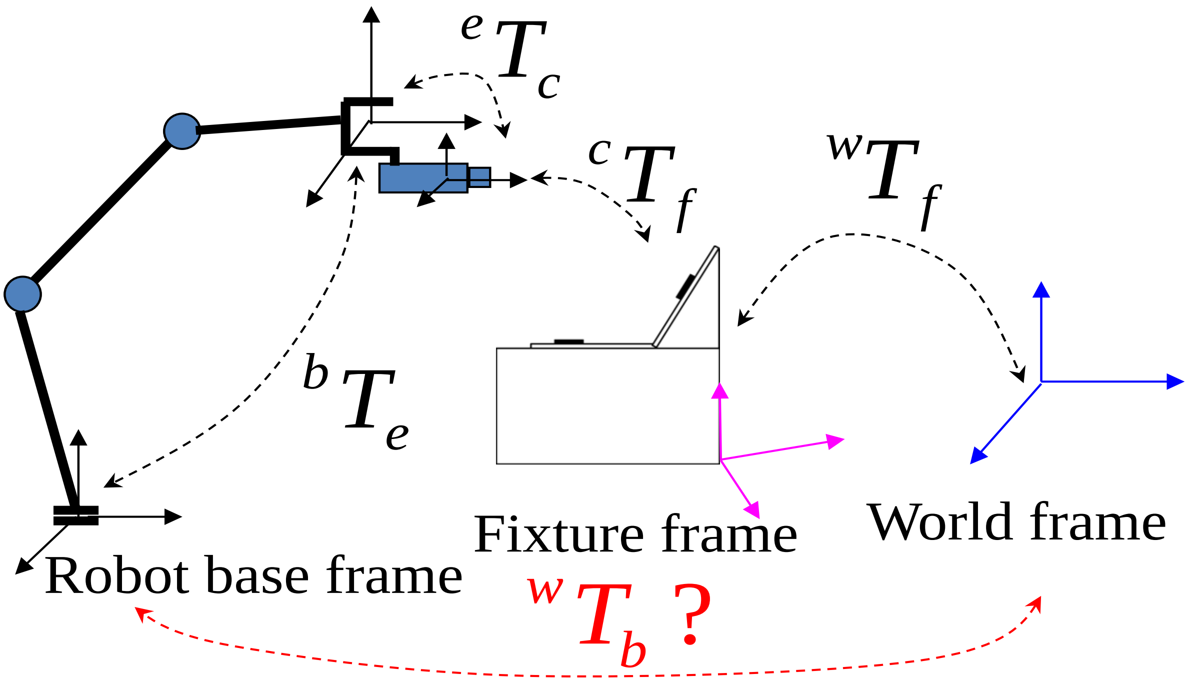


Figure 1.5: Schematic of a traditional robot workpiece frame calibration

1.3 Thesis Statement

This research is intended to present and describe our newly developed calibration system for industrial robots. In particular, this system or tool uses a combination position sensitive detectors and laser technology to accurately achieve the calibration task for industrial robots.

Thesis Statement: *By designing the most accurate, fastest, portable, reliable and cost-effective system for robot calibration, we intended to revolutionize the way traditional calibration systems performs now a days.*

1.4 Anticipated Research Contributions

This section is intended to give the reader an idea about the anticipated research contributions and to eliminate any misconceptions. First of all this dissertation only covers the work of the first phase of this research. It is particularly important to understand that the presented system does not at all claim to be a finished product ready for use. In fact, it is intended to be a technical feasibility evaluation performed by Michigan State University Robotics and Automation Laboratory and funded by Asea Brown Boveri (ABB). The long term motivation for this is to be able to replace calibration technologies used today that has some undesirable side effects (discussed in Chapter 2). In particular, an alternative to old fashioned systems as well as other state of the art systems is to be found. The expectations towards this first stage of the dissertation are to design and manufacture a system consisting of PSDs (calibration fixture) and laser devices (laser fixture) and show that calibration can be performed using this method with particular advantages over any other calibration system.

Anticipated Advantages:

- A simple low-cost design.
- Robust to external disturbances commonly found in production lines.
- Easy to operate by the user.
- Design of a convenient wireless portable calibration device.
- Ability to be used by any industrial robot manipulator.
- The fastest automatic robot calibration system yet designed.

- Ability to perform both, joint offset calibration and workpiece frame calibration at the same time.

In order to achieve those goals, first a data acquisition system must be designed and implemented. Next, a PC-based controller strategy has to be designed, tested and implemented. Then, a calibration system has to be designed in order to identify the kinematic parameters accurately. Lastly, further experiments have to be performed in order to prove the system was able to overcome the above anticipated advantages.

1.5 Dissertation Outline

The remainder of this dissertation is organized as follows. In Chapter 2 we review background material on robot calibration systems, previous work on robot calibration, camera calibration techniques, and some other related work. Although closely related work is presented in each of the subsequent chapters, Chapter 2 overviews high-level related work on calibration approaches. Chapter 3 provides an overview of the calibration system as well as the hardware needed to complete the calibration process. Next, Chapter 4 presents the data acquisition system needed to provide feedback to the controller. This chapter explains in more details the hardware used, the working principle of the position sensitive calibration device and its internal components. Experimental performance results are also illustrated, analyzed and discussed at the end of Chapter 4. Chapter 5 proceeds to present detailed derivations on the control design used for our calibration system. Also in Chapter 5, controller simulations and experiments are presented and discussed. Chapter 6 however, focuses on the analytical and computational aspect of the actual calibration algorithms used to find our parameter errors. Also this chapter presents the analysis of the kinematics error model, and simulation results. Chapter 7 presents the experimental results of the calibration system, including a comparison between our previous approach (Single PSD Calibration System) and the proposed calibration system. Lastly, Chapter 8 presents conclusions, summarizes the contributions, and discusses the remaining investigations for this dissertation.

Chapter 2

Background

This chapter provides selected literature review and background information on the topic of robot calibration, which is fundamental to this dissertation. First, a brief historical review is provided. Then, related work from other areas of robot calibration is discussed. Next, we overview previous work on robot calibration, specifically those related to this dissertation. Is at this point where camera calibration techniques as well as previous work performed in our labs including the single PSD calibration approach is briefly presented and discussed. During this chapter, the advantages and limitations experimental results suggest about previous work on robot calibration will also be presented and discussed, specifically those fundamentally related to this dissertation.

2.1 Historical Background

The exponential growth in robotics research registered around the late 1970's was a result of successfully applying robotic systems to assembly lines in the manufacturing industry. Specifically, the automotive industry was the one with the most benefits. At that time the computer industry was also rapidly growing making computers more accessible than ever. However, integration between robotic systems and programmable computers did not come until the late 1980's. Therefore, the predominant method for programming robotic systems at that time was by physically moving the manipulator's TCP to each task point and recording the encoder information for each joint, so that they can be replayed later on. Likewise at that time, robotic systems were designed with low accuracy but high repeatability since this last one was the most important quality for most applications.

In the 1980's, numerous publications emerged from researchers working on robotic systems. In [22] the author has developed a succession of interesting ideas concerning representation, specifically the use of homogeneous matrices. His Denavit-Hartenberg (D-H) kinematic modeling formulation for robot path planning and control as well as his systematic use of homogeneous transformations, quickly became standards for researchers. As a result, manipulator controllers were designed and built using D-H model conventions. The growth of task space robot path planning and the introduction of sophisticated sensors such as force sensors, cameras, proximity sensors and the personal computer revolution, fueled the idea that robot manipulators will soon be able to implement fully automated factories, as well as to be key elements in many sophisticated systems involving the online interaction of large amounts of sensory data. Theoretically, those applications will require repeated use of the robot inverse kinematic model. Moreover, they introduce the need to program the robot off-line to align the TCP to the new task point without previous knowledge of such task point. Is at this point that the new problem of robot accuracy starts to arise. The accuracy of a robotic system highly depends on how accurate the robot kinematic model embedded on its own controller software is. Therefore, researchers began to study the effects of geometrical parameter errors, joint axis misalignments, joint offsets, and other possible sources of error in the robot's TCP positioning.

In 1983 a major discovery was found in [23] stating that the D-H model was singular for robots with parallel joint axes. Thus, they introduced a modification to the so called D-H model gaining popularity among researchers and eventually was adopted by many. Since most industrial robots are designed to be simple and user friendly, i.e., to have perpendicular and parallel joint axes, the singularity problem represents a major issue among researchers.

During the mid 1980's robot calibration was born as a new research area in order to improve the accuracy on robotic systems. To improve the accuracy of a robot it is necessary to measure the world coordinates of the robot's TCP at different joint space configurations or robot poses and record each joint value at those configurations. From a practical point of view, the robot task point at each configuration computed by the nominal robot's kinematic model will always differ from the actual TCP absolute position and orientation measurements, it all depends on what level of accuracy you are looking for. Therefore, if those two measurements differed, the nominal robot's kinematic model needs to be suitably modified. As more accurate demands for robots in the manufacturing and surgical industries arises, the need of robot calibration becomes a crucial and absolutely necessary element to enhance robot accuracy. In other words, the more demands for higher level of accuracy, the more important robot calibration became.

Many robot calibration studies during the late 1980's were done by numerous researchers.

In particular, the paper presented in [24] presents a systematic method to compute the identification jacobian, which is the matrix relating the TCP pose errors with the robot kinematic parameter errors. In 1988 two techniques for accuracy compensation were described also in [25]. As robot calibration research expanded, a four step problem was featured in [26] consisting of modeling, measurement, identification and compensation. In the early 1890's that work was expanded into a full scope book [1]. The book features one of the first tutorials to the fundamental concepts and methodologies involved in robot calibration.

During the 1990's many more robot calibration studies were done by researchers, however some key problems and questions still persisted that even today are not fully solved and answered, such as:

- How an optimal robot measurement configuration can be chosen?
- Is observability of kinematic errors related to the selection of configurations?
- Should robotic systems be designed differently to accommodate online calibration capabilities?
- Is it possible to perform robot calibration cheaper and yet faster?
- What current technology can be used to develop a calibration system economically feasible?
- Can the entire process of robot calibration be entirely automated?
- Can such a system be easy to operate by the user?

In 2006 a book was released ([27]) documenting and addressing some of these research issues using a robot calibration method aided by cameras, in fact most of our research is inspired by that book where most of the work was done in the late 1990's. In 2009 we began to investigate and develop our new calibration system which essentially answers the rest of those issues and questions.

2.2 Related Work

From a data collection point of view robot performance evaluation and robot calibration are similar research areas. Robot performance evaluation assess repeatability as well as accuracy of robotic systems. Many researchers focused on evaluation of machine tools and robot manipulators at the National Bureau of Standards. An excellent survey about robot end-joint sensing techniques can be found in [28]. In fact, one of the first major reports of actual robot calibration experiments was the paper in [29]. Data collection was performed by them using theodolites.

Many robot testing studies were performed by a wide range of measurement techniques, from expensive Coordinate Measuring Machines (CMM) and tracking interferometer systems to the other employing customized fixtures. The main idea was the measurement in world coordinates of one point on the robot TCP. Often those coordinates are defined based on the calibration equipment. Measured point represents the TCP position and the measurement of more than three coordinate points provides both position and orientation of the TCP.

On the other hand, camera calibration is a key element related to this dissertation. In camera calibration the biggest issue is the difficult tradeoff between camera resolution and camera field of view. To solve this problem the camera should always remain close to the moving robot target. Therefore, it is necessary to move the cameras together with the robot TCP. Experimental results were reported in [30] showing and demonstrating that robot calibration using moving cameras was feasible. The main idea is that the camera attached to the robot TCP will continue to be calibrated with respect to a fictitious camera calibration fixture that moves together with the moving robot avoiding the problem of calibrating the actual camera before each calibration procedure was performed.

2.3 Review of Previous Work

The following section will be focused on a review of the previous work on robot calibration that is specifically relevant to this dissertation. First, we present a review camera calibration techniques followed by our own previous work on robot calibration using a single position sensitive detector (PSD). The intent of this section however is to focus on those techniques that have practical relevance to the problem of robot calibration and to this dissertation, rather than provides a comprehensive review of this rich area.

2.3.1 Camera Calibration Techniques

There are many different camera calibration models that can be classified into linear or nonlinear models. For the purpose of this dissertation an ideal distortion free camera model is presented to provide the reader with the basic concepts. The purpose of this model is to relate the coordinates of an object visible by the camera, to the coordinates of such object in a reference coordinate system. In this dissertation this object image is the laser beam spot visible on the surface of a calibration device, creating a point in the image frame. Let $\{x_w, y_w, z_w\}$ be the world coordinates, and $\{x, y, z\}$ denote the camera coordinates, whose origin is at the optical center point O. Here z axis coincides with the optical axis, and $\{X, Y\}$ denote the image coordinates with center at O_I, i.e. the intersection of the optical axis z and the image plane. As shown in Figure 2.1, $\{X, Y\}$ lies on a plane parallel to the x and y axes. The relationship between the world coordinates $\{x_w, y_w, z_w\}$ and the camera coordinates $\{x, y, z\}$ is given by,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + t \tag{2.1}$$

where the rotation matrix R and translation vector t are denoted as

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \tag{2.2}$$

and

$$t = [\ t_x \quad t_y \quad t_z\ ]^T \tag{2.3}$$

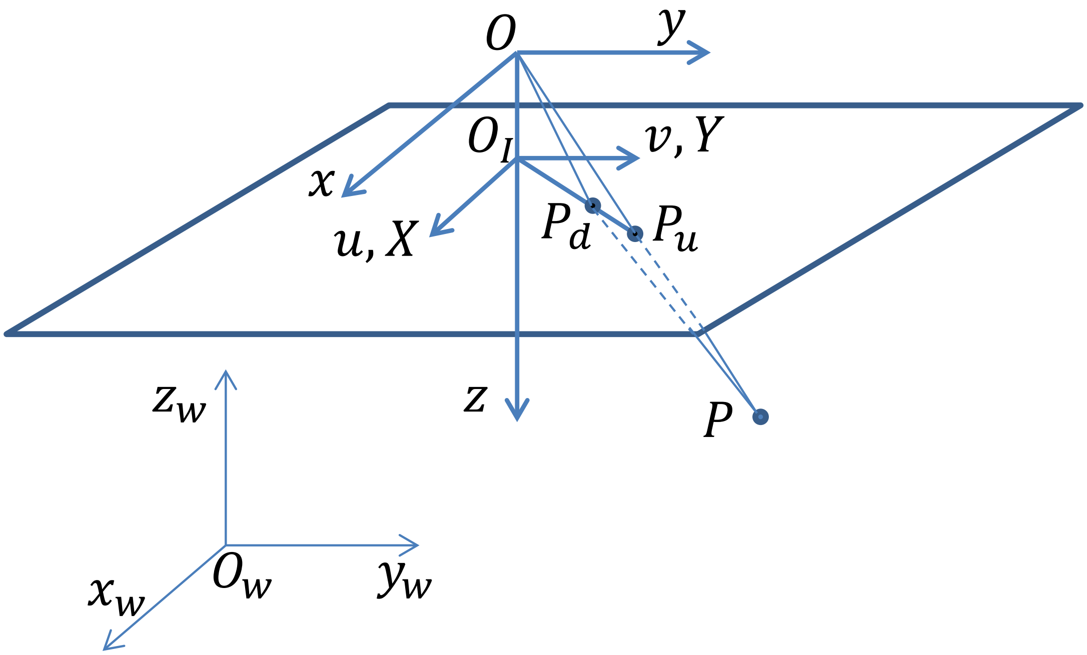


Figure 2.1: Camera coordinate system

The main idea of this model is that every object point is connected to its corresponding image point in a straight line through the focal point of the camera lens. Therefore, the following equation holds:

$$u = f\frac{x}{z} \tag{2.4}$$

$$v = f\frac{y}{z} \tag{2.5}$$

here f is the focal length of the camera and (u, v) are the coordinates of the object point in the image plane. Also the image coordinates (X, Y) are related to (u, v) as follows,

$$X = s_u u \tag{2.6}$$

$$Y = s_v v \tag{2.7}$$

where s_u and s_v are scale factors from the camera coordinates (u, v), measured in mm, to the image coordinates (X, Y) measured in pixels. f, s_u and s_v are the intrinsic parameters that carry the internal information about the camera components and the interface of the camera to the vision system. Because there are two independent parameters in the set of intrinsic parameters, lets define

$$f_x \equiv f s_u \tag{2.8}$$

$$f_y \equiv f s_v \tag{2.9}$$

Combining equations (2.8) and (2.9) with equation (2.1), we have

$$X = f_x \frac{r_1 x_w + r_2 y_w + r_3 z_w + t_x}{r_7 x_w + r_8 y_w + r_9 z_w + t_z} \tag{2.10}$$

$$Y = f_y \frac{r_4 x_w + r_5 y_w + r_6 z_w + t_y}{r_7 x_w + r_8 y_w + r_9 z_w + t_z} \tag{2.11}$$

Note that this model relates the world coordinate system $\{x_w, y_w, z_w\}$ to the image coordinate system (X, Y). Usually the image coordinates stored in the computer memory of the vision system are not equal to the image coordinates (X, Y). Thus, let (X_f, Y_f) be the computed image coordinates an arbitrarily selected point, and (C_f, C_f) be the computed center O_I of the image coordinates in the image plane. Thus the following equation can be derived,

$$X = X_f - C_x \tag{2.12}$$

$$Y = Y_f - C_y \tag{2.13}$$

where the ideal values of C_x and C_y can be obtain by the image size parameters of the vision system.

The camera parameters to be calibrated from the camera model defined by equations (2.10) and (2.11) are the independent extrinsic parameters of R and t, and the intrinsic parameters of f_x, f_y, C_x and C_y. The camera calibration is done by taking a set of n points which have world coordinates $\{x_{w,i}, y_{w,i}, z_{w,i}\}$, $i = 1, 2, ..., n$ which are known, and are within the field of view of the camera. Those points can be detected on the camera image at their respective image coordinates (X_i, Y_i). The camera calibration problem is to identify the unknown coefficients of the camera model given the above known data. Identification of the parameters in (2.10) and (2.11) will provide the pose of the camera in world coordinates.

One of the most basic camera calibration method consist of linear least squares identification of the transformation matrix. Using this method, the model in (2.10) and (2.11) can be rewritten as

$$X = \frac{a_{11}x_w + a_{12}y_w + a_{13}z_w + a_{14}}{a_{31}x_w + a_{32}y_w + a_{33}z_w + a_{34}} \tag{2.14}$$

$$Y = \frac{a_{21}x_w + a_{22}y_w + a_{23}z_w + a_{24}}{a_{31}x_w + a_{32}y_w + a_{33}z_w + a_{34}} \tag{2.15}$$

Here we can set $a_{34} = 1$ since the scaling of the coefficients $a_{11}, ..., a_{34}$ does not change the values of X nor Y. Equations (2.14) and (2.15) can now be combine into the model as follows:

$$\begin{bmatrix} x_w & y_w & z_w & 1 & 0 & 0 & 0 & 0 & -Xx_w & -Xy_w & -Xz_w \\ 0 & 0 & 0 & 0 & x_w & y_w & z_w & 1 & -Yx_w & -Yy_w & -Yz_w \end{bmatrix} \begin{bmatrix} a_{11} \\ \vdots \\ a_{33} \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix} \tag{2.16}$$

The minimum number of calibration points is six to be able to obtain the unknown coefficients $a_{11}, ..., a_{33}$ using linear least squares since each data point pair $\{(x_{w,i}, y_{w,i}, z_{w,i}), (X_i, Y_i)\}$ contributes two equations for the unknown vector $[a_{11}, ..., a_{33}]^T$ in (2.16). However is important to note that such points cannot be all coplanar. For instance, consider the case where $z_{w,i} = C$ and $C = Constant$, then the left hand side matrix will consist of columns 3 and 4 as well as 7 and 8 being linearly dependent and therefore singular. Moreover, the solution in equation (2.16) is not a global optimal solution of the camera calibration problem since it was derived from a distortion free model.

However using the same distortion free model, if we consider the use of two cameras to obtain two sets of coefficient vectors, then is possible to compute the unknown coordinates $\{x_w, y_w, z_w\}$ of any point present in both fields of view using the point's image coordinates. Then for the unknown point, the corresponding image points are $\{X^A, Y^A\}$ and $\{X^B, Y^B\}$, from here we can construct the following equation,

$$\begin{bmatrix} a_{11} - a_{31}X & a_{12} - a_{32}X & a_{13} - a_{33}X \\ a_{21} - a_{31}Y & a_{22} - a_{32}Y & a_{23} - a_{33}Y \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} X - a_{14} \\ Y - a_{24} \end{bmatrix} \tag{2.17}$$

Two pairs of equations from camera A and camera B provide a 3D measurement of a point from its measured image coordinates. In terms of this dissertation this fact was important and inspired the use of two sensors instead of one to be able to reduce the computational load and increase speed in the calibration process among other advantages. Full details of these advantages will become evident in Chapter 6 and Chapter 7.

2.3.2 Single PSD Calibration System

A parameter calibration approach called virtual lines-based single-point constraint (VLB-SPC) was proposed and implemented in our labs to overcome most calibration problems. A comprehensive and exhaustive review of this calibration system can be found in [17] and [31]. Unlike previous calibration methods, this approach does not need any physical contact and the developed device is affordable and automated. The proposed method depends mainly on a laser pointer attached on the tool center point (TCP) of a robot and only one position-sensitive detector (PSD). A schematic of this developed system is depicted in Figure 2.2. The coordinates of the PSD on the workcell are unknown. The automated calibration procedure involves aiming the laser mounted on the robot towards the center of the PSD surface from various robot positions and orientations. Once the precise positioning is done by PSD-based servo, all the laser lines will shoot on the same point with a very small range of error and a set of robot joint angles will be recorded. Based on the recorded joint angles and forward kinematics, a joint angle offset estimation method has been developed. Evidently, if offset values of all joints are zero, the intersections of every laser-line pair computed from the recorded joint angle and forward kinematics are the same point. However, if offset values of all joints are not zero, the intersections of every laser-line pair will be different points. In other words, the distribution of the intersections depends on the robot offset. An optimization model and algorithm was also formulated to identify the robot constant offset.

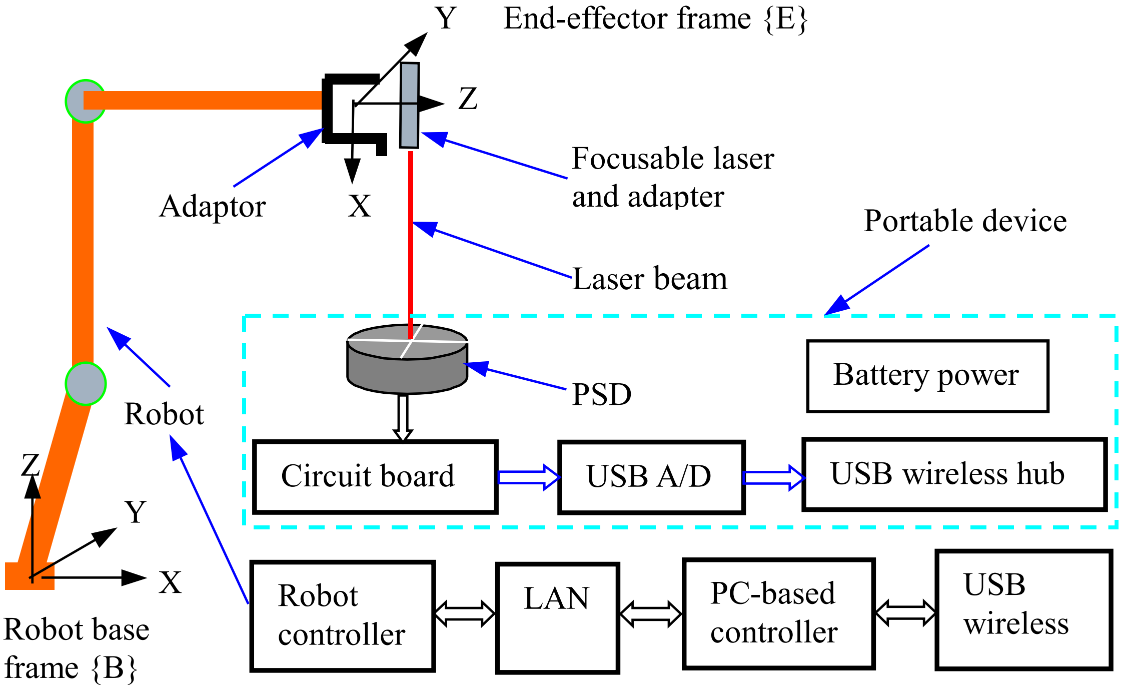


Figure 2.2: Schematic of the single PSD calibration system

For the purpose of precise positioning on the same point, the segmented PSD was also employed for high precision feedback with a resolution of better than 0.1 μm and a PSD-based controller was designed and implemented. The center point of the PSD will function as the single-point constraint. The interface circuit was well constructed and the signal tuning board was able to process the raw output of the laser spot on the PSD surface for two-dimension position feedback.

A new parameter calibration approach called VLBSPC, was developed to calibrate the joint offsets. The calibration procedure, as shown in Figure 2.3, is performed by pointing the laser beam at the same point from the various positions and orientations. This point is the center point of the PSD and the coordinates of the point in the robot base frame are unknown. It is guaranteed that the laser beams shoot on the same point because the robot aims the laser at the center of the PSD through PSD-based feedback and servo.

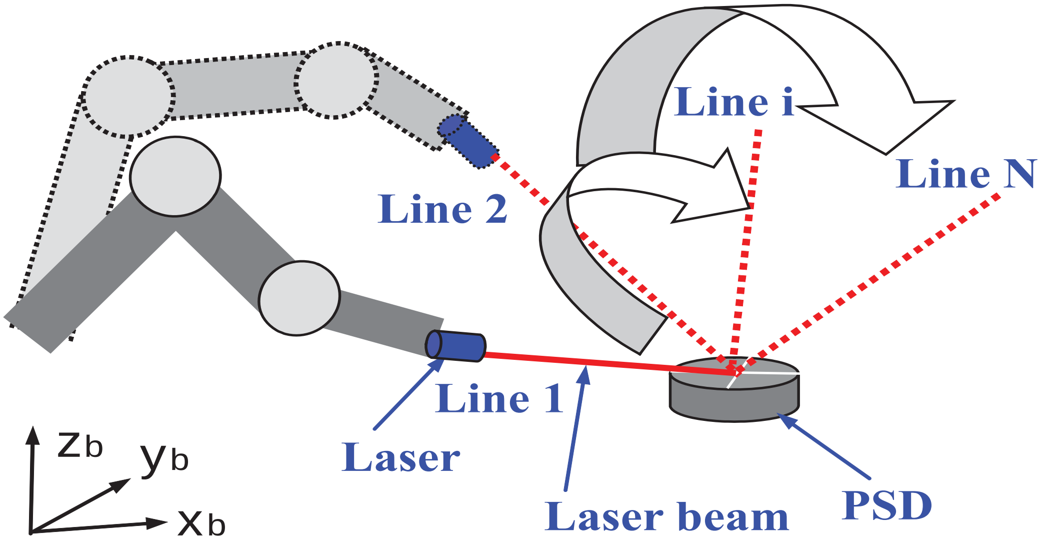


Figure 2.3: single PSD calibration method

Sets of robot joint angles are recorded during the localization. Substituting the recorded joint angle into the forward kinematics with offset error, the homogeneous transformations of end-effector fame with regard to the robot base frame are given by

$$\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.18}$$

Note that the unknown parameters are the joint offsets. Therefore, Combining the tool parameters and the forward kinematics with offset error, one of the laser lines translated from the TCP frame to robot base frame is described by

$$\frac{x_B - x_{iB}}{m_{iB}} = \frac{y_B - y_{iB}}{n_{iB}} = \frac{z_B - z_{iB}}{p_{iB}} \tag{2.19}$$

where (x_{iB}, y_{iB}, z_{iB}) are the coordinates of one point of the laser line in the robot base fame and (m_{iB}, n_{iB}, p_{iB}) the unit vector of the laser line direction in the robot base fame.

Suppose N sets of joint angle are recorded after calibration. From equation 2.19 N laser lines are obtained. Let Γ_{Li} denote the ith laser line, P_k denote the intersection or the center of the shortest distance between Γ_{Li} and Γ_{Lj} $(i \neq j, i, j \in N, k \in M)$, and ${}^{n}\mathrm{P}_{Ave}$ denote the mean point of the total intersections P_k $(k = 1, \cdots, M)$. The coordinate errors of the points between P_k and ${}^{n}\mathrm{P}_{Ave}$ are denoted as ${}^{x}\Psi_k, {}^{y}\Psi_k, {}^{z}\Psi_k$ in the x, y, z directions, respectively. The parameters δ of joint offset are identified by minimizing the total sum of the squares of the coordinate errors.

$$\delta^* = \arg\min_{\delta} \left({}^{x}\Psi_k{}^2 + {}^{y}\Psi_k{}^2 + {}^{z}\Psi_k{}^2 \right) \tag{2.20}$$

where M is the number of the intersections between laser lines. Note ${}^{n}\mathrm{P}_{Ave}$ is updated during the minimization iteration process and P_k is the center of the line of the shortest distance from the lines between Γ_{Li} and Γ_{Lj} if the two lines do not have the real intersection.

The method for the non-linear optimization is iterative. For this non-linear square problem, the Levenberg-Marquardt algorithm (LMA) is used and integrated by C++ code. The algorithm finds the minimization quickly, mostly after less than 10 iterations. The optimum algorithm is a damped Gauss-Newton method based on the Jacobian J and damping parameter μ. The step h_{lm} is defined by

$$\left(J^T J + \mu I\right) h_{lm} = -g \tag{2.21}$$

where $g = J^T \Psi$ and $\mu \geq 0$.

Figure 2.4 shows the experimental setup of the system using an ABB IRB 120 robot arm. Several experiments, including those reported on [31] using an ABB industrial robot (IRB1600) verified the effectiveness of both the proposed method and the developed system. They also suggest the system fits the need for easy to set-up, totally automated, low-cost, and high precision robot offset calibration.

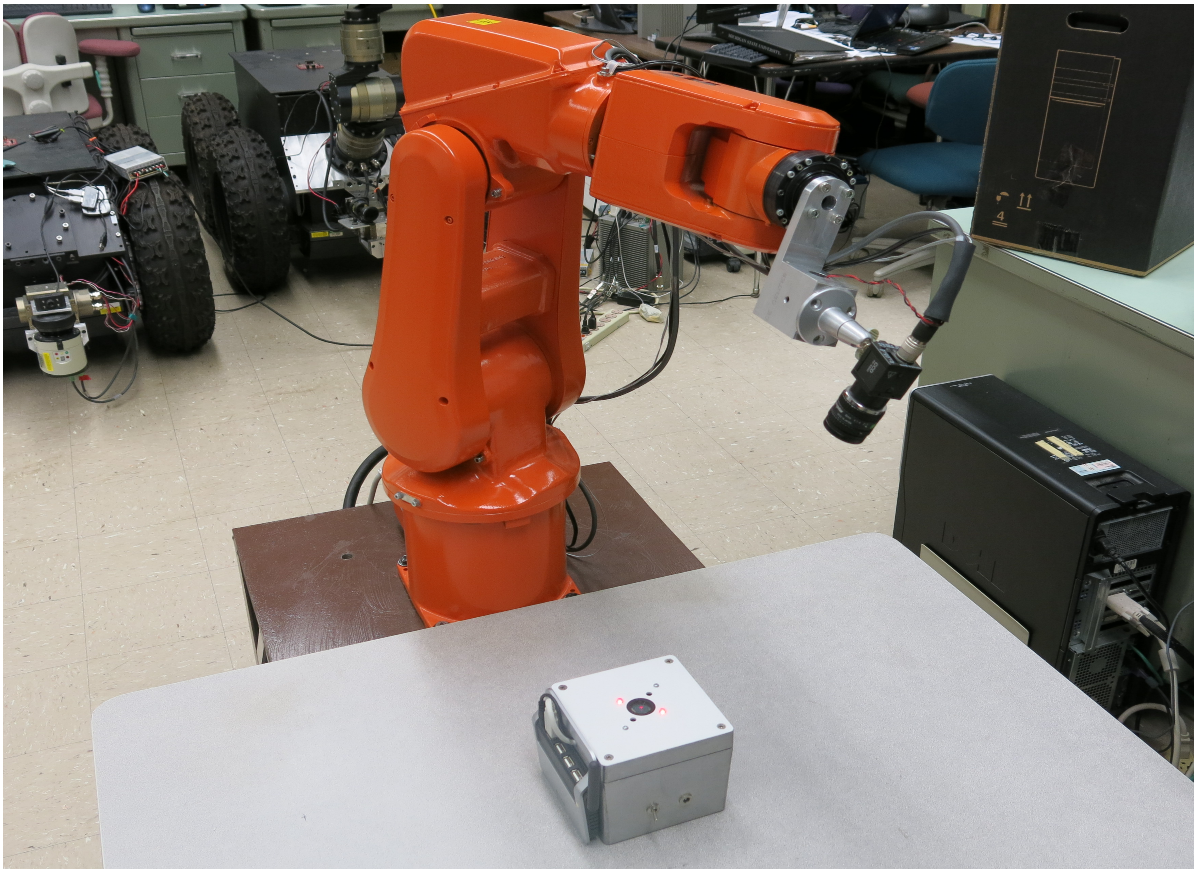


Figure 2.4: Single PSD experimental setup

While this developed calibration system was a promising solution, further experiments suggest several problems concerning the reliability of the system. Experiments were perform using 0.1 μms, 0.5 μms, and 0.01μms of error tolerance. Those experiments suggest that there is a correlation between the error tolerance and the accuracy and reliability in the solution. However, as we improve the calibration solutions by decreasing the tolerance, the real speed of the system had to be compromised, and in the case of using 0.01μms of tolerance, the system was not able to finish the calibration process.

Figure 2.5 shows a magnified view of the center of the PSD in the nano level. In theory, we expect to have at least seven lines intersecting each other in one point in common. If this is true, then our calibration system will guarantee the reliability in the solution for the calibration algorithms. However, in the practical point of view, this is nearly impossible to achieve.

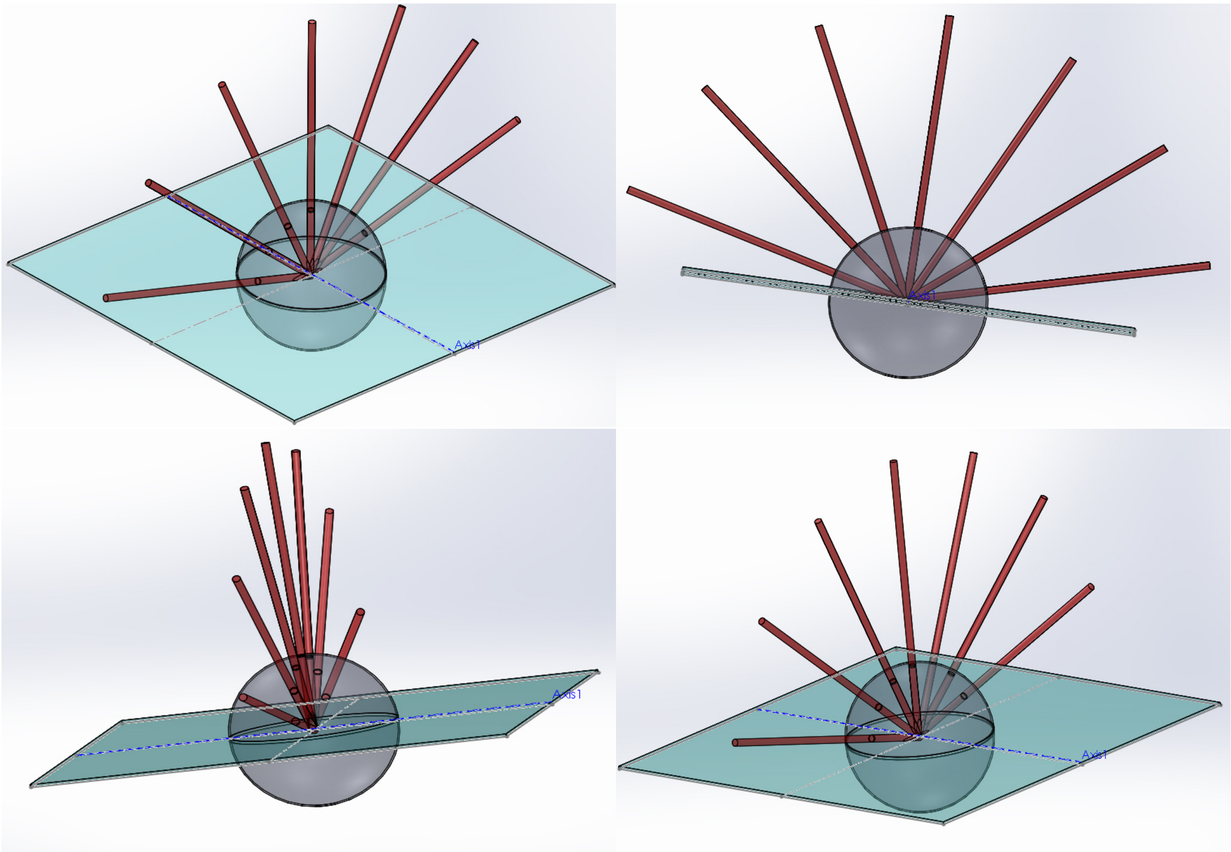


Figure 2.5: Ideal intersection point for the laser lines.

Instead, Figure 2.6 shows a magnified view of the center of the PSD in reality. From this figure you can see that the lines will not intersect each other. They will carry some small errors from each other and will be determined by the error tolerance allowed by the system before moving from one pose to another. This error tolerance is represented by a 3-D sphere equal to the tolerance radius provided. The calibration system proposed by this dissertation will essentially eliminate this problem by guaranteeing a perfect intersection point from the reflection of the sensors. This advantage will become evident when we discuss the experimental results on Chapter 7.

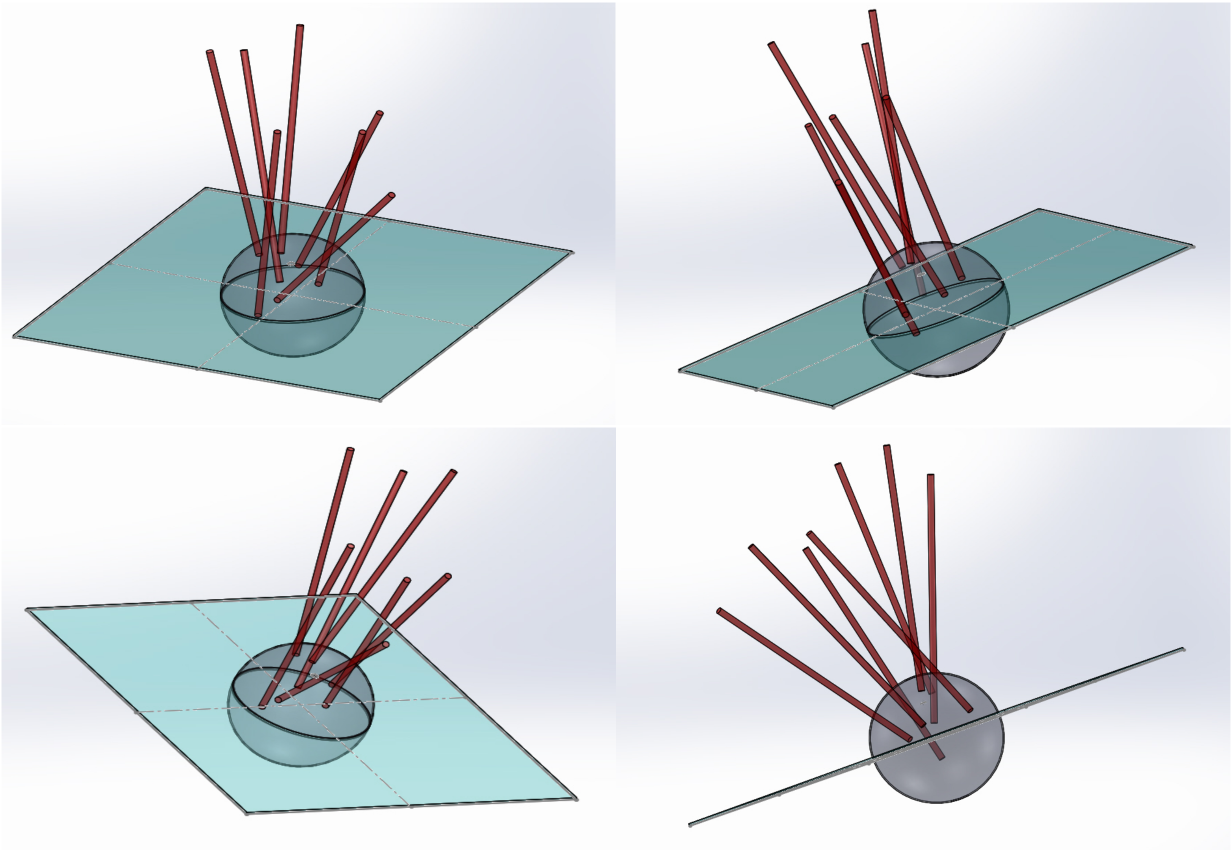


Figure 2.6: Real intersection point for the laser lines.

Chapter 3

Calibration System Overview

This chapter presents an overview of the proposed calibration system and is intended to give the reader an idea of the hardware needed as well as a brief summary of the calibration working principle. As described in Chapter 1, the basic idea of calibration is the comparison of robot external (world coordinates) and robot internal (joint encoder values) representation of a certain location in the task space to determine the error. To perform the calibration task the external representation has to be known. One of the guiding aspects in the design process was that the system has to be portable. This indicates however, that it is not feasible to fix the calibration fixture to the workcell floor and measure its location (also error-prone). Therefore, an alternative way to determine the points relative to the world coordinate system without a fixed location in the workcell has to be found. Therefore, it is necessary to put some extra effort into the design process. A local coordinate system has to be found that better suits our requirements, also the way the robot will move according to our virtual linear constraint must be defined, and finally the hardware needed to accomplish the entire calibration process have to be developed.

3.1 Local Coordinate System

Lets assume a local coordinate system is established on the lower right edge of the calibration fixture as shown in Figure 3.1. Since, as discussed in the overview, a fixed connection of the device in the any external world coordinate system cannot be established, another option must be considered. The world coordinate system (WCS) can basically be placed anywhere since neither the origin nor the orientation are restricted in any way. In this case, the best and also most convenient location for the WCS would be the origin of the calibration fixture coordinate system (CFCS). This way further complex transformations from CFCS to WCS are not required. Another enjoyable side-effect of this WCS position is that this source of positional error can therefore be eliminated in advance. The tradeoff for this however is that one has to come up with a way to determine the PSDs center positions relative to the calibration fixture coordinate systems (CFCS) origin O_f and its coordinate axis x, y, z (see Figure 3.1), in other words, the device itself have to be calibrated before use. Furthermore, a way to relate this to the robot has to be found, too. This can be simultaneously achieved using the calibration system designed for this dissertation.

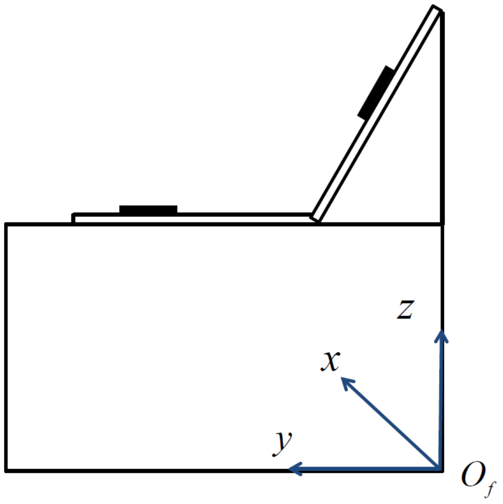


Figure 3.1: Sketch of the calibration fixture device.

3.2 Introduction to the System

Figure 3.2 shows the schematic model of the calibration system, implemented and verify by an ABB robot under lab testing as shown in Figure 3.3 This robotic system comprises of an ABB robot controller (IRC5 Compact) and a six degree of freedom (6-DOF) robot manipulator (IRB120).

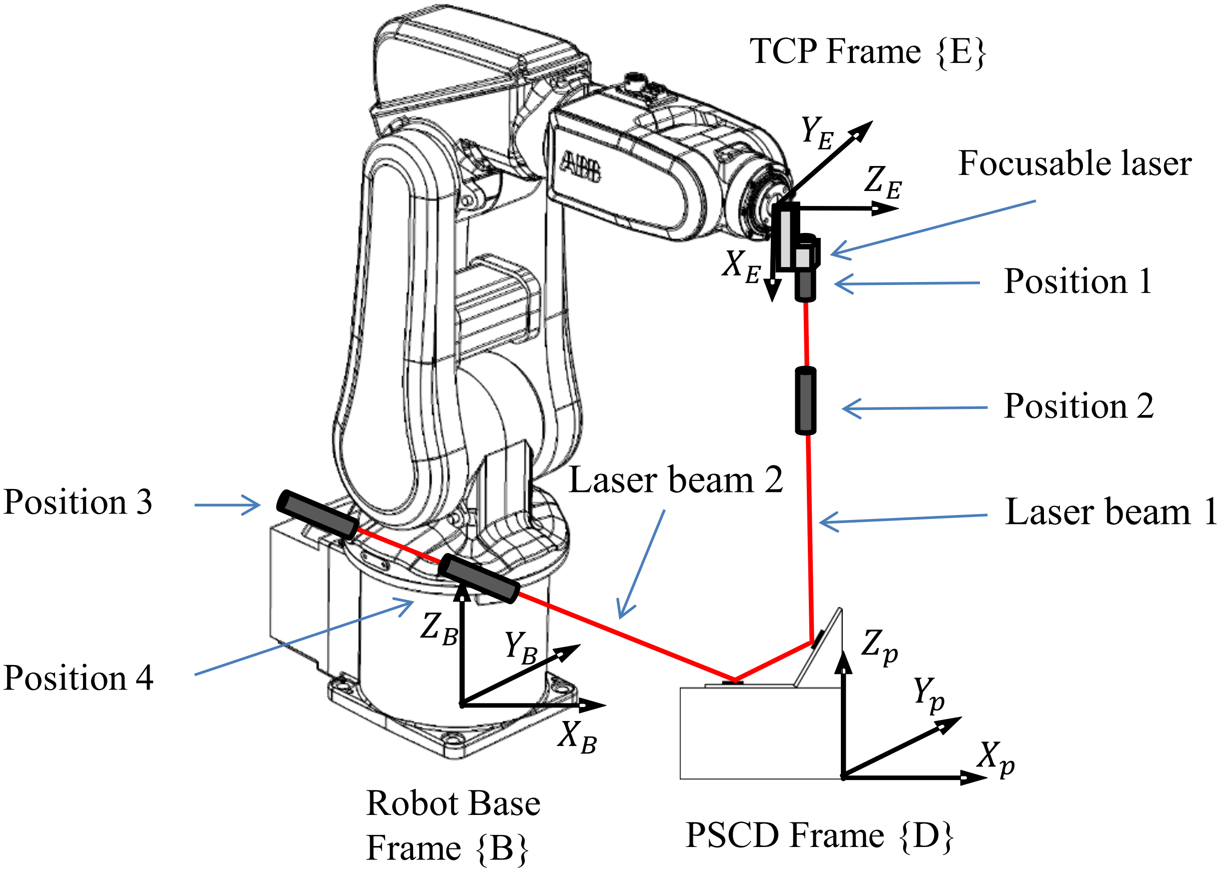


Figure 3.2: Proposed dual PSD calibration system schematic.

In Figure 3.2 the model of the proposed robot calibration system mainly consists of a laser and adapter, a portable dual PSD calibration device, and the robotic system to be calibrated. A laser pointer is mounted on its fixture and rigidly attached to the robot TCP. A magnification of this fixture is shown in Figure 3.4. The laser beam is tuned to align

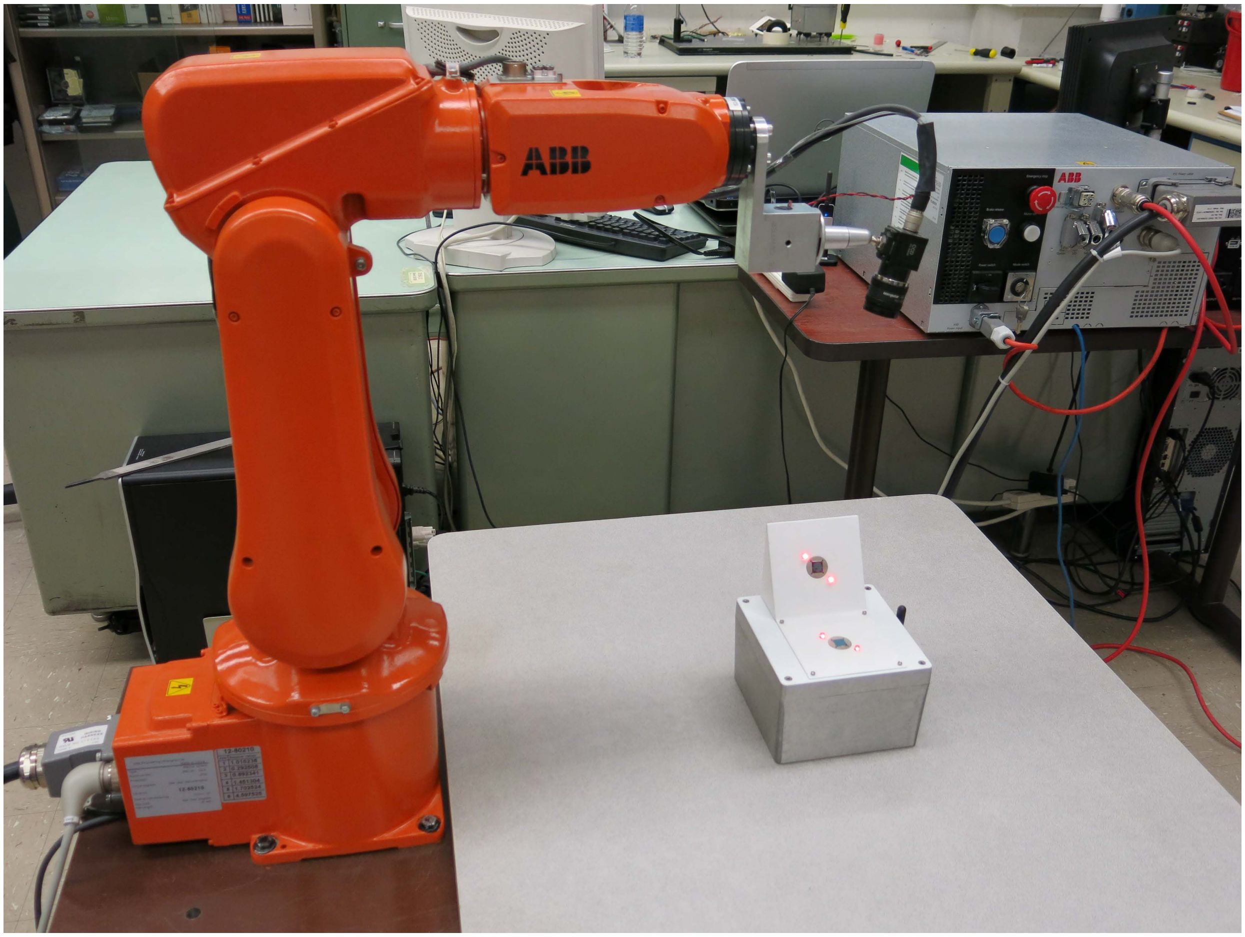


Figure 3.3: ABB IRB120 calibration system implementation

its orientation toward the X-axis of the TCP frame. Two PSD sensors are mounted on a portable custom-built, high-precision fixture. The location of the fixture with respect to the workpiece frame $\{D\}$ is known, while its location with respect to the robot base frame $\{B\}$ is unknown. For the portable PSD device we adopt the segmented PSD for its high precision feedback as shown in Figure 3.5 The segmented PSD has a higher resolution than $0.1\mu m$ in theory. Even under the experimental conditions, its resolution may reach approximately $0.2\mu m$. The protective glass of each sensor was carefully removed in order to be able to create a more pure and perfect reflection from one sensor to the other.

The calibration process, shown in Figure 3.2, is completed by locating the TCP and the laser pointer at four different positions (position 1-4). While the laser pointer is located at position 1 and 2, the laser beam should be aimed at the center of PSD1 and reflected off the PSD1 surface in a direction toward the center of PSD2. Similarly, the laser beam should be aligned to the center of PSD2 and reflected off the PSD2 surface in a direction toward the center of PSD1, while the laser pointer is located at the position 3 and 4. Hence, four sets of robot joint angles can be recorded by the robot controller. The idea behind this method is to be able to overcome the problems we faced with the previous calibration system. By definition, the reflection from one sensor to another, should create a perfect intersection point between two lines. By aiming the two laser spots in the center of each sensor, the ability to find a unique line that passes through these points is guaranteed and as a consequence, improving consistency and reliability in the calibration solutions. Based on the recorded joint angles at these four positions and the robot's forward kinematics, robot calibration algorithms are developed to identify the unknown parameters.

3.3 Calibration System Hardware

This section presents the system hardware needed for our calibration approach. Figure 3.4 shows the laser fixture used to attach both the laser pointer and a camera. The use of a camera is crucial in the effort of making the system autonomous, guiding the laser spot to roughly be align into the active area of the PSD, and therefore, avoiding the need of an operator to manually place the robot position and orientation such that the laser beam hit the surface of the PSD initially. Using the so called camera servoing we are able to solve this problem and guide the laser beam to the initial target. Complete coverage of the roles and the working principle of such camera will be explained in details on Chapter 5.

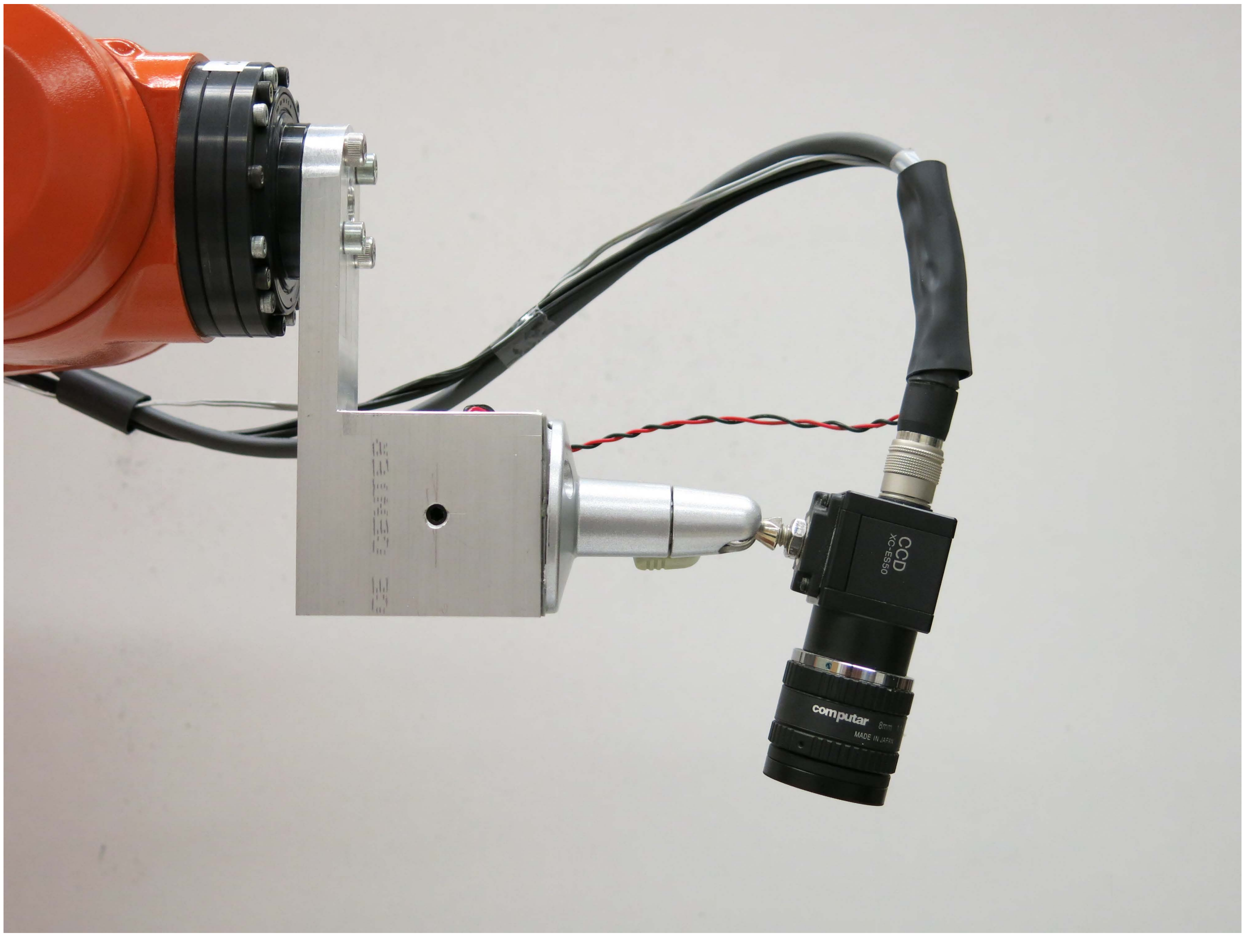


Figure 3.4: Laser pointer and camera attached in the robot TCP.

Another hardware needed and perhaps the most important of them all is the portable dual PSD calibration device also known as the position sensitive calibration device (PSCD). The PSCD used in this dissertation is shown in Figure 3.5. The working principle of this device will be explained more in details in Chapter 4 as well. Note that this is a prototype, a better, lighter and smaller device might be developed in the future. Therefore, by using only a camera, a laser pointer and our PSCD, the complex task of robot calibration can be performed in a simple and cost-effective manner.

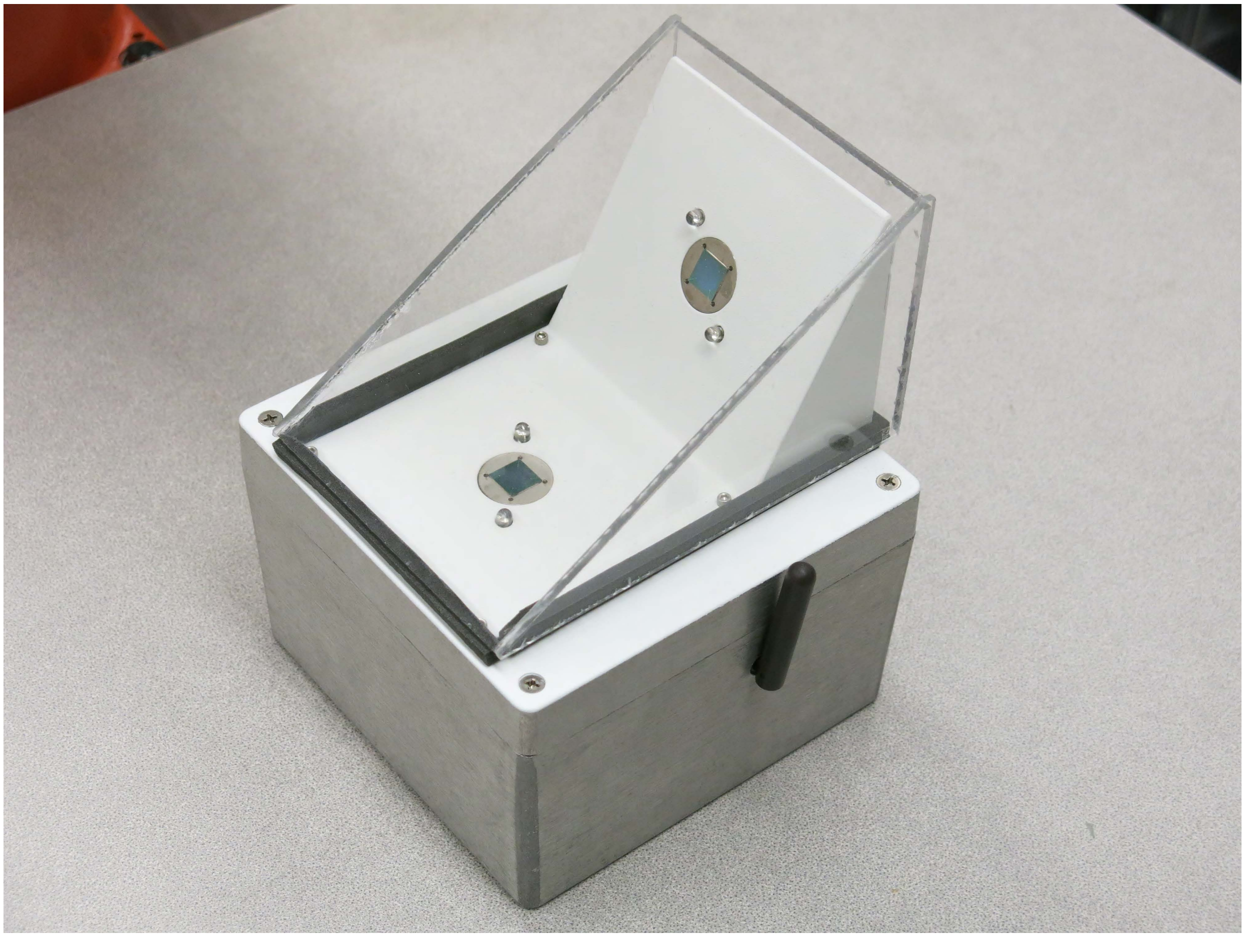


Figure 3.5: The position sensitive calibration device (PSCD).

Chapter 4

Data Acquisition System

This chapter presents the data acquisition system (DAS) used to gather the feedback information needed for the robot controller. It is perhaps the most important component to ensure system accuracy and reliability. First, the fundamentals and working principle of the sensors chosen for this dissertation, i.e. position sensitive detectors (PSDs)will be presented and discussed . Then, a processing board circuit design is analyzed and tested. The position sensitive calibration device (PSCD) prototype is introduced along with the graphical user interface (GUI) used to display the feedback data. Finally, experiments on the PSCD performance were conducted and analyzed to ensure quality positioning that meets our requirements.

4.1 Position Sensitive Detector Working Principle

In general a position sensitive detector is a sensor capable of tracking the location of a light intensity beam on its surface in the nanoscopic level. It essentially consists of either one or two resistive layers placed on the surface of a high resistive substrate, as shown in Figure 4.1. Such a device consists of three semiconductor layers of which only the top one is used to determine the position in the case of one dimensional PSD. In two dimensional devices the bottom layer is used in a similar manner to collect positional information.

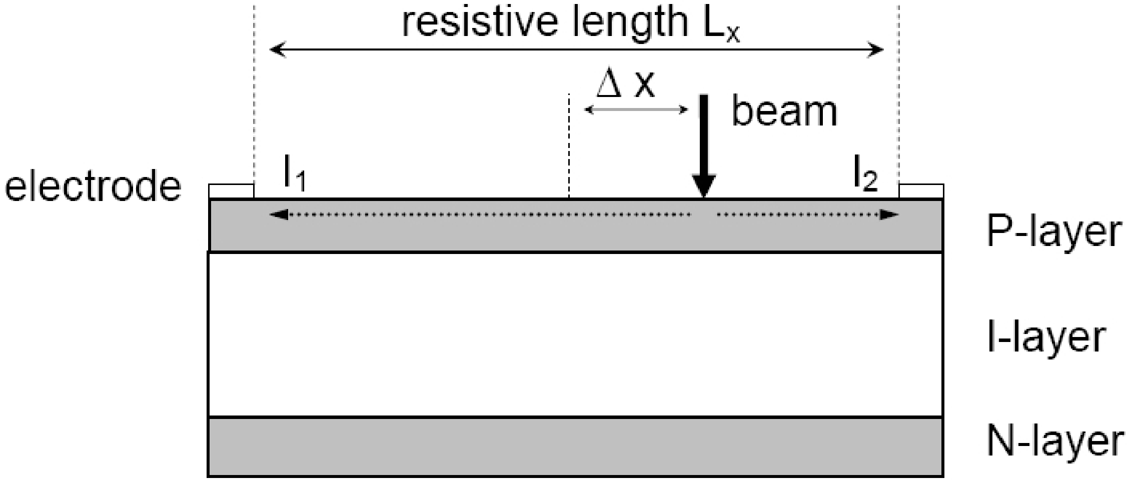


Figure 4.1: Schematic of a one dimensional PSD chip.

The working principle of a PSD is quite simple. If the top P-layer is stimulated with a beam emitted from a light source, an electric charge is generated that is proportional to the light intensity. This formed potential in the resistive layer causes photo-currents to flow between the spot of stimulation and two electrodes on either end of the layer. Due to the uniformity of the resistive layer, I_1 and I_2 are inverse proportional to the distance between the location of the potential, i.e. laser beam light location, and the respective electrode. The above statement can be summarized in the form of an equation as follows,

$$U = R \cdot I \rightarrow I = \frac{U}{R} \tag{4.1}$$

Assuming uniformity of resistive layer, i.e. $\rho = const$ and $A = const$, the following equation holds true,

$$R = \rho \cdot \frac{l}{A} \rightarrow R \sim l \tag{4.2}$$

Combining equations 4.1 and 4.2 we have;

$$I \sim l \tag{4.3}$$

where I is the occurring current (A), U is the generated potential (V), R the resistance of the photo-active area (Ω), l the distance between the light spot and respective electrode (mm), A is the cross-sectional area of the resistive layer (mm^2), and ρ the specific electrical resistance of resistive layer material $(\Omega \cdot m)$. Therefore, the relation between the location of the laser beam spot and the occurring photocurrents can be expressed as [32];

$$I_1 = \frac{\frac{L_x}{2} - \Delta}{L_x} \cdot (I_1 + I_2) = \frac{\frac{L_x}{2} - \Delta}{L_x} \cdot I_0 \tag{4.4}$$

$$I_2 = \frac{\frac{L_x}{2} + \Delta}{L_x} \cdot (I_1 + I_2) = \frac{\frac{L_x}{2} + \Delta}{L_x} \cdot I_0 \tag{4.5}$$

Hence;

$$\frac{I_1 - I_2}{I_1 + I_2} = \frac{2\Delta x}{L_x} \Leftrightarrow \frac{I_1}{I_2} = \frac{L_x - 2\Delta x}{L_x + 2\Delta x} \tag{4.6}$$

where, I_1 and I_2 are the Output currents of resistive layer(A), I_0 is the total photo-current $I_1 + I_2$(A), L_x is the total length of active area (resistive layer) in between electrodes (mm), and Δx is the distance of laser beam spot from the center of the PSD(mm)

There are two main types of PSDs produced by various manufacturers worldwide, lateral PSDs, and segmented PSDs. Both types are produced in both one-dimensional and two-dimensional versions. However, they have some things in common that are unique and superior compared to other optical tracking devices. They offer outstanding positional resolution for a wide spectral range of light used to stimulate the PSD. They also respond to changes almost without delay even without any additional biasing efforts that can be performed to reduce the delay. In this dissertation the two types are analyzed to compare the advantages and disadvantages that could lead to a better PSCD design.

4.1.1 Lateral PSDs

The first type of PSD is usually referred to as lateral PSD. This type of PSD is also available in one-dimensional and two-dimensional realizations. The one-dimensional version is shown in the previous section (Figure 4.1). The chip used for 2-D position detection is manufactured in a similar way. The only difference is that the bottom-layer is also equipped with two electrodes. This second layer works exactly the same way as the top layer. The only difference is that the electrodes mounted on the bottom layer are aligned in a 90 angle relative to the top layer to represent the Y-axis as shown in Figure 4.2. Also in this figure, the sensor is compared with a dime in terms of size.

In the lateral PSDs, the relative two-dimensional position on the active surface of the chip can be expressed as:

$$X = \frac{I_{x1} - I_{x2}}{I_{x1} + I_{x2}} \tag{4.7}$$

$$Y = \frac{I_{y1} - I_{y2}}{I_{y1} + I_{y2}} \tag{4.8}$$

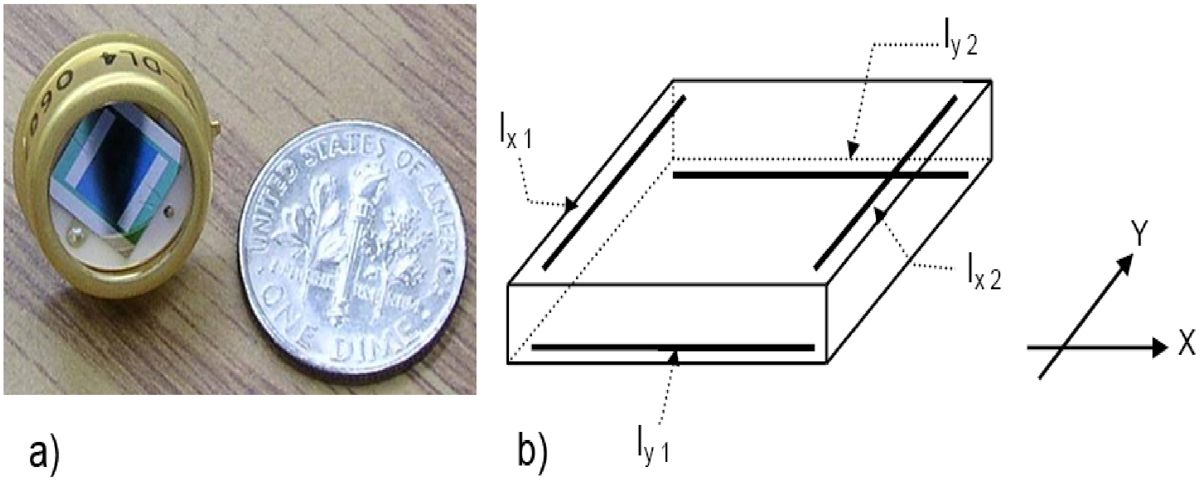


Figure 4.2: a) Photograph of a lateral PSD. b) Schematic illustration.

where X is the relative laser beam position on the X-axis (mm), Y is the relative beam position on Y-axis (mm) and $I_{x1}, I_{x2}, I_{y1}, I_{y2}$ are the photo-currents measured (A) (Figure 4.2). The top resistive layer is used to determine the beam location in X-axis direction. Therefore two electrodes are mounted on the left and right end of the active area to gather the occurring photocurrents. A bottom layer is equipped in a similar manner to determine location in Y-axis direction. The major advantage of this type is that the accuracy of the output is not affected by the spot profile of the beam or its intensity distribution. The positional resolution is lower than the one offered by the segmented type. The achieved resolution of $0.5 \mu m$. Another outstanding property of this type is the position linearity over the whole active surface of the chip. This is important for our task since it allows us to keep the error at a low level during the mapping and compensation process. Furthermore, the area of the chip that offers high resolution is a lot bigger than the one of the segmented.

4.1.2 Segmented PSDs

Segmented PSDs are common substrate photo-diodes that are divided in segments and separated by a gap as shown in Figure 4.3 . This gap, also referred to as the *dead region*, is a section of the chip that is not affected by any form of light stimulating it. The gaps are necessary to electrically isolate PSD segments. Segmented PSDs are produced with either two (one-dimensional) or four (two-dimensional) segments. When light hits the active surface, photo-currents occur in each segment. Those currents can be measured at the respective electrodes attached to each segment to determine the position.

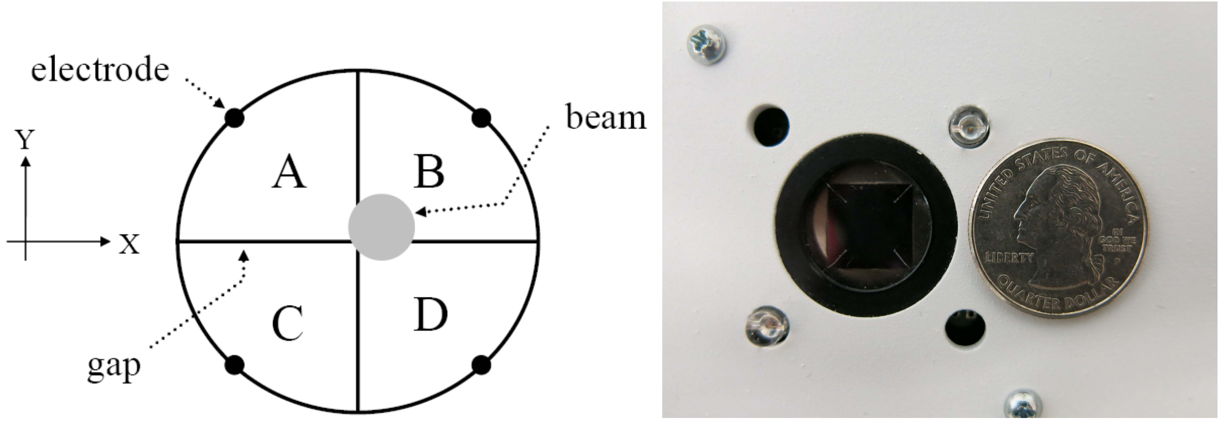


Figure 4.3: Schematic illustration along with the photo of a segmented PSD.

The operation principle is also simple. When a beam of light hits the surface of the chip, photo-currents are generated as explained in the previous section. Again the currents are proportional to the intensity of the light the segment is exposed to. Therefore the relative position of the beam can be expressed as:

$$\begin{aligned} X &= \frac{(I_B + I_D) - (I_A + I_C)}{I_A + I_B + I_C + I_D} \\ Y &= \frac{(I_A + I_B) - (I_C + I_D)}{I_A + I_B + I_C + I_D} \end{aligned} \tag{4.9}$$

where X is the relative laser beam position on X-axis (mm), Y is the relative beam position on Y-axis (mm), and I_A, I_B, I_C, I_D are photo-currents (A) measured in PSD segment noted in the indices as shown in Figure 4.3. The position in X and Y direction can easily be calculated based on the photo-currents that can be measured in each segment. The achievable resolution, i.e. minimal detectable change in beam position, with this type of PSD is approximately 64 nm. However there are several restrictions that need to be fulfilled in order to get correct results:

- The beam has to overlap all segments at all times.
- The diameter of the focused beam has to be larger than the gap in order to reach the active area and generate an output.
- The beams intensity distribution must the uniform since photocurrents in the respective segments are proportional to intensity.

After carefully analyze and compare both PSD types, in terms of accuracy, delay and signal properties, the segmented type was chosen for this dissertation.

4.2 PSD Processing Circuit Design

Before the actual calibration task could be approached, a signal processing circuit had to be designed and set up in order to test the properties of the chosen PSD chips (i.e. position linearity over active chip area, positioning sensitivity etc.). Since the active chip area of the PSD is comparatively small the measurements had to be performed applying high precision tools to position the laser beam over the surface. Therefore, a Signatone CAP-945 high precision probe was used to acquire sufficient accuracy to calibrate the chip. The purpose of this board is simply to process the PSD sensor raw output to be able to determine the relative 2D position of the beam on the chip's surface. The design can be structured into three functional stages. In the first stage, the output signal of the four PSD electrodes is amplified using operational amplifiers. The second and third stages are used to perform the computations needed to be able to determine the spot position, essentially implementing equation 4.9. In the second stage summing amplifiers and differential amplifiers are occupied to generate the input signals for the divider (third stage). A sample of a recommended circuit can be found in [32] as shown in Figure 4.4 in the next page.

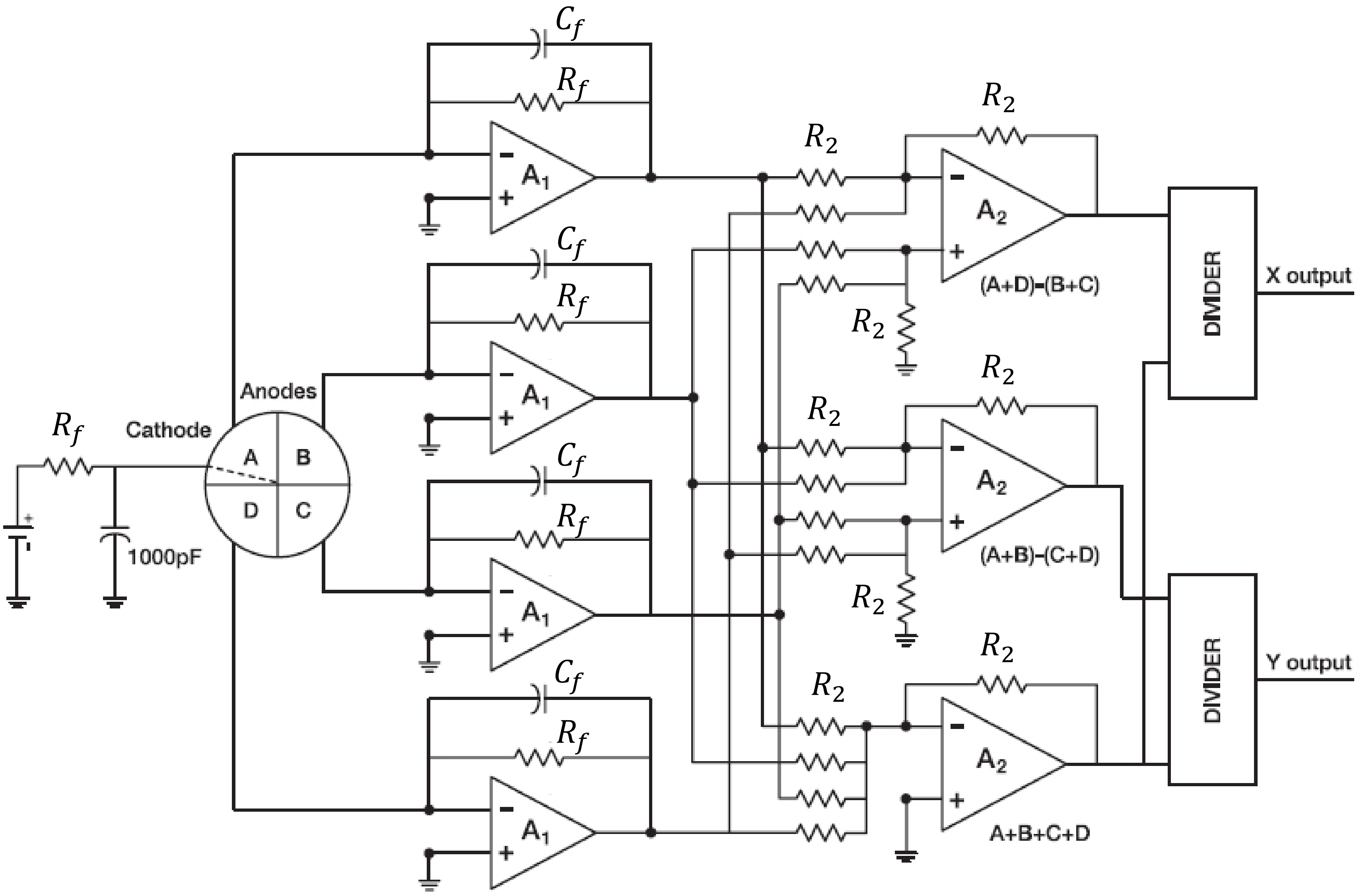


Figure 4.4: Schematic of the circuit board used with segmented PSDs.

4.2.1 PSD Experimental Performance Results

After carefully analyze the circuit schematic, the processing board is designed and the component values were set. Therefore, the first experiments concerning the PSD sensor can be performed at this point. For this purpose, the same type of laser pointer used in the laser fixture on the actual robot was fixed on a Signatone CAP-945 computer aided probe (CAP) perpendicular to the PSD surface. This device allows to position the laser in 3 DOF (x,y,z) over the PSD chip surface with high precision (accuracy 2.5μm, repeatability 1μm). The PSD chip was fixed on a floating, shock absorbing workbench together with the probe to eliminate external influences of e.g., operator movement and vibrations. After the setup was prepared two different experiments were conducted with segmented PSDs. The first experiment was completed to determine if the positional output of the board is dependent on the direction the position is reached from. Therefore, the laser was moved in a straight line over the PSD surface in both directions. First, from bottom to top and then vice versa. The second experiment deals with the linearity of the output signals of the two positional channels (X,Y). In order to achieve results with sufficient positional resolution the laser was moved over the chip surface in two cycles. In the first cycle, the probe was moved on a straight line parallel to the y-axis with a line spacing of approx. 0.2 mm. Therefore, pseudo-continuous y-positions with discrete x-locations were recorded. In the second cycle, the y-position remained discrete in 0.2 mm steps while the x-position was altered continuously.

Figure 4.5 along with Figure 4.6 show the correlation between the segmented PSDs physical surface layout and the respective signal recorded during sweep. The almost linear inclining section at the beginning (0 to 800μm) and end (6800 to 7600μm) of the sweep are caused by the laser beam moving in and out of the photo-active area. Except for the section

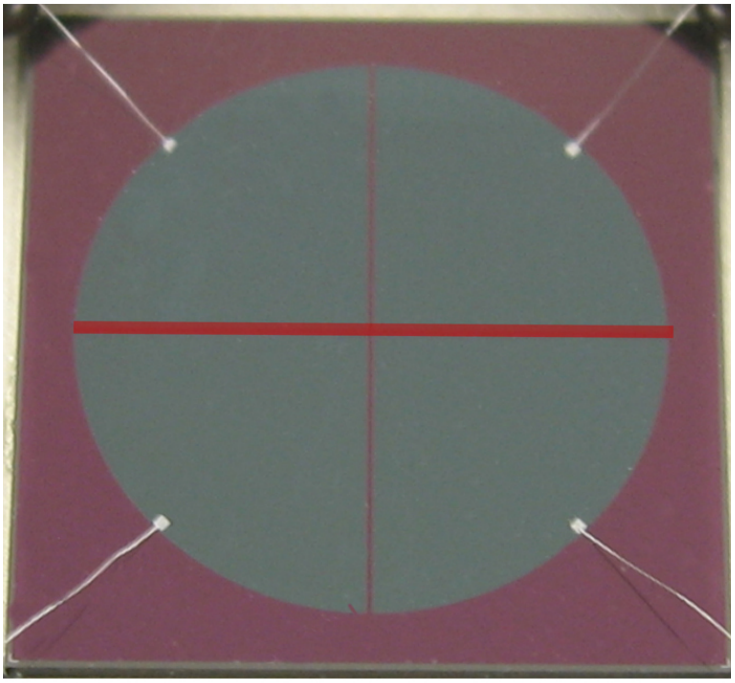


Figure 4.5: Segmented PSDs physical surface layout

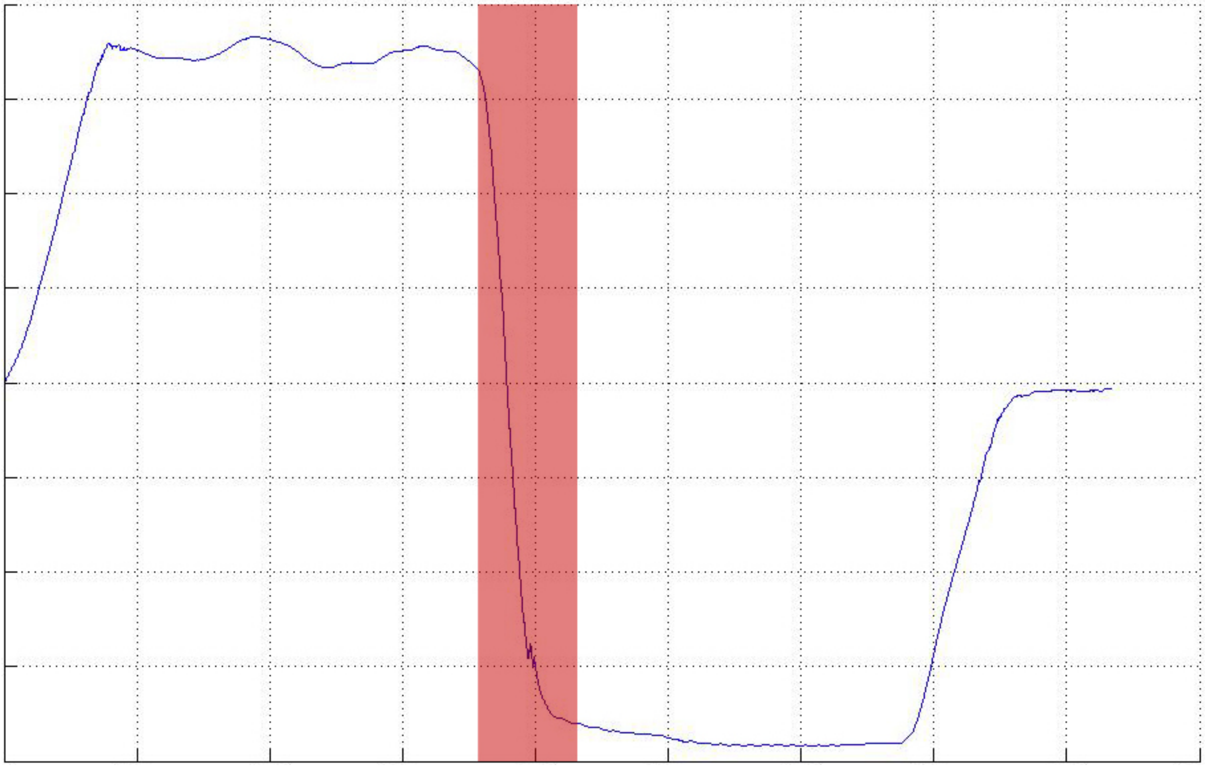


Figure 4.6: Plot of the signal recorded during sweep.

close to the middle gap (3700 to 4200μm) the sensor shows a rather linear output in respect to the change of beam position. As presented in Figure 4.6 the changes in signal can again be traced back to the design features of the sensor. The data sets presented were gathered while performing a y-axis sweep with an x-location slightly right of the middle gap. In the first part, a steady and almost linear incline in output voltage can be observed (position 0 to 800μm). This can be associated with the sensor approaching the photo-active area form the outside and slowly moving positive y-direction towards the middle. Once the peak is reached, the signal maintains at nearly the same level while the location of the laser beam was steadily altered. Therefore, it can be determined that the relationship between position and output, is strongly non-linear for this section. As the beam traverses the middle gap section (3700 to 4200μm), the output starts to decline again. In this section, a very linear change of output to position can be observed. Since the change of amplitude of this region is considerably high, the positional resolution can be assumed to be at its best at this section. Once the beam passed the section, again the same pattern as before occurs. The output maintains at almost the same level as before declining as the laser beam reaches the edge of the active area (6800 to 7600μm).

Now that the general signal properties of the segmented PSD output were discussed, signal characteristics can be analyzed in more details. A very important one is the occurrence of hysteresis effects. The results are presented in Figure 4.7.

Figure 4.7 shows the plot of two sweeps at identical positions on the segmented PSD but with opposite directions. Minor hysteresis effects can be found especially in the first section (y-position 0 to 2000μm). For high precision area around the middle gap no significant signal deviation of the two recorded outputs (one moving in negative- the other one in positive y-direction over the same part of the surface) are noticeable. Once this region is left however,

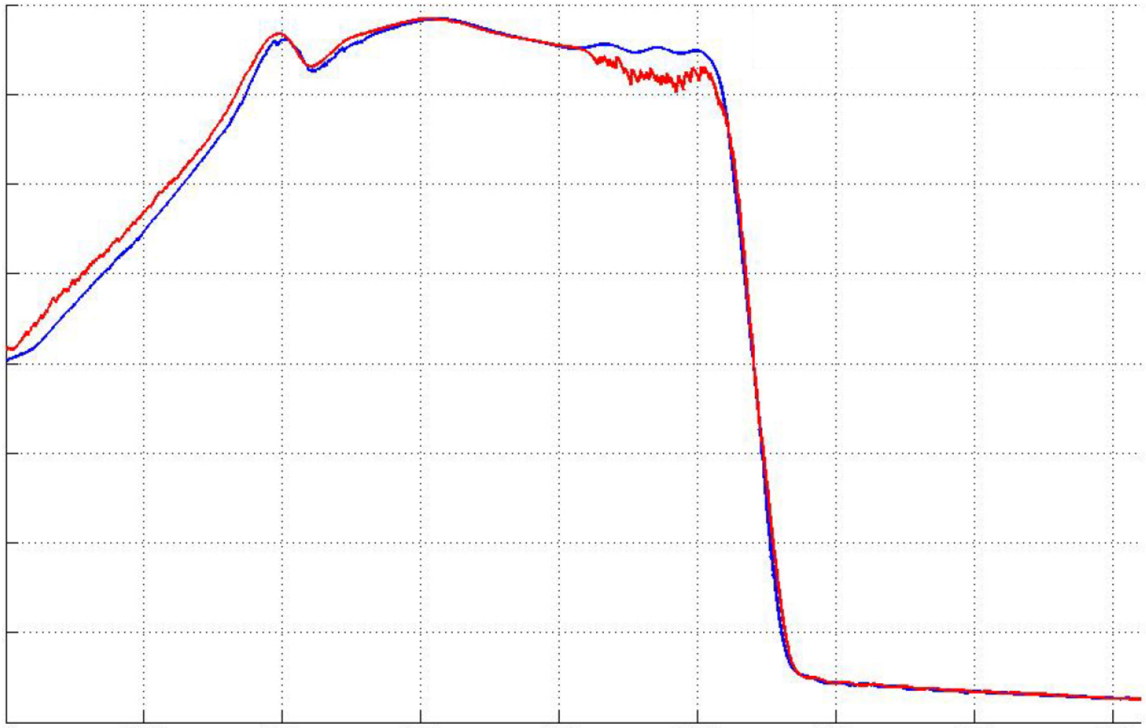


Figure 4.7: Sweeps at identical positions but with opposite directions.

the signal recorded moving in negative y-direction (red) shows a steady offset in the section close to the edge of the active area (2000 to $0\mu m$).

Although similar experiments were performed for the Lateral PSD, in this dissertation only the segmented PSDs are presented in details since the later was the chosen one. In general we choose the segmented PSD because of its superior resolution and good repeatability.

4.3 Position Sensitive Calibration Device (PSCD)

To implement the design of the controller, we first need to be able to get the accurate feedback the controller needs to control the robot movements in the way required by the calibration method. After adequate feedback is provided and tested, the controller can be designed in Chapter 5.

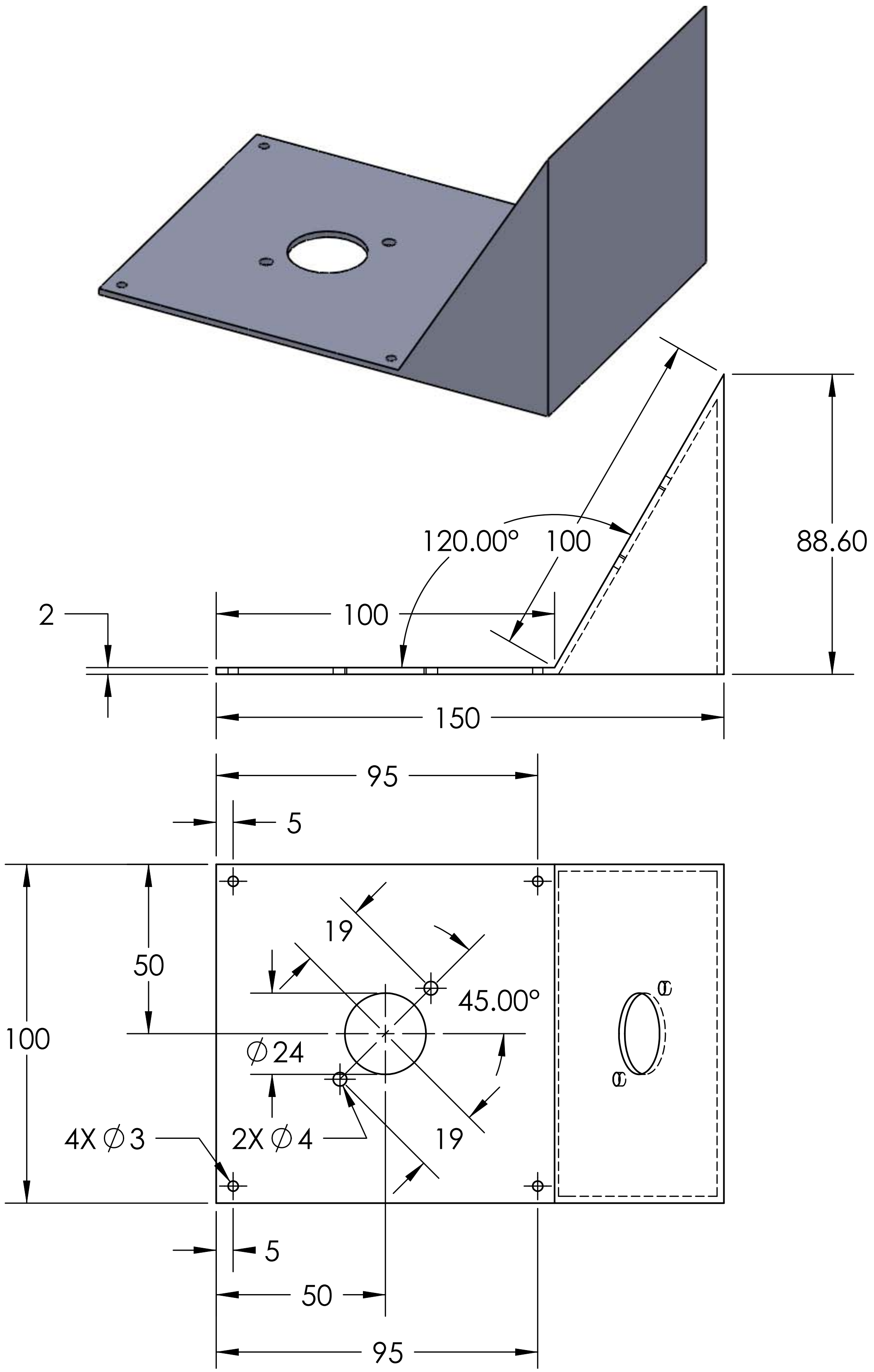


Figure 4.8: PSCD top geometrical design.

Our proposed calibration system approach requires a laser beam to shoot into one PSD in a way that the reflection lands into the other PSD. The idea behind this is to be able to find a unique line based on these two points (found accurately). Therefore we employed two PSDs fixed with an angle between them to found such unique line. As shown in Figure 4.8, the angle between them was chosen to be 120.00^o. We found that more than 120.00^o will make it difficult to find a good reflection from one PSD to the other, and less than 120.00^o will make the system less accurate. Distances between the sensors as well as the distances between LEDs were carefully design and crafted for computational purposes.

Internally our device must be able to carry the signals from the PSDs to the computer in order to control the robot to the desired state. Figure 4.9 highlights the internal components of the device as well as the interaction between the device and the robot controller.

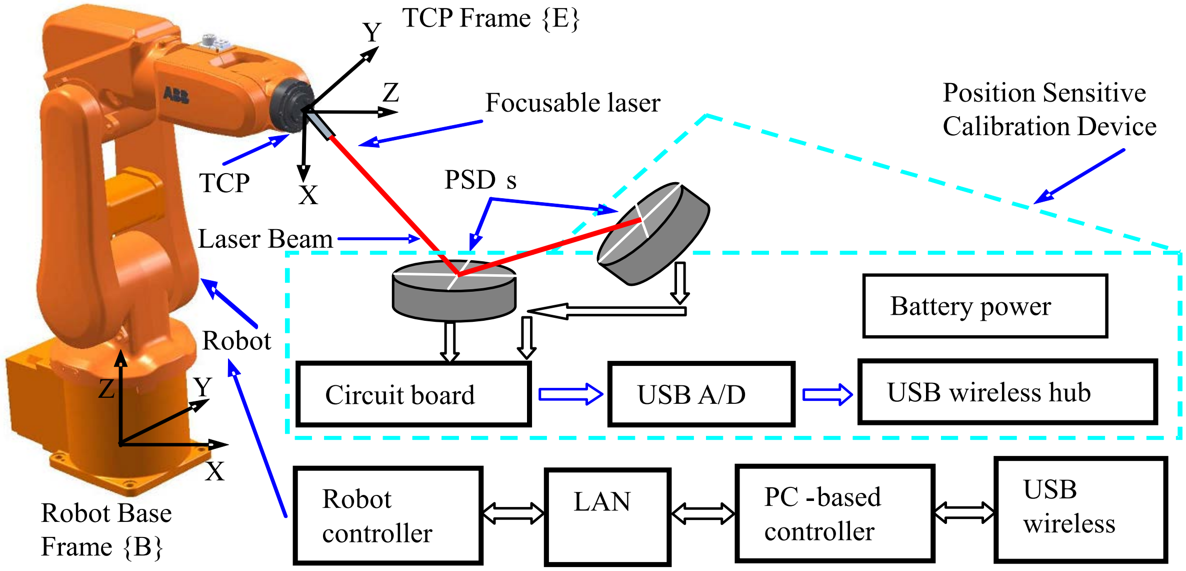
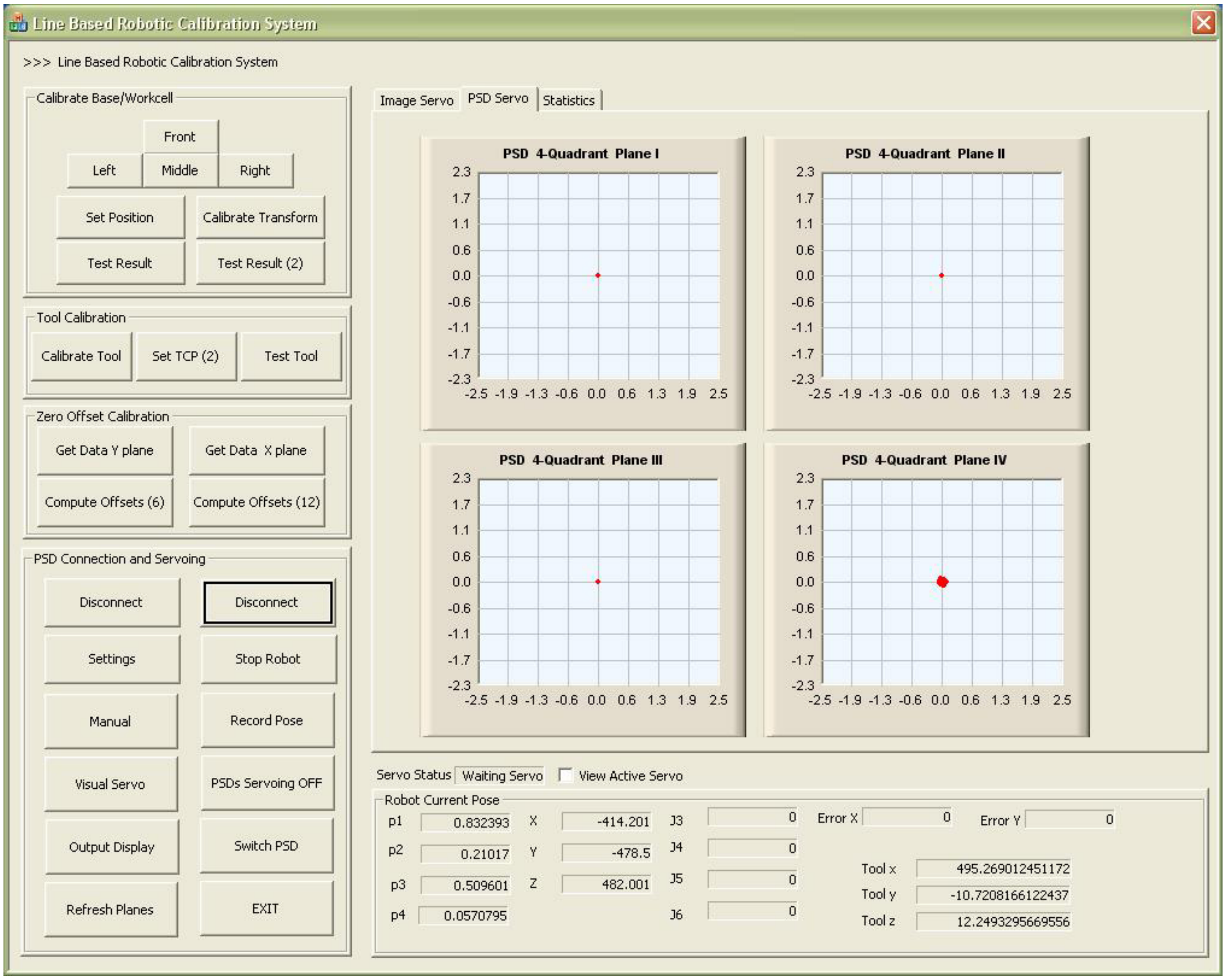


Figure 4.9: PSCD internal components representation and DAS interaction.

After the processing circuit board gets the raw data from the two sensors, the signals are taken by a wireless USB hub from the data acquisition card. The feedback produced by the device have around $47ms$ of delay which is acceptable for our application. Once the data reach the computer we implement our PC-based controller through the graphical user interface so that the robot TCP move to the desired position relative to the sensors.

4.3.1 Graphical User Interface (GUI)

The graphical user interface was design to provide a user-friendly representation of the calibration process. It shows the user option buttons and real time process of PSD-based localization servo as shown in Figure 4.10. The computer-based controller, and calibration algorithms (explained in details in Chapters 5 to 6) are embedded in the designed GUI.



Servo Status: Waiting Servo □ View Active Servo

p1	0.832393	X	-414.201	J3	0	Error X	0
p2	0.21017	Y	-478.5	J4	0	Error Y	0
p3	0.509601	Z	482.001	J5	0	Tool x	495.269012451172
p4	0.0570795			J6	0	Tool y	-10.7208166122437
						Tool z	12.2493295669556

Figure 4.10: Graphical user interface. Note: The text in this image is irrelevant

4.3.2 Feedback Mapping and Testing

Figure 4.11 shows the Plot of the recorded sweep data along x-axis (blue) and the estimated matching polynomial. Middle region shows almost perfect match, but significant deviation occurs close to the edge on both sides. Change in x-location will change the y-signal even if the y-position is maintained constant. This is based on the fact that the lines of equal potential show cushion shape i.e. signal output for constant y-location while changing the x-location is represented by a parabola.

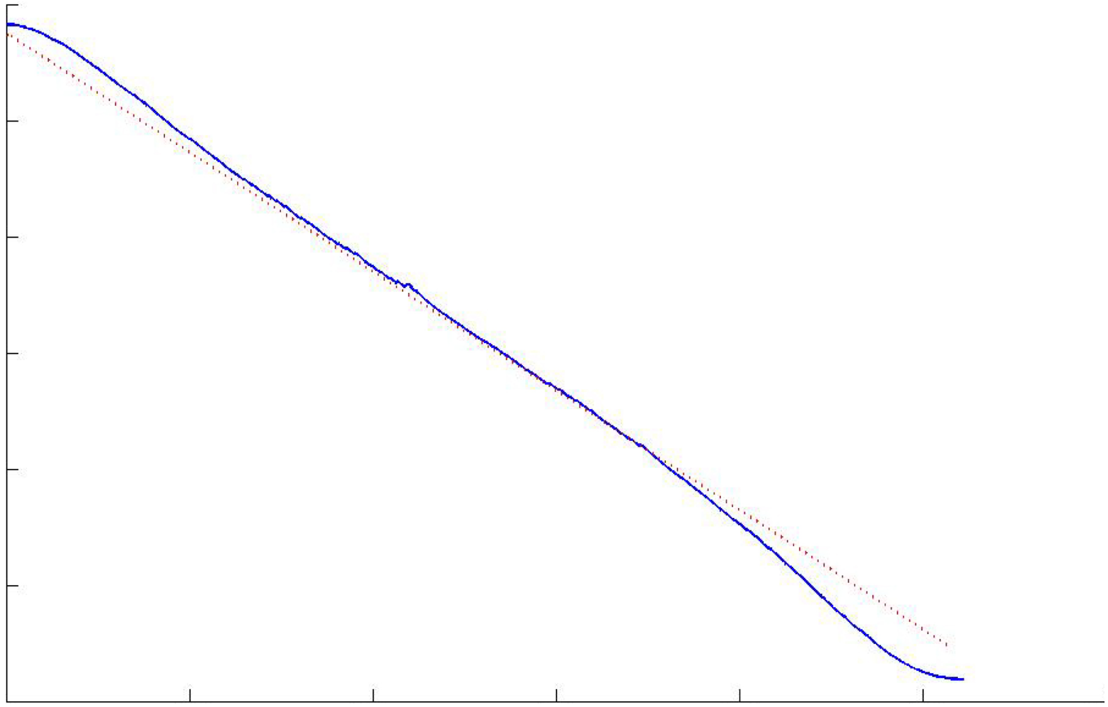


Figure 4.11: Plot of the recorded sweep data.

Hence, the signal needs to be decoupled first to achieve satisfactory mapping results. Therefore a decoupling matrix D_A is employed as follows;

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = D_A \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.10}$$

After the signals are decoupled, the mapping result has to be rotated around the origin (0, 0) to map the PSD axis to the coordinate axis. Hence, a 2D-rotation matrix R is used as follows:

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix} = R \cdot \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{4.11}$$

This results in the final mapping position (x'', y''). Figure 4.12 shows a Screenshot of PC-interface software showing trace of beam movement along straight line in x-direction (left) and y-direction (right). The trace shows the already assumed deviation at both ends of the signal cause by the use of just a single mapping polynomial for the whole active PSD surface. Never the less, the trace follows the beam movement precisely in the middle of the chip.

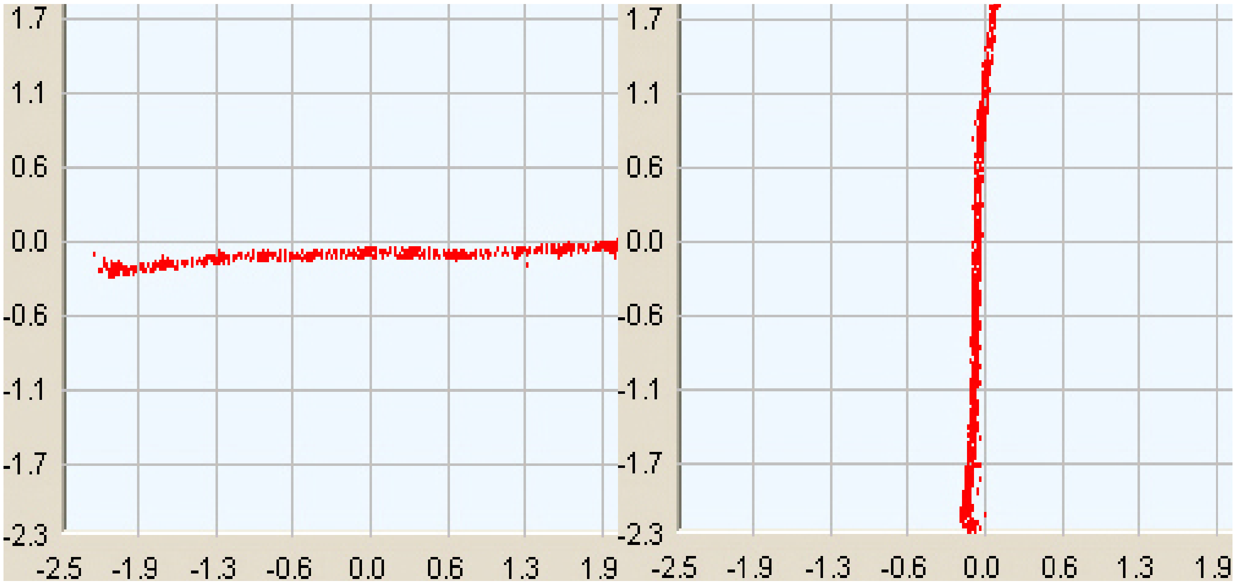


Figure 4.12: Trace of beam movements along x and y directions.

Figure 4.13 shows a trace of beam moving in a square pattern. The small square trace (left) only covering the middle section of the active area shows almost no deviation from the movement (lines straight, 90 angles) while the trace of the bigger square covering the outside areas of the chip shows the expected curvature and deviation.

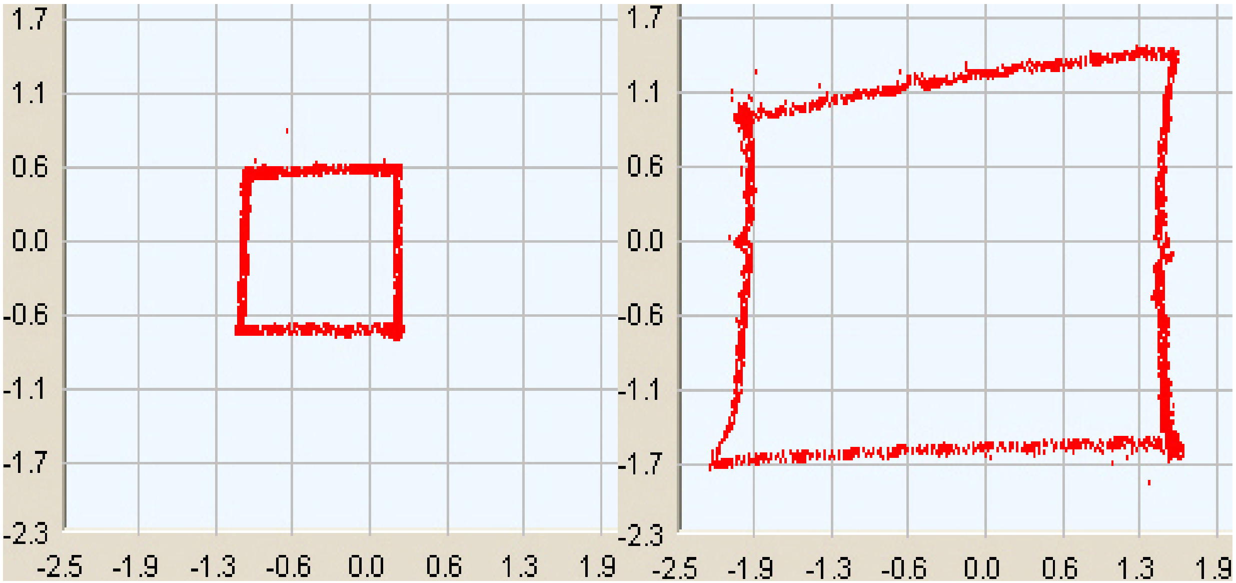


Figure 4.13: Trace of beam moving in a square pattern.

Both tests also confirm once more how critical is to use the center of the sensor to perform our approach to achieve a high accuracy positioning.

4.3.3 PSCD Feedback Experimental Results

During the PSCD performance experiments we essentially test the feasibility of the developed device (after all components were assembled and mapped) to provide feedback to the PC-based controller. After proper calibration and mapping of the sensors were performed, we were able to determine with high accuracy, the location of the laser beam in the nanoscopic scale. For the first stage of the experiment, the same type of laser pointer used in the laser fixture on the actual robot was fixed on a Signatone CAP-945 computer aided probe (CAP) perpendicular to the PSD surface as shown in Figure 4.14. This device allows to place the laser beam in 3 DOF (x,y,z) over the PSD active surface with high precision and accuracy (accuracy 2.5μm, repeatability 1μm). The second stage of the experiment deals with changing in orientation of the beam while maintaining the beam in the center of the first sensor simultaneously. In this case we used Newport 481-A to control orientation of the laser over this point.

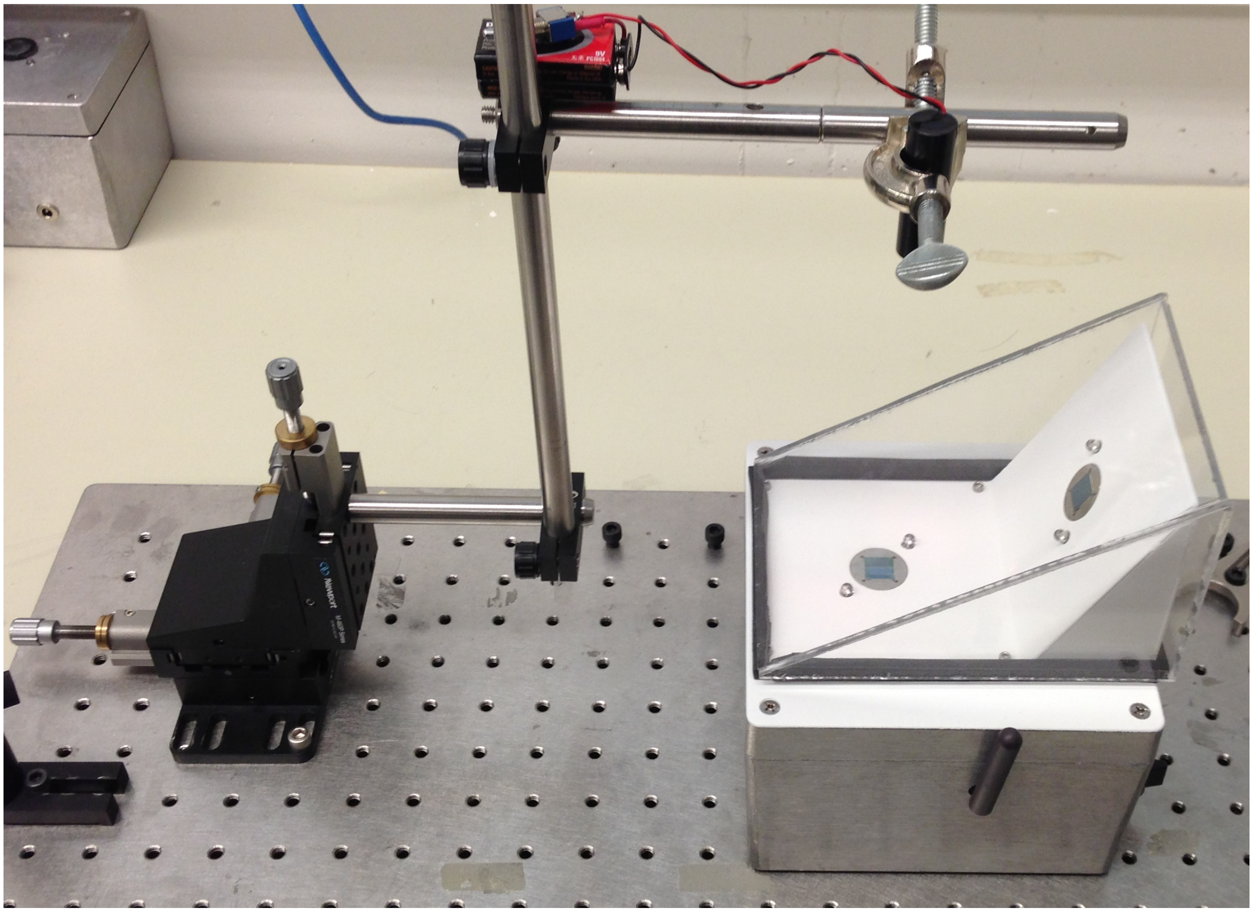


Figure 4.14: PSCD experimental setup.

Figure 4.15 shows the results of the experiments. The lower plane shows the feedback from the PSD1 while the upper plane shows the feedback obtained by the PSD2. For the first stage, we draw linear movements to show how accurate we can determine and control the laser spot over the sensor and eventually reach the center of the sensor which is ultimately the point we are looking for. After it reaches the center of PSD1, we proceeded to control orientation to be able to find the center of PSD2. The ability of the PSCD to carry accurately the feedback information needed for the controlled system is verified by the results obtained from the GUI and presented in Figure 4.15 on the next page.

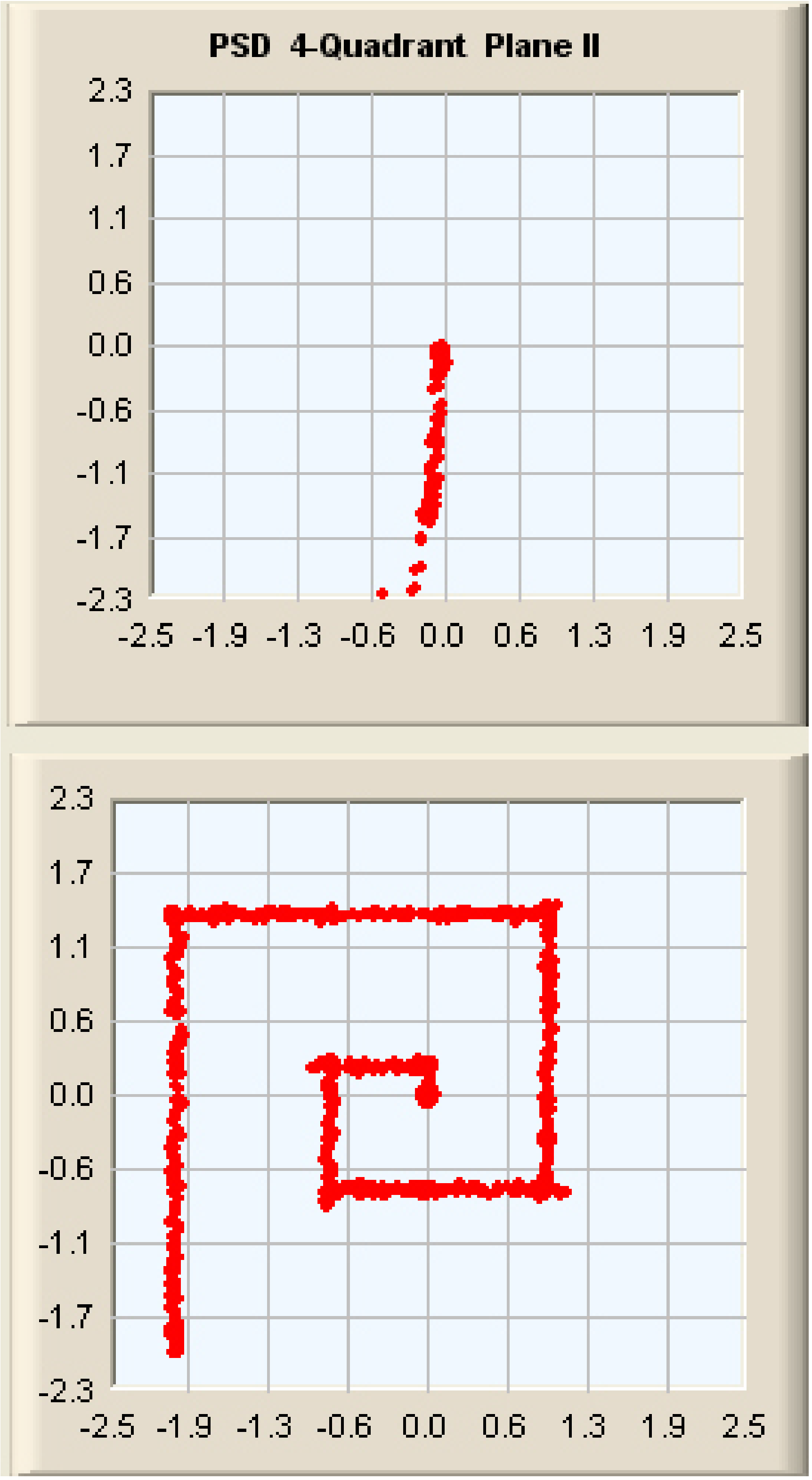


Figure 4.15: PSCD experimental performance results.

Chapter 5

Robot Control

The subject to be discussed in this chapter is amongst the most crucial and important elements of the proposed calibration system: the robot controller. During the calibration process the controller is required to move the robot TCP to a centered position over one of the two PSD sensors automatically. A very similar problem is known in the field of robotics as visual servoing. A good introduction to this topic can be found in [14]. Afterwards, the reflection from the PSD sensor must also be controlled to reach a centered position of the other sensor simultaneously. It is a challenging problem that must be overcome in order to make the entire calibration system an automated and faster procedure. After adequate feedback is provided and tested, a higher level of control can be designed that satisfies the calibration process goals. First, a breakdown of the controller design will be discussed and presented. Simulation results will also be presented at the end of Chapter 5 to test the feasibility of the designed controller.

5.1 Controller Design Overview

In order to address the problem we used a blend of visual and PSD-based servo controllers divided into 3 stages. All stages are meant to control the same robotic system, the difference will be the type of errors determined by different feedback sources. In the first stage, the robot TCP will be controlled by image-based visual servo control. A good piece of work related to this topic can be found in [33]. At this point the control task will be limited to find a rough approximation of one of the two sensors as well as to determine the length of the laser line using the method described in [34], also shown below in Figure 5.1.

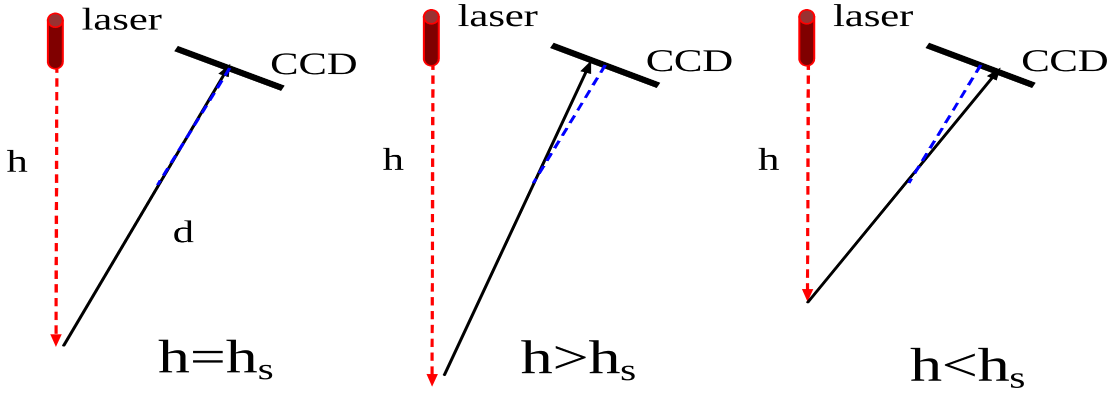


Figure 5.1: Schematic of laser line length control.

The camera is mounted and tilted towards the laser spot to capture the image over the surface forming a triangle. The depth d can be computed accurately using the triangular relationship. After the laser pointer hits the active area, the controller switches to the second stage using PSD-based servo control for an accurate positioning. Once the laser spot is guided to the center of the first PSD, i.e. PSD1, then the third and final stage is applied. At this point, we should have the laser line length information from the first stage. Therefore, to modify the kinematic model of the robot such that the point found in the second stage is fixed. Once this step is done, the orientation of that fixed point can be controlled such that the reflection hits the center of the second PSD, i.e. PSD2, without changing its location in PSD1. At this stage, a similar approach was used to move the laser beam to the center of PSD2 but this time, controlling only orientation. A block diagram of the entire process is shown below in Figure 5.2,

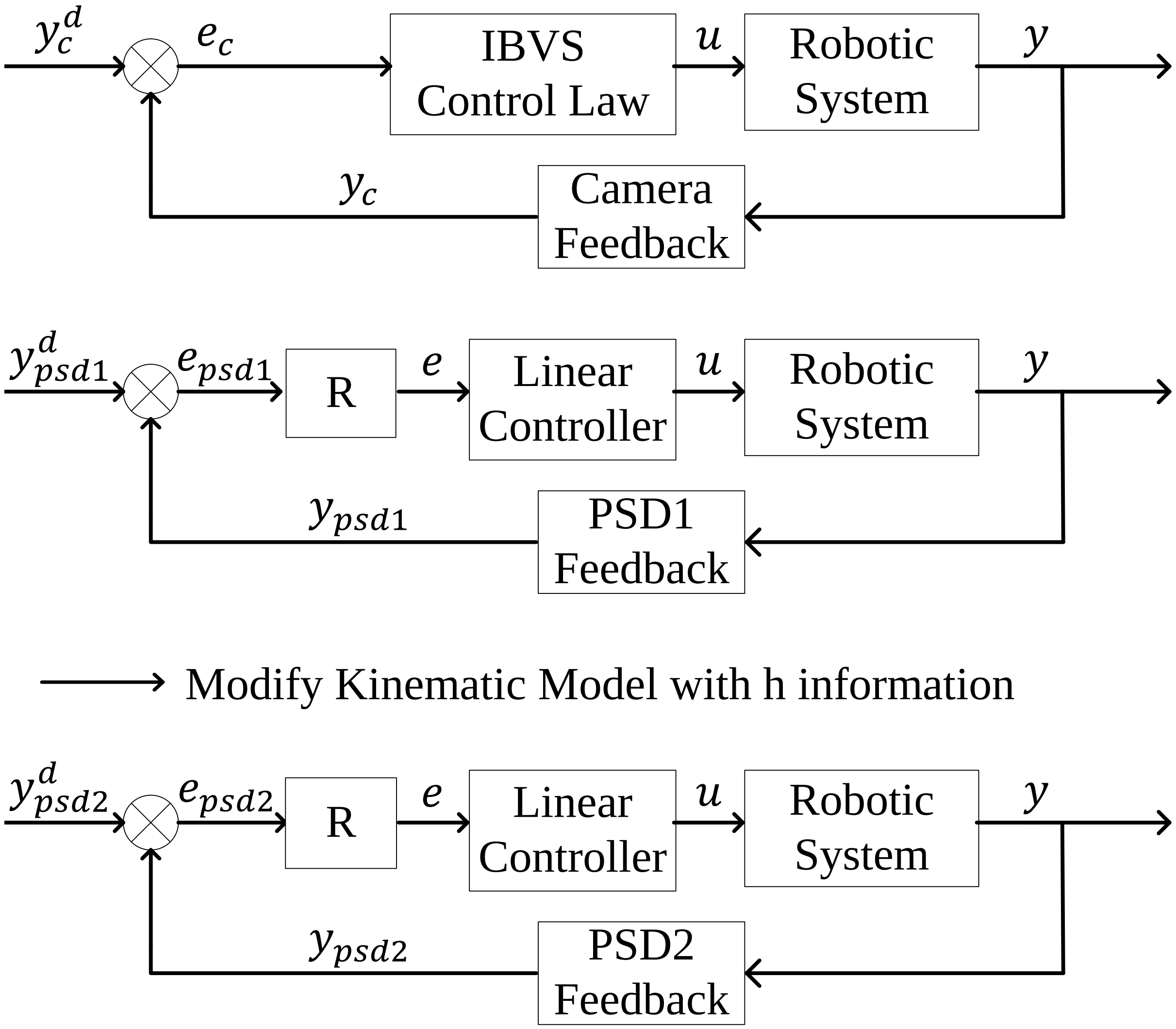


Figure 5.2: Robot control system block diagram.

where R is the matrix to translate the errors given by the PSD to those in the robot TCP and is computed online.

5.1.1 Linear Control Law

First of all, the kind of control scheme that best suits this specific task has to be chosen and developed. In this case the computed-torque controller as presented in [35] seemed to be the most promising controller. From this controller family the PD+G controller was chosen for this dissertation. This control model will be utilized to linearize the six degree of freedom (6DOF) nonlinear system which is essential for simulation purposes. To begin with this process, first the general dynamics of a robot arm will be discussed. Detailed derivations can be found at [35] (for symbols explanation see Table 5.1). The general robot arm dynamics are given as;

Table 5.1: Symbol definitions

Symbol (deg)	Name
q	Joint variables
$M(q)$	Inertia matrix
$V(q, \dot{q})$	Coriolis / Centripetal vector
F_v, F_d	Viscous friction matrix and dynamic friction vector
$G(q)$	Gravity vector
τ	Input vector
τ_d	Disturbance vector
$e(t)$	Tracking error
$q_d(t), q(t)$	Actual and desired trajectory

$$M(q)\ddot{q} + V(q, \dot{q}) + F_v\dot{q} + F_d + G(q) + \tau_d = \tau \tag{5.1}$$

These relatively complex terms can be simplified for convenience by merging the nonlinear terms into the vector $N(q, \dot{q})$:

$$N(q,\dot{q}) = V(q,\dot{q}) + F(\dot{q}) + G(q) \tag{5.2}$$

This yields;

$$M(q)\ddot{q} + N(q,\dot{q}) + \tau_d = \tau \tag{5.3}$$

The tracking error can be defined as the deviation of the actual trajectory from the desired trajectory, i.e.;

$$e(t) = q_d(t) - q(t) \tag{5.4}$$

Differentiating equation 5.4 and solving 5.3 for $\ddot{q}$ will lead to;

$$\ddot{e}(t) = \ddot{q}_d(t) + M^{-1}(N + \tau_d - \tau) \tag{5.5}$$

Defining the control input u as,

$$u = \ddot{q}_d(t) + M^{-1}(N - \tau) \tag{5.6}$$

yields the computed-torque control law:

$$\tau = M(\ddot{q}_d - u + N) \tag{5.7}$$

The control law characterizing the class of computed-torque-like controllers can be denoted as:

$$\tau = \tilde{M}(\ddot{q}_d - u + \tilde{N}) \tag{5.8}$$

The choice of $\tilde{M}$ and $\tilde{N}$ characterize the specific type of controller of this family. Since a PD plus gravity controller was chosen they are set to $\tilde{M} = I$ and $\tilde{N} = G(q) - \ddot{q}_d(t)$. This yields:

$$\tau_c = I(\ddot{q}_d - u) + G(q) - \ddot{q}_d = -u + G(q) \tag{5.9}$$

Selecting PD feedback for u will lead us to:

$$\tau_c = -K_v\dot{q} - K_p q + G(q) \tag{5.10}$$

where K_v and K_p denotes the positive scalar constant controller gains.

Figure 5.3 below shows an illustration of the computed-torque based approach. Note that the inner nonlinear loop is nested in the outer linear loop which includes the robotic arm.

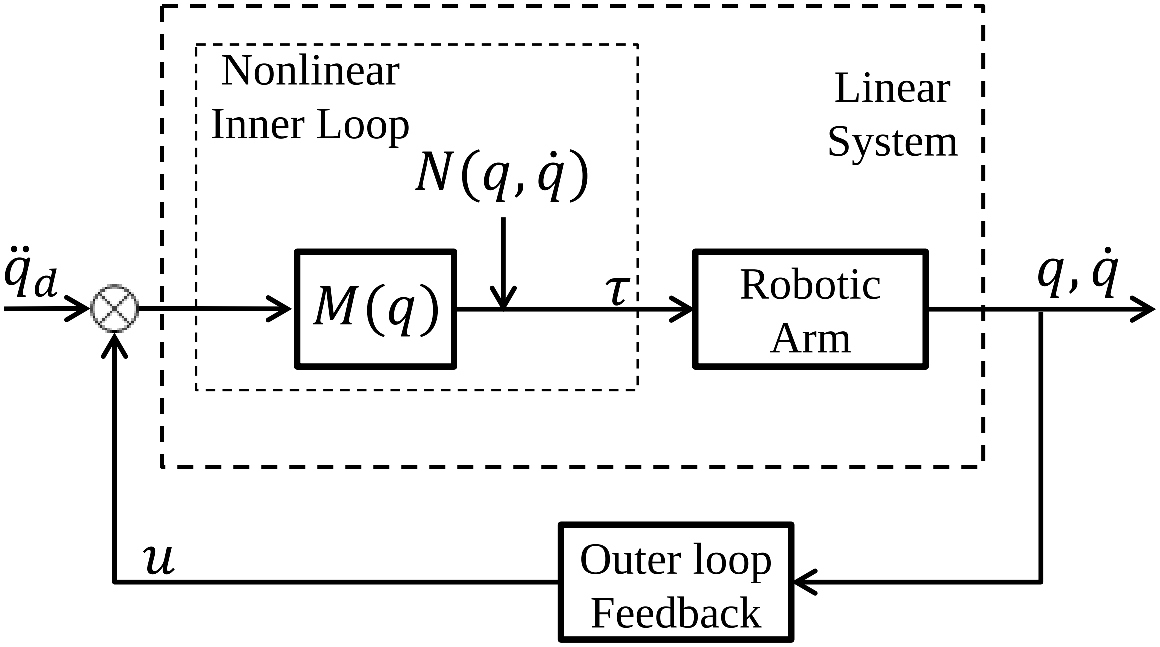


Figure 5.3: Computed-torque approach schematic.

5.1.2 Image-Based Visual Servo Control

The first stage of the controlled method is achieved by using image-based visual servo control (IBVS). In a traditional IBVS system, the image jacobian is commonly used to translate errors from the image feature velocity to the robot TCP velocity [34]. i.e.

$$\dot{f} = J(u, v, z)\dot{r} \tag{5.11}$$

where;

$$J = \begin{bmatrix} \frac{\lambda}{z} & 0 & -\frac{u}{z} & -\frac{uv}{\lambda} & \frac{\lambda^2 + v^2}{\lambda} & -v \\ 0 & \frac{\lambda}{z} & -\frac{v}{z} & -\frac{\lambda^2 + v^2}{\lambda} & \frac{uv}{\lambda} & u \end{bmatrix} \tag{5.12}$$

$f = (u, v)^T$ are the current image coordinate features, $\dot{f} = f^d - f$ is the feature error and λ is the focal length of the camera $\dot{r} = [v, w]^T$. $w = [w_x, w_y, w_z]$ is the angular velocity vector and $v = [v_x, v_y, v_z]$ is the translational velocity vector. The commonly used control law is given by using the jacobian relationship, given that the image jacobian is a full rank square matrix;

$$\dot{U} = \dot{r} = \Gamma J^{-1}(u, v, z)\dot{f} \tag{5.13}$$

where U is the control input and Γ a gain matrix.

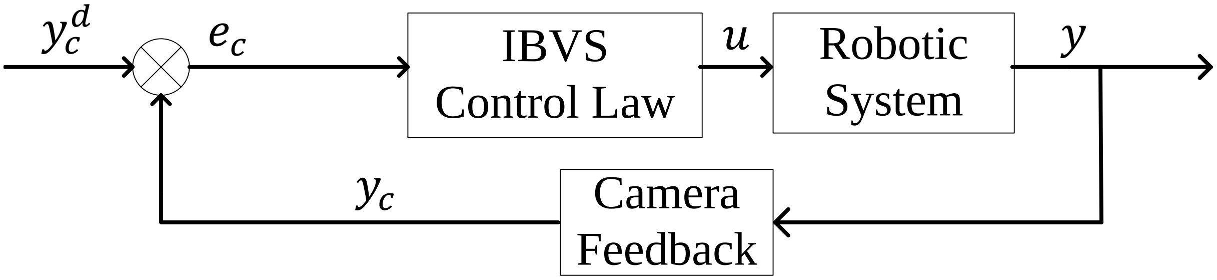


Figure 5.4: IBVS control system block diagram.

5.1.3 Laser Line Length control

To avoid the well-known problem of singularity in the image Jacobian, equation (5.11) can be decomposed into its rotational and translational components, i.e.

$$\dot{f} = J(u, v)w + J(u, v, z)v \tag{5.14}$$

Therefore, we can completely decouple orientation from translation control, i.e., no orientation control is performed when translational control is conducted and vice versa. In the case that only orientation control is conducted orientation is denoted as;

$$w = \Gamma J^{-1}(u, v)\dot{f} \tag{5.15}$$

where $\dot{f} = f^d - f$ represent the error of the LEDs features on the image. However, during translation control the features are the coordinates of the laser beam on over the image plane. Based on height control, the coordinates of the laser beam should not change and the errors are based on the coordinates between the laser spot and the center of the two LEDs. The Logic of our method can be described as follows:

- In Figure 5.1, if $h \neq h$, h will be adjusted using the laser line height control.
- If the desired orientation is not achieved, the orientation control will take over using equation (5.15).
- If there are coordinate errors, then translational control is activated.
- The decoupled IBVS is switched automatically according to the switch logic.

5.1.4 PSD-based Servo Control (Translational)

The second stage of the controlled method is achieved by using PSD-based servo control. Similar to IBVS, the image Jacobian is used to translate errors from the image feature velocity to the robot TCP velocity. In this case, the image is represented by the feedback acquired by the PSCD. For instance, let us define the homogeneous transformation matrix of the base frame {B} as ${}^{P}T_{B}$, which can be written as;

$$ {}^{P}T_{B} = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \tag{5.16} $$

where $R \in \Re$ denotes the rotation matrix of the base frame {B} relative to the PSD1 frame {P}. Similar to the way we perform translation control in IBVS, the orientation will remain constant. This fact implies that movements in the TCP holding the laser pointer will be equals to those in the PSD1 surface. Hence, we only need to get the velocity relationship between both frames to be able to control the robot in the task space coordinates.

Let the 3 components of both, the translational velocity and angular velocity, be represented by ξ ;

$$ \xi = \begin{bmatrix} v & w \end{bmatrix}^{T} \tag{5.17} $$

Since the PSD frame {P} is fixed relative to the base frame {B}, the relationship between them is constant and can be written as;

$$ \xi_{P} = \begin{bmatrix} R_1 & 0_{3x3} \\ 0_{3x3} & R_2 \end{bmatrix} \xi_{B} \tag{5.18} $$

Here ξ_P represents the velocity of the TCP with respect to the PSD frame, while ξ_B represents the velocity of the TCP with respect to the robot base frame. Solving equation (5.18) for ξ_B we have;

$$\xi_B = \begin{bmatrix} R_1^T & 0_{3x3} \\ 0_{3x3} & R_2^T \end{bmatrix} \xi_P \tag{5.19}$$

Now let $s(t)$ denote the vector of position values obtained by the PSD. Then, $\dot{s}(t)$ will represents the position velocity;

$$\dot{s}(t) = \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} \tag{5.20}$$

Because of the orientation remaining constant, i.e. movements are performed only along the Z plane, we can state that $v_z = 0$ and also $w = 0$: Therefore, if we combine this with equations (5.19) and (5.20) we have the following;

$$\dot{s}(t) = \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = R_1 v_B \tag{5.21}$$

where $v_B = [v_x, v_y, v_z]$ represent the components of the translational vector with respect to the robot base frame. Using the PSCD feedback the desired position is defined by the center of the sensor, therefore, the image features error can be represented by;

$$e(t) = s(t) - s^d \tag{5.22}$$

In order to control the TCP position error using the PSD feedback, we shall compute a desired TCP velocity v_B and use for the controller design. This is done by solving equation

(5.21) by v_B;

$$v_B = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = R_1^{-1}\dot{s}(t) \tag{5.23}$$

Therefore, a proportional controller is designed such that;

$$\dot{e} = -K_p e \tag{5.24}$$

Finally, substituting equation (5.24) into (5.23) we have;

$$v_B = -K_p R_1^{-1} e \tag{5.25}$$

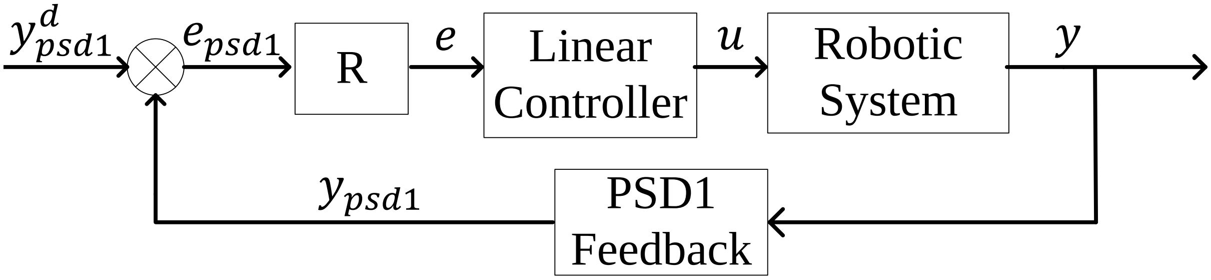


Figure 5.5: PSD1 control system block diagram.

5.1.5 PSD-based Servo Control (Rotational)

The third stage of the controlled method is achieved by using PSD-based servo control with rotational movements over the point found on PSD1. In this case, the image is represented by the feedback acquired by the PSD2. Because we set the orientation previously based on measurements of the PSCD, once we hit the center of PSD1 the reflection should be somewhere in the active area of the second (no IBVS required), very close to the origin as well. For instance, let us define the homogeneous transformation matrix of the base frame {B} as ${}^{P}T_{B}$, which again can be written as;

$$ {}^{P}T_{B} = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \tag{5.26} $$

Letting the 3 components of both, the translational velocity and angular velocity to be represented by ξ ;

$$ \xi = \begin{bmatrix} v & w \end{bmatrix}^{T} \tag{5.27} $$

Since the PSD frame {P} is fixed relative to the base frame {B}, the relationship between them is still constant and can be written as;

$$ \xi_{P} = \begin{bmatrix} R_1 & 0_{3x3} \\ 0_{3x3} & R_2 \end{bmatrix} \xi_{B} \tag{5.28} $$

here ξ_P represent the velocity of the TCP with respect to the PSD frame, while ξ_B represent the velocity of the TCP with respect to the robot base frame. Solving equation (5.28) for

ξ_B we have;

$$\xi_B = \begin{bmatrix} R_1^T & 0_{3x3} \\ 0_{3x3} & R_2^T \end{bmatrix} \xi_P \tag{5.29}$$

Now let $s(t)$ denote the vector of position values obtained by the PSD2. Then, $\dot{s}(t)$ will represent the angular velocity over the reflection in PSD1;

$$\dot{s}(t) = \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} \tag{5.30}$$

Because of the position remaining constant, we can state that $v = 0$ and also $w_z = 0$: Therefore, if we combine this with equations (5.29) and (5.30) we have the following;

$$\dot{s}(t) = \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = R_2 w_B \tag{5.31}$$

where $w_B = [w_x, w_y, w_z]$ represent the components of the rotational vector with respect to the robot base frame. Using the PSCD feedback the desired position is defined by the center of the sensor PSD2, therefore, the image features error can be represented by;

$$e(t) = s(t) - s^d \tag{5.32}$$

In order to control the PSD1 orientation using PSD2 feedback, we shall compute a desired PSD1 angular velocity w_B and use it for the controller design. This is done by solving equation (5.31) by v_B;

$$w_B = \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = R_2^{-1} \dot{s}(t) \tag{5.33}$$

Therefore, a proportional controller is designed such that;

$$\dot{e} = -K_p e \tag{5.34}$$

Finally, substituting equation (5.34) into (5.33) we have;

$$w_B = -K_p R_2^{-1} e \tag{5.35}$$

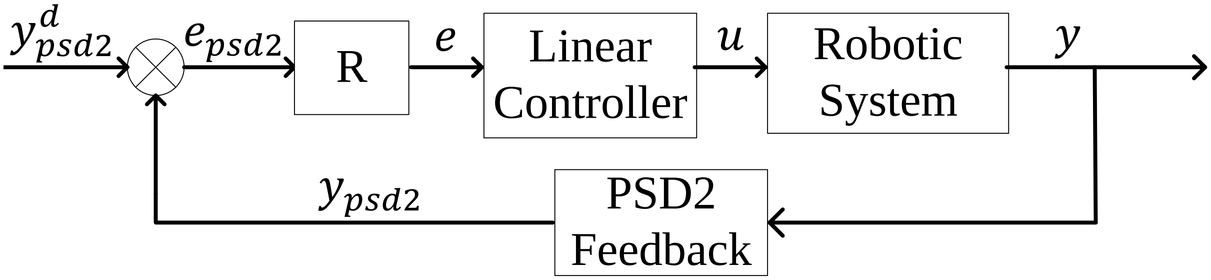


Figure 5.6: PSD2 control system block diagram.

5.2 Controller Simulation & Experimental Results

5.2.1 Simulation Results

Simulations were performed using the kinematic model of the ABB IRB 120 robot for the second and third stages only, since the first stage using IBVS was analyzed and discussed in our previous work [34]. The mathematical model of the 6 DOF robot was used to simulate the dynamic movements of the TCP based on the feedback generated by the virtual PSCD. For the second stage of the controller, the simulation results of PSD servo control (translational) demonstrated the ability to track the position of the TCP relative to the PSD1 down to zero with a $K_p = 2$ as shown in Figure 5.7 at the top. Similarly, in Figure 5.7 at the bottom, the simulations show stability as well for the third stage of the controller using orientation control over PSD2 while keeping position errors over PSD1 equal to zero. The simulation results demonstrated the feasibility of the controller designed.

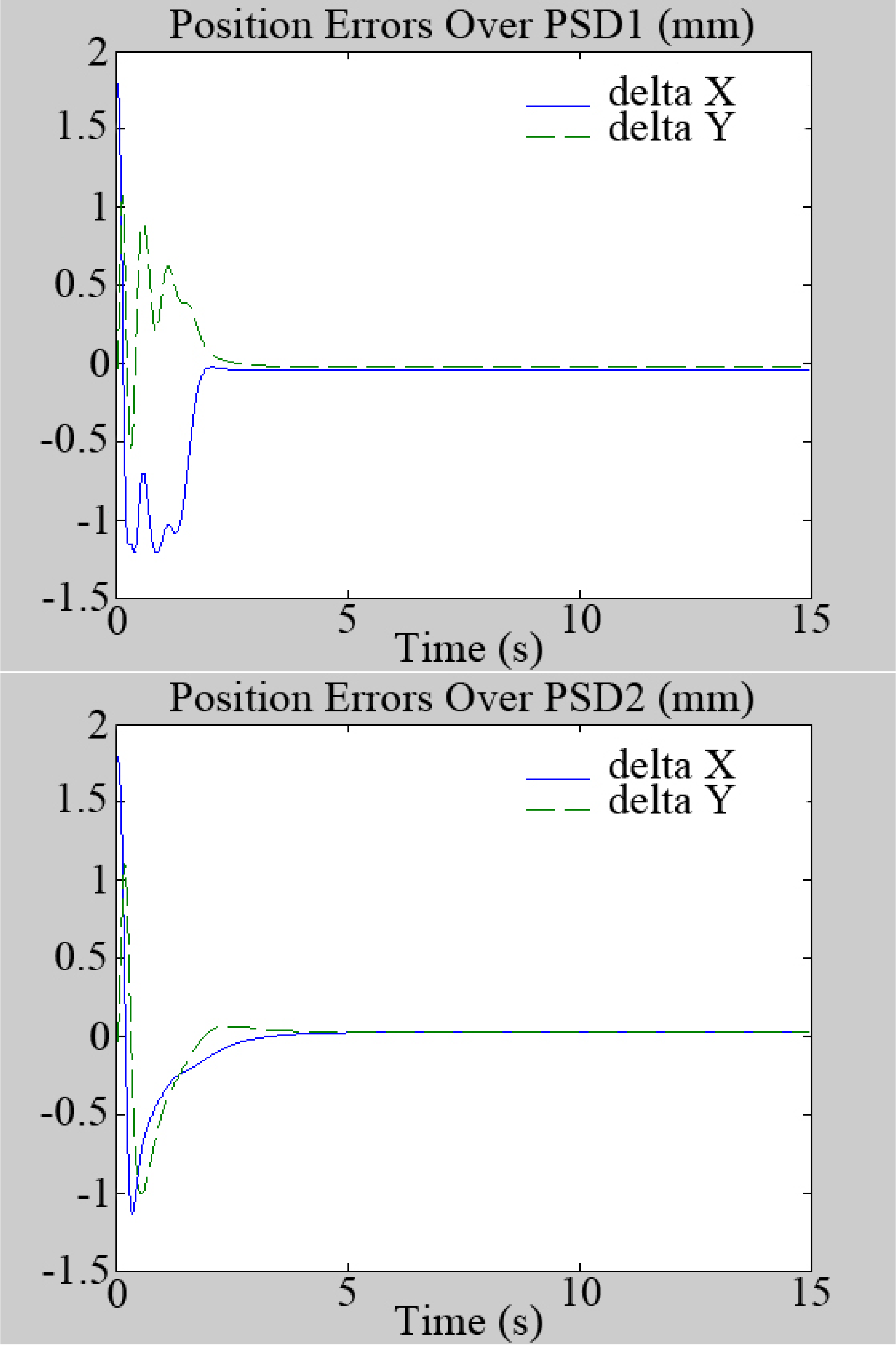


Figure 5.7: Robot control simulation results.

5.2.2 Experimental Results

One of the most crucial designs for the calibration system to work is the ability to successfully align the laser beam at the center of both PSDs sensors at the same time. This will ensure the compatibility of our systematic virtual linear constraint method as well as ensuring the accuracy on the solution. All three stages of the control system were successfully tested and implemented using the PSCD and the IRB 120 robot. First, servo control was able to align the laser into the active area of PSD1, as well as accurately compute the laser line length (determined to be 494.269 mm for these experiments). Figure 5.8 shows the image features of the two LEDs along with the features of the laser spot. The IBVS is completed in less than 10 seconds, and Figure 5.9 shows and demonstrates the convergence between the center of the two LEDs and the laser spot feature. Since the laser is fixed along with the camera due to the triangulation setup, the LEDs features actually move to the features of the laser spot. the sampling time during the servoing was found to be up to $100ms$. The experimental results found for the first stage using IBVS control clearly demonstrates that the laser beam can be guided towards the active area of the PSD quickly and effectively.

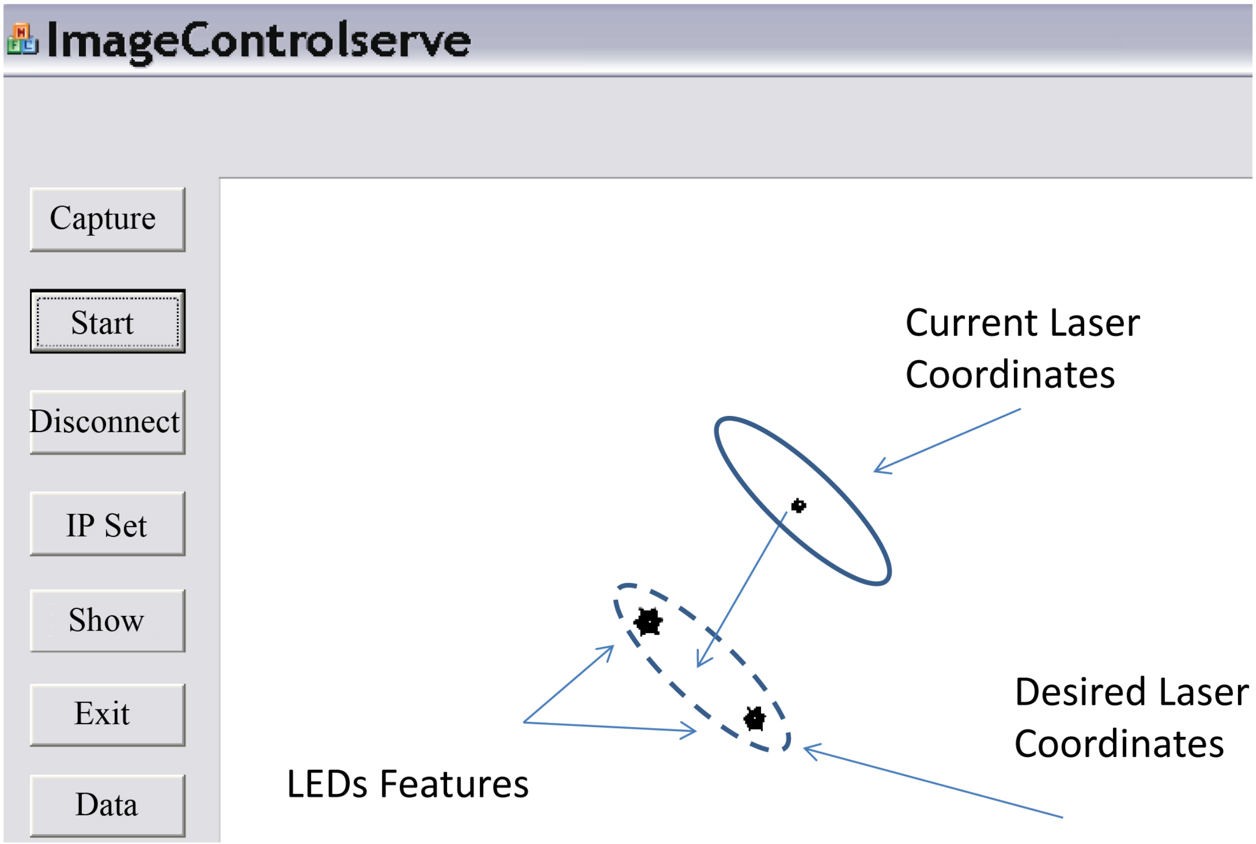


Figure 5.8: Image features before IBVS control.

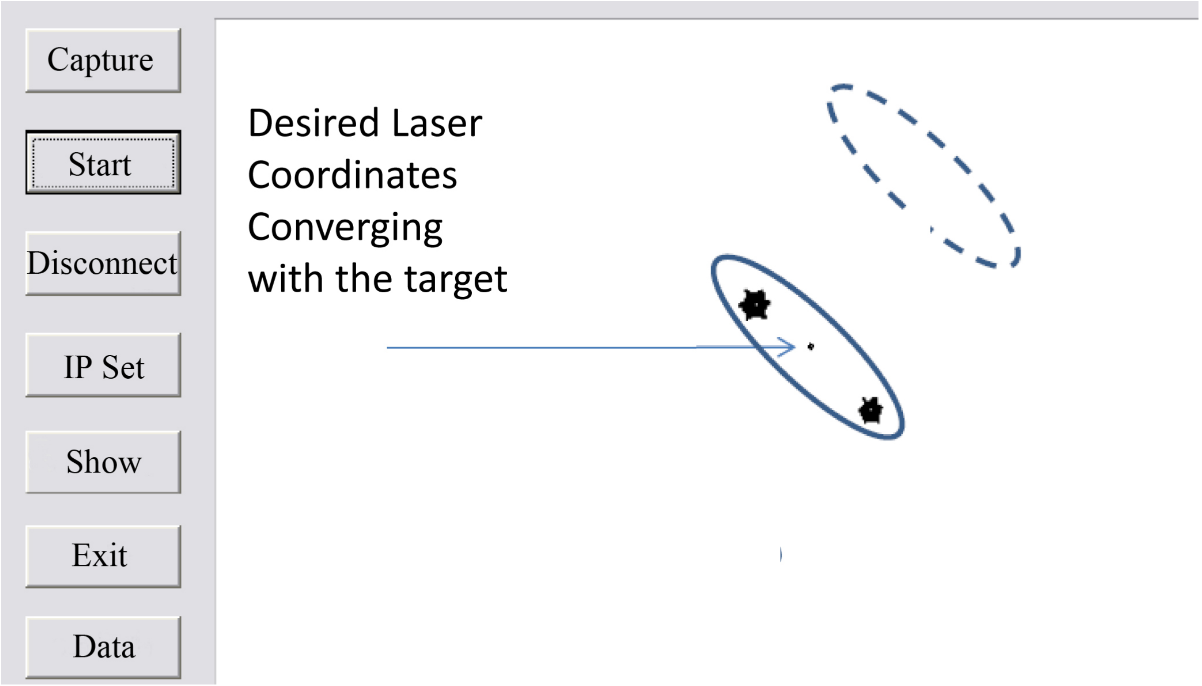


Figure 5.9: Image features after IBVS control.

In stage two, the PSD-based controller takes over and proceeds to move the laser spot accurately into the center of PSD1. This process will take approximately 15 seconds. For the third stage, using the laser line length found during the servo control process, the kinematic model of the robot is modified such that orientation control can be performed around the point found in the center of PSD1. Then using orientation control, we were able to not only find accurately the center of PSD2, but also able to maintain the original position in PSD1. The exact same process will be repeated three more times (for positions 2, 3 and 4) to complete the calibration process. Figure 5.10 presents the GUI image of the beam trace of an experiment performed. This figure basically presents how the PSD-based servo control was able to reach the center of the PSD1, below in the figure, and also able to reach the center of PSD2 afterwards without changing its position in PSD1. The results achieved by these experiments essentially verified the feasibility of the control system proposed and the eventual automation of the entire calibration system.

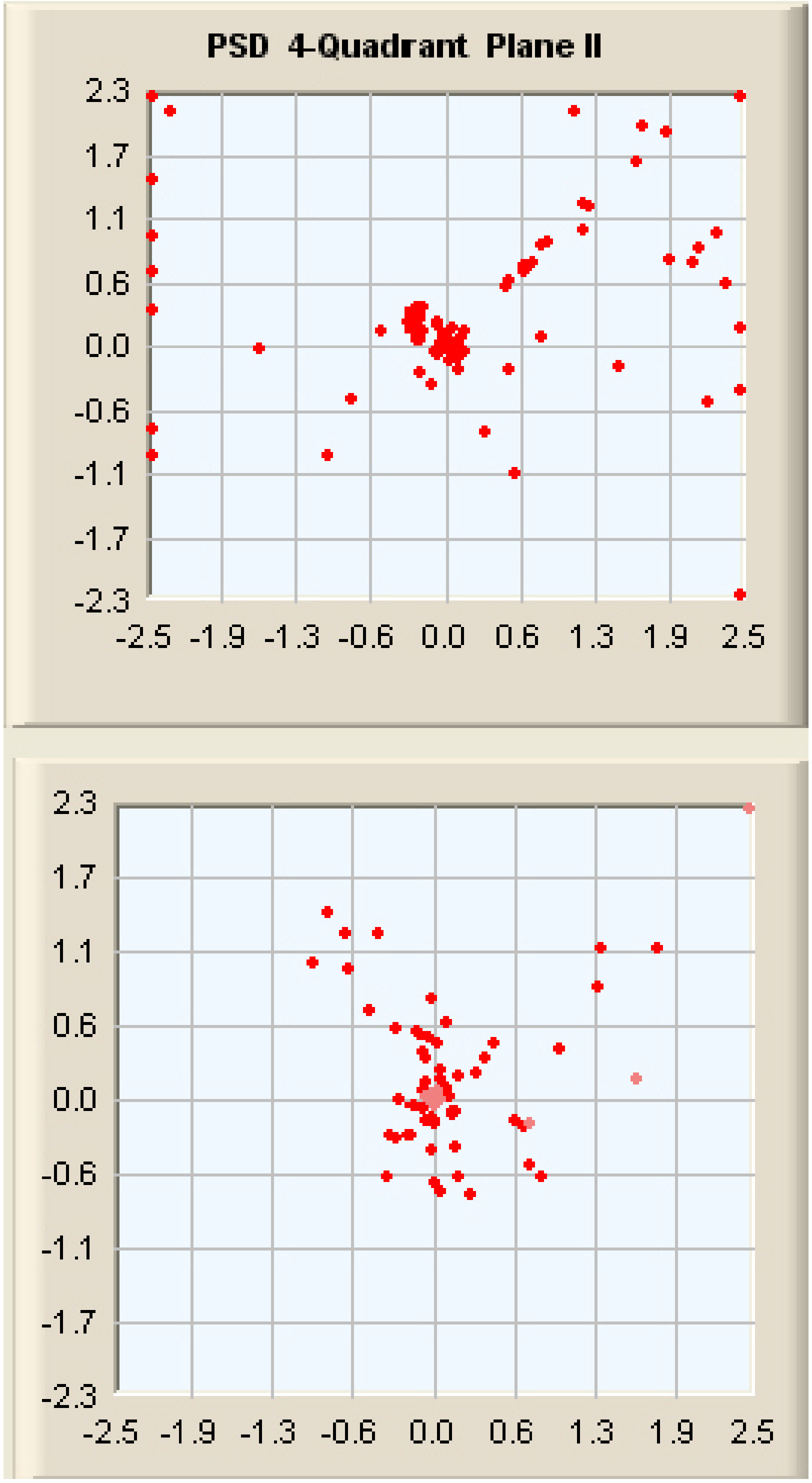


Figure 5.10: Robot controller results after all stages were completed.

Chapter 6

Calibration System & Algorithms

No matter what principles they are based on, such as optical, mechanical, etc., all calibration systems have the same basic idea in common: An object or point in the robot's task space with known properties (i.e. position, orientation, size etc.) is used to perform a calibration of the robot system. This is achieved by comparing the known properties with the data measured for this object or point by the robot. The data considered for this depends on the specific kind of system used. A camera system for example, could be used to measure the size and position of a certain known object in the camera's field of view. This can then be used to compare the computed position (extracted from the captured image) with the robot's position given by the joint encoder values and forward kinematics. Based on this comparison the deviation of the measured position from its intended position can be determined. If this is done for several points, the pose error of the robot can be determined. By modifying the robot controller software one is then able to compensate the error and achieve better positional accuracy in the future. In this chapter, the calibration algorithms will be developed and discuss as well as simulation and experimental results.

6.1 Analysis of the Kinematics Error Model

Kinematic modeling of the robot manipulator is one of the most important steps towards a successful calibration process. A kinematic model essentially relates the output of the robot joint sensors to the position and orientation of the robot's TCP. It is precisely the main goal of robot calibration, to improve the accuracy of the robot by modifying the kinematic parameters predefined in the robot controller. Kinematic modeling can also be extended to include tools attached to the TCP, and even the entire workcell. In that case, the relative positions and orientations of each system components are assigned in a unique way. A widely used and convenient mathematical approach for describing geometric relationships among those components is called homogeneous transformations. The modeling starts by setting a reference coordinate system to relate the other subsystems. Then, local coordinate frames are assigned to moving components. In the case of robotic systems, generally this frames are assigned to the joints of the robot between links. Finally, a set of homogeneous transformation matrices is assigned to relate each link coordinate system to its neighbor's coordinate frame.

As stated by researchers, a kinematic model suitable for robot calibration must be complete and proportional. A proportional kinematic model essentially implies no singularities. Basically, a kinematic model is proportional if continuous changes in position and orientation of any joint axis result in continuous changes in the model link parameters. Lack of proportionality in the model may invalidate linearized accuracy error model and may also lead to instabilities in the kinematic identification process. A complete kinematic model refers to the capability of relating the joint displacements to the TCP configuration for any robot geometry, while allowing for the arbitrary placement of the world frame.

The Denavit-Hhartenberg [36] is a commonly used convention to represent frame references in the forward kinematic model of a robot manipulator. For the robot model with n joints (n=6), the frame references can be represented as shown in Figure 6.1.

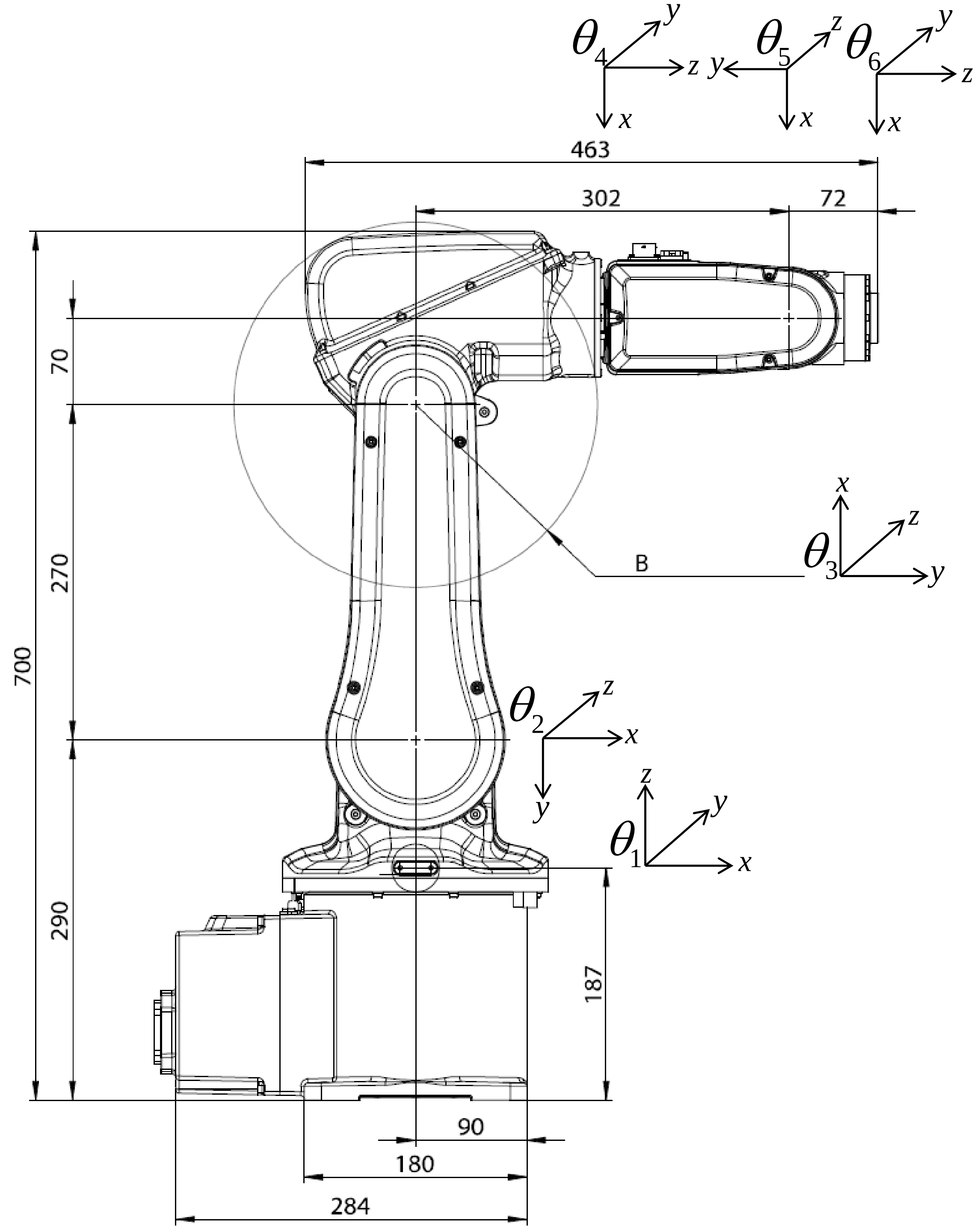


Figure 6.1: The D-H model used for the ABB IRB120 kinematics

The forward kinematics model can be represented by the following equation,

$$^{B}T_{E} = \prod_{i=1}^{n} A_i \tag{6.1}$$

where $^{B}T_{E}$ is the transformation matrix that expresses the position and orientation of the robot TCP frame {E} with respect to the robot base frame {B}; A_i is the homogeneous transformation matrix associated with link i and joint i .

By Denavit-Hartenberg (D-H) model, each homogeneous transformation matrix A_i can be written as,

$$A_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.2}$$

where the four quantities θ_i , a_i, d_i and α_i, denotes the joint angle, link length, link offset and link twist respectively. They are all parameters associated with link i and joint i . $c\theta_i$ denotes $cos(\theta_i)$ while $s(\theta_i)$ denotes $sin(\theta_i)$.

We can also use the same model to include the joint offset of each link as follows; let δ_i denotes the offset value of the joint i , then each homogeneous transformation can be rewritten as,

$$\tilde{A}_i = \begin{bmatrix} c\tilde{\theta}_i & -s\tilde{\theta}_i c\alpha_i & s\tilde{\theta}_i s\alpha_i & a_i c\tilde{\theta}_i \\ s\tilde{\theta}_i & c\tilde{\theta}_i c\alpha_i & -c\tilde{\theta}_i s\alpha_i & a_i s\tilde{\theta}_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.3}$$

where $c\tilde{\theta}_i$ denotes $cos(\tilde{\theta}_i + \tilde{\delta}_i)$ and $s\tilde{\theta}_i$ denotes $sin(\tilde{\theta}_i + \tilde{\delta}_i)$. Combining the joint offset and substituting (6.3) into (6.1), forward kinematics with the offset is written as,

$$^{B}T_E = \prod_{i=1}^{6} \tilde{A}_i = \tilde{A}_1\tilde{A}_2\tilde{A}_3\tilde{A}_4\tilde{A}_5\tilde{A}_6 = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.4}$$

Note that joint 1 depends on the robot base frame. So in (6.4) there are five unknown parameters, which are the last five offsets $\delta_i (i = 2, 3, 4, 5, 6)$.

6.2 Calibration Algorithms

This section presents the computational aspect and derivations of the calibration algorithms used to compute the parameter errors. Both joint offset calibration and workpiece frame calibration techniques are discussed and presented.

6.2.1 Joint Offset Calibration

The proposed calibration process, shown in Figure 3.2 at Chapter 3, is performed by locating the TCP and the laser pointer several times at different locations. However, no matter where the laser pointer is, it should always aim at the center of one PSD and be reflected off the PSD surface with a direction towards the center of the other PSD. Hence, four sets of robot joint angles can be collected and recorded by the robot controller.

Because the laser pointer is rigidly fixed on the robot TCP, we denote the j^{th} position of the laser pointer as $P_j = [\ P_{xj} \quad P_{yj} \quad P_{zj}\ ]^T$ $(j = 1, 2, 3, 4)$ and the unit directional vector of the laser line as, $[\ m_j \quad n_j \quad p_j\ ]^T$. Replacing the recorded joint angles into the forward kinematic equation (6.4), we can decompose them as follows,

$$\begin{bmatrix} m_j \\ n_j \\ p_j \\ 0 \end{bmatrix} = ({}^{B}T_E)_j \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} T_{11} \\ T_{21} \\ T_{31} \\ 0 \end{bmatrix}_j, j = 1, 2, 3, 4 \tag{6.5}$$

and,

$$\begin{bmatrix} P_{xj} \\ P_{yj} \\ P_{zj} \\ 0 \end{bmatrix} = ({}^{B}T_E)_j \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} T_{14} \\ T_{24} \\ T_{34} \\ 1 \end{bmatrix}_j , j = 1, 2, 3, 4 \tag{6.6}$$

Because P_1 and P_2 are in the same laser line 1, they have following relationships,

$$\begin{aligned} m_1 &= m_2 \\ n_1 &= n_2 \\ p_1 &= p_2 \end{aligned} \tag{6.7}$$

and,

$$\frac{P_{x1} - P_{x2}}{m_1} = \frac{P_{y1} - P_{y2}}{n_1} = \frac{P_{z1} - P_{z2}}{p_1} \tag{6.8}$$

Hence, for these two positions of the laser pointer, we have a squared error.

$$\psi_1 = \psi_{11}^2 + \psi_{12}^2 + \psi_{13}^2 + \psi_{14}^2 \tag{6.9}$$

where

$$\begin{aligned} \psi_{11} &= (T_{11})_2 - (T_{11})_1 \\ \psi_{12} &= (T_{21})_2 - (T_{21})_1 \\ \psi_{13} &= \frac{(T_{11})_1}{(T_{21})_1} - \frac{(T_{14})_2 - (T_{14})_1}{(T_{24})_2 - (T_{24})_1} \\ \psi_{14} &= \frac{(T_{21})_1}{(T_{31})_1} - \frac{(T_{24})_2 - (T_{24})_1}{(T_{34})_2 - (T_{34})_1} \end{aligned} \tag{6.10}$$

Similarly, consider P_3 and P_4 are in the same laser line 2, they have the same relations. Then we can get another squared error.

$$\psi_2 = \psi_{21}^2 + \psi_{22}^2 + \psi_{23}^2 + \psi_{24}^2 \tag{6.11}$$

where

$$\begin{aligned} \psi_{21} &= (T_{11})_4 - (T_{11})_3 \\ \psi_{22} &= (T_{21})_4 - (T_{21})_3 \\ \psi_{23} &= \frac{(T_{11})_3}{(T_{21})_3} - \frac{(T_{14})_4 - (T_{14})_3}{(T_{24})_4 - (T_{24})_3} \\ \psi_{24} &= \frac{(T_{21})_3}{(T_{31})_3} - \frac{(T_{24})_4 - (T_{24})_3}{(T_{34})_4 - (T_{34})_3} \end{aligned} \tag{6.12}$$

Hence the unknown parameters, i.e. the last offsets $\delta_i (i = 2, 3, 4, 5, 6)$, are found by minimizing the total sum of the squared errors.

$$\Psi = argmin \sum_{k=1}^{2} \psi_k \tag{6.13}$$

Because of the complexity of the optimization problem with 5 parameters, the well-known Levenberg-Marquardt algorithm (LMA)[10, 11] will be used to minimize the sum of squared errors.

6.2.2 Robot Workpiece Frame Calibration

With the joint offset calculations ahead, the $({}^{B}T_{E})_j$ $(j = 1, 2, 3, 4)$ can also be found. Let R and t be the rotation matrix and the translation vector of ${}^{B}T_{D}$, respectively. Denote L_D and L_B as direction vectors of the laser beam with respect to the $\{D\}$ and the $\{B\}$ frames, respectively. Considering the relationship between L_D and L_B , the following equation can be found,

$$L_B = RL_D = qL_Dq^* \tag{6.14}$$

where

$$q = cos\left(\frac{\theta}{2}\right) + sin\left(\frac{\theta}{2}\right)\vec{k} \tag{6.15}$$

Substituting (6.15) into (6.14), we have;

$$L_D - L_B = tan\left(\frac{\theta}{2}\right)(L_D + L_B) \times \vec{k} \tag{6.16}$$

Let $\Omega(v)$ denote a skew symmetric matrix, therefore (6.16) can be written as;

$$\Omega(v)\vec{g} = b \tag{6.17}$$

where $L_D + L_B = a$,$L_D - L_B = b$ and $\vec{g} = tan\,(\theta/2)\,\vec{k}$. A minimum of two laser beams are needed. Let $C = [\ \Omega(a_1) \quad \Omega(a_2)\ ]^T$, then (6.17) can be express as;

$$C\vec{g} = D \tag{6.18}$$

From (6.18), the vector $\vec{g}$ can be solved by using least squared method. $\vec{k}$ and θ can be computed by;

$$\vec{k} = \frac{\vec{g}}{\|\vec{g}\|} \tag{6.19}$$

$$\theta = 2 \times atan\left(\frac{\vec{g}_{max}}{\vec{k}_{max}}\right) \tag{6.20}$$

where $\vec{g}_{max}$ and $\vec{k}_{max}$ represent the vectors maximum values. Hence the rotation matrix R can be computed as follows,

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{6.21}$$

where,

$$\begin{gathered} R_{11} = k_x^2(1 - cos(\theta)) + cos(\theta) \\ R_{12} = k_x k_y(1 - cos(\theta)) - k_z sin(\theta) \\ R_{13} = k_z k_x(1 - cos(\theta)) + k_y sin(\theta) \\ R_{21} = k_x k_y(1 - cos(\theta)) + k_z sin(\theta) \\ R_{22} = k_y^2(1 - cos(\theta)) + cos(\theta) \\ R_{23} = k_y k_z(1 - cos(\theta)) - k_x sin(\theta) \\ R_{31} = k_z k_x(1 - cos(\theta)) - k_y sin(\theta) \\ R_{32} = k_y k_z(1 - cos(\theta)) + k_x sin(\theta) \\ R_{33} = k_z^2(1 - cos(\theta)) + cos(\theta) \end{gathered} \tag{6.22}$$

For each laser beam, we have;

$$\begin{bmatrix} x_{1Bk} \\ y_{1Bk} \\ z_{1Bk} \end{bmatrix} = \begin{bmatrix} x_{0Bk} \\ y_{0Bk} \\ z_{0Bk} \end{bmatrix} + \lambda_k \begin{bmatrix} m_{Bk} \\ n_{Bk} \\ p_{Bk} \end{bmatrix}, k = 1, 2 \tag{6.23}$$

Then;

$$\begin{bmatrix} x_{1Bk} \\ y_{1Bk} \\ z_{1Bk} \end{bmatrix} = R \begin{bmatrix} x_{0Dk} \\ y_{0Dk} \\ z_{0Dk} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \lambda_k \begin{bmatrix} m_{Bk} \\ n_{Bk} \\ p_{Bk} \end{bmatrix}, k = 1, 2 \tag{6.24}$$

where $[\ x_{0Dk} \quad y_{0Dk} \quad z_{0Dk}\ ]^T$ denotes the center point of the k^{th} PSD with respect to $\{D\}$; $[\ x_{0Bk} \quad y_{0Bk} \quad z_{0Bk}\ ]^T$ denotes the center point of the k^{th} PSD with respect to $\{B\}$; $[\ x_{1Bk} \quad y_{1Bk} \quad z_{1Bk}\ ]^T$ denotes the k^{th} TCP position with respect to the $\{B\}$, which can be obtained by the robot controller. Let $[\ a_{xk} \quad a_{yk} \quad a_{zk}\ ]^T = R[\ x_{0Dk} \quad y_{0Dk} \quad z_{0Dk}\ ]^T$, then the following relations can be obtain,

$$\frac{x_{1Bk} - a_{xk} - t_x}{M_{Bk}} = \frac{y_{1Bk} - a_{yk} - t_y}{N_{Bk}} = \frac{z_{1Bk} - a_{zk} - t_z}{p_{Bk}} \tag{6.25}$$

In these relationships, the values of $[\ t_x \quad t_y \quad t_z\ ]^T$ are unknown and can be computed by,

$$\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} \dfrac{(z_{1B2} - z_{1B1}) - (a_{z2} - a_{z1}) - \dfrac{P_{B2}}{M_{B2}}(x_{1B2} - a_{x2}) + \dfrac{P_{B1}}{M_{B1}}(x_{1B1} - a_{x1})}{\dfrac{P_{B1}}{M_{B1}} - \dfrac{P_{B2}}{M_{B2}}} \\ \dfrac{(x_{1B2} - x_{1B1}) - (a_{x2} - a_{x1}) - \dfrac{M_{B2}}{N_{B2}}(y_{1B2} - a_{y2}) + \dfrac{M_{B1}}{N_{B1}}(y_{1B1} - a_{y1})}{\dfrac{M_{B1}}{N_{B1}} - \dfrac{M_{B2}}{N_{B2}}} \\ \dfrac{(y_{1B2} - y_{1B1}) - (a_{y2} - a_{y1}) - \dfrac{N_{B2}}{P_{B2}}(z_{1B2} - a_{z2}) + \dfrac{N_{B1}}{p_{B1}}(z_{1B1} - a_{z1})}{\dfrac{N_{B1}}{P_{B1}} - \dfrac{N_{B2}}{P_{B2}}} \end{bmatrix} \tag{6.26}$$

Therefore the calibration matrix will be given by;

$${}^{B}T_{D} = \begin{bmatrix} R_{3\times3} & t_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{6.27}$$

6.3 Simulation and Experimental Results

A 3-D computer model of the IRB120 was built in the computer using Matlab. The DH parameters of the IRB 120 robot manipulator were derived as shown in Table 6.1. In the simulations the laser pointer was attached on the TCP to align the laser line with the X-axis of the robot TCP frame, the same way we did for the experimental design. A virtual PSD device was also built and fixed, so that the algorithm could find the robot configuration needed to align the laser line to hit the center of both PSDs at 4 different positions. Experimental data was also collected, analyzed and compared to our previous method of robot calibration.

Table 6.1: DH Parameters of the ABB IRB120 manipulator

Joint	a(mm)	α(deg)	d(mm)	θ(deg)
1	0	-90	290	0
2	270	0	0	-90
3	0	90	0	180
4	70	-90	302	0
5	0	90	0	0
6	0	0	72	0

6.3.1 Simulation of joint Offset Calibration

Simulations of the joint offset calibration were performed using initial joint offset parameters, such that we can identify them using our proposed method. In other words, we introduce offset values for each joint in the kinematic model and compare them with the values our proposed method was able to compute. The results of the calibration simulations are shown in Table 6.2. The first column holds the joint offset number while column two holds the actual offset parameters used to initialize the simulations. In the third column the initial parameters used by the optimization algorithm (LMA) are shown. In the fourth column the solution of the algorithm is shown while the fifth column shows the error between them. The results shows perfect results, therefore they confirmed the efficiency of the new proposed method.

Table 6.2: Simulations results on joint offset calibration

Parameter (deg)	Actual Value	Initial Value	Simulation Results	Error
δ_2	1.20	0.0	1.2000	0.0000
δ_3	0.80	0.0	0.8000	0.0000
δ_4	-1.40	0.0	-1.4000	0.0000
δ_5	-0.60	0.0	-0.6000	0.0000
δ_6	-1.00	0.0	-1.0000	0.0000

6.3.2 Simulation of Workpiece Frame Calibration

Similarly, simulations of the workpiece frame calibration were performed. This time we initialize the relationship between the base and the workpiece frame, such that we can identify each parameter using our proposed method. The results are shown in Table 6.3. In the second column the actual parameter value used to initialize the simulation are presented while in the third column we shows the initial parameters for the algorithm (LMA). The fourth column shows the solution of the algorithm and in the fifth column the error between them were presented. Again we obtained perfect results therefore we can also claim it also confirmed the efficiency of the new method.

Table 6.3: Simulations results on workpiece frame calibration

Parameter (deg)	Actual Value	Initial Value	Simulation Results	Error
R_{11}	0.4293	0.0	0.4293	0.0000
R_{12}	-0.8779	0.0	-0.8779	0.0000
R_{13}	-0.2120	0.0	-0.2120	0.0000
R_{21}	0.8738	0.0	0.8738	0.0000
R_{22}	0.4632	0.0	0.4632	0.0000
R_{23}	-0.1485	0.0	-0.1485	0.0000
R_{31}	0.2285	0.0	0.2285	0.0000
R_{32}	-0.1215	0.0	-0.1215	0.0000
R_{33}	0.9659	0.0	0.9659	0.0000
t_x	240.0000	0.0	240.0000	0.0000
t_y	160.0000	0.0	160.0000	0.0000
t_z	180.0000	0.0	180.0000	0.0000

6.3.3 Experimental Analysis

To prove our proposed method works, we perform real experiments on the ABB robot IRB120. Due to the fact that we have not fully developed the control system for the robot to move to the center of both PSDs, we perform the experiments manually as shown in Figure 6.2. Also in this figure the right corners, shows how we move the robot manually such that it aligns the laser line to both PSDs centers at position 1 and 2. Similarly, on the left corners, shows how we moved the robot for position 3 and 4. Once we achieve the desired positions, the joint parameters of the robot were recorded from the robot controller and eventually processed into our proposed calibration method. The experimental results are presented in Table 6.4 for the joint offset and were compared with our previous method for robot calibration. The first column shows the joint offset number while the second column shows the actual offset parameters which are unknown in the experiments. The third column shows the initial parameters of the LMA for both experiments. The fourth column shows the results of the optimization for our previous calibration method and in the fifth column we show the results for the method proposed in this paper. The results show similar values in both methods therefore we can also claim that using our new dual PSD method we can achieve a similar level of accuracy of joint offset calibration, even under manual guidance.

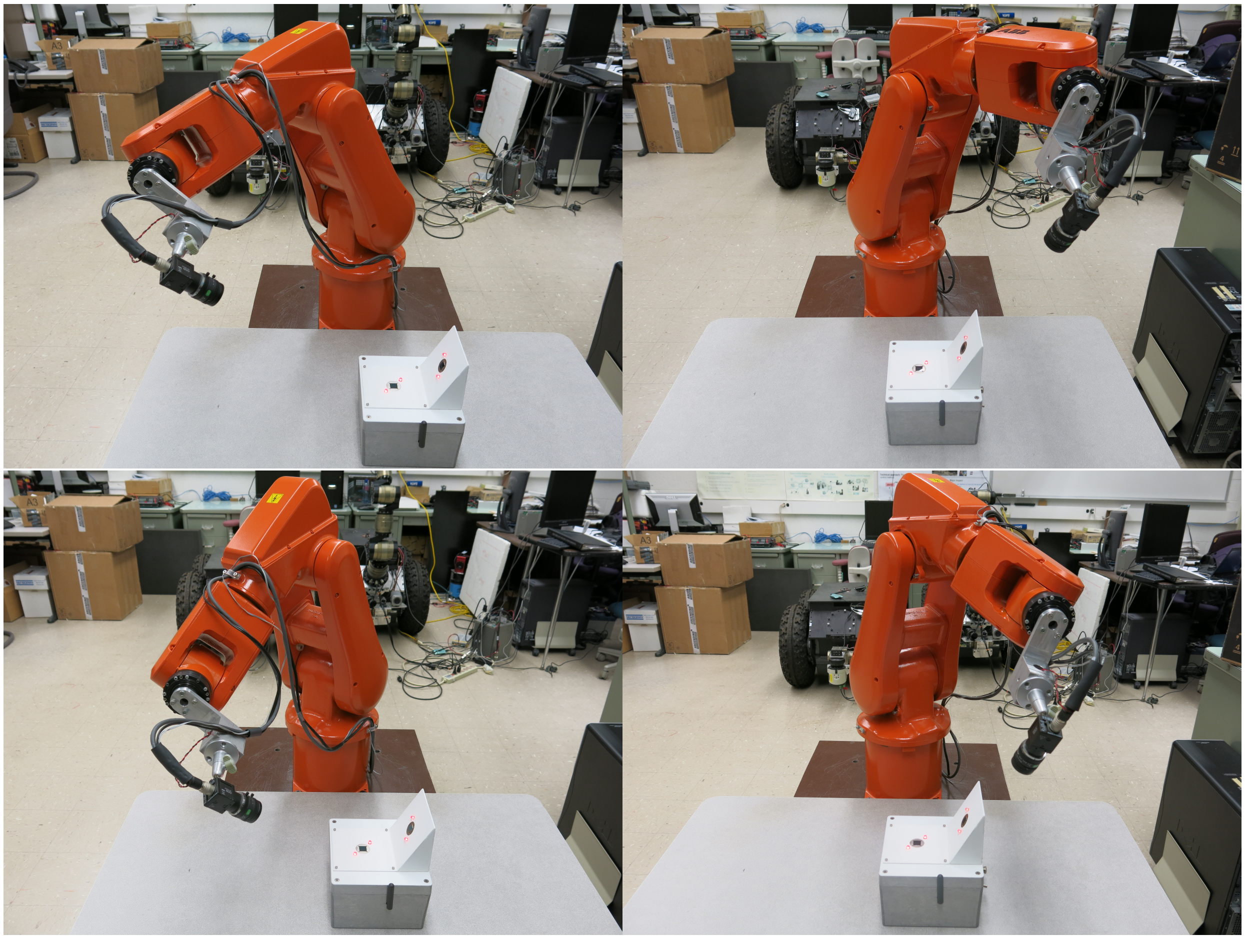


Figure 6.2: Robot pose and location at positions 1,2,3 and 4.

Table 6.4 essentially prove not only that both method have similar level of accuracy of joint offset calibration, but also that they are feasible. Among the advantages of the new proposed method is the ability to perform the whole process using only 4 sets of position data instead of 7 needed for the previous method. Therefore the new approach is simpler and faster.

Table 6.4: Simulations results on joint offset calibration

Parameter (deg)	Actual Value	Initial Value	Single PSD Method	Dual PSD Method
δ_2	unknown	0.0	1.9240	2.2163
δ_3	unknown	0.0	0.5800	0.8722
δ_4	unknown	0.0	-0.1831	-0.1596
δ_5	unknown	0.0	0.0973	-0.2107
δ_6	unknown	0.0	-0.2453	-0.6281

Chapter 7

Experimental Calibration Results

This chapter describes the methodology used to perform the calibration experiments using an ABB IRB 120 robotic arm. The main goal is to compare the speed, performance and reliability of our calibration system. Therefore, the calibration results obtained from the factory calibration system will be used as a reference for the unknown real values. To compare speed, performance and reliability, we also perform experiments using the single PSD calibration approach previously developed and tested. The idea behind this is to be able to compare both methods and verify the theoretical advantages of the proposed dual calibration system explained before while having a reliable reference for the solution. Finally, experimental results will be presented and discussed demonstrating the feasibility of the overall calibration system including device hardware, software and calibration algorithms.

7.1 Experimental Methodology

The factory motor offset values are the results found by ABB using their current calibration system which were used to calibrate the robot just before leaving the factory. Although, these values do not represent the exact real values, they must be very close to the actual real offset values. This is due to the fact that calibration has not been performed since assembly on July 2011. Hence, we can use them as an approximate reference for the real actual values. Table 7.1 includes both the motor offset values found by ABB in the factory, and the corresponding gear ratios for each robot joint for the ABB IRB 120 robot. Gear ratios are used to translate the joint offset found by the calibration system into motor encoder values. Those values typically vary from 0.0 to 6.28 to comprise an entire motor revolution (2π). Depending on the brand, type and model of the robot manipulator, the gear ratios for each joint will be different and sometimes unique for each robot manipulator.

Table 7.1: Factory motor offset values & gear ratios

Parameter	Motor Offset (rads)	Gear Ratio
δ_1	1.01524	121
δ_2	0.29250801	121
δ_3	0.89234102	101
δ_4	1.4513	50
δ_5	1.70252	51
δ_6	4.5975299	50

For both cases, the calibration device was placed arbitrarily on a table flat and parallel to the base frame plane of the robot. Figure 7.1 shows this placement in the left for the single PSD approach and in the right for the proposed dual PSD approach respectively.

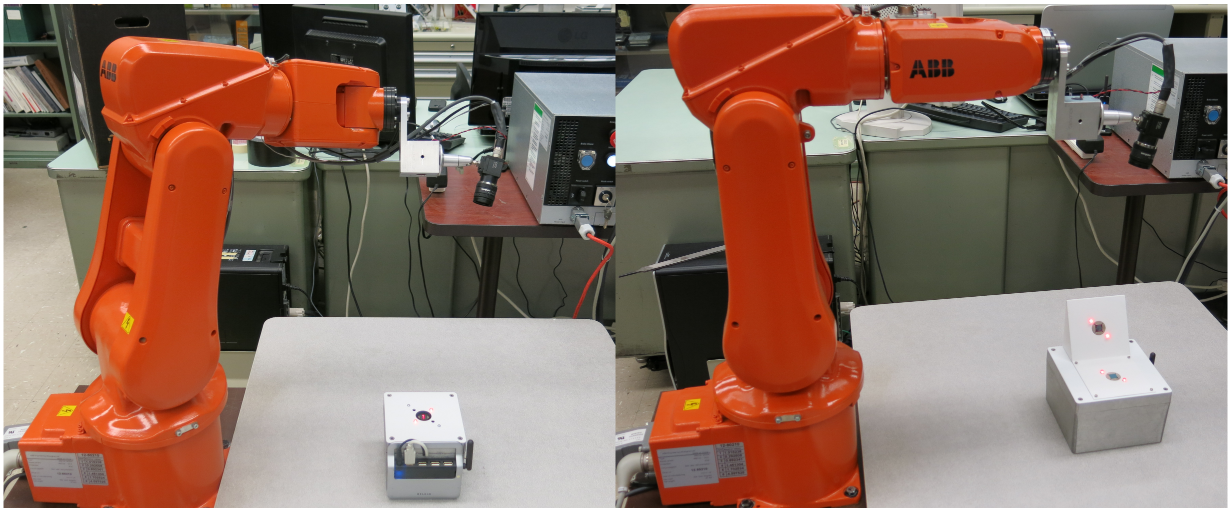


Figure 7.1: PSCD arbitrary placement

After the arbitrary placement of the device, we introduce an artificial offset value of π for each motor joint offset in the robot's controller instead of the initial motor offset values from Table 7.1. Next, experiments were performed for each method according to their respective calibration process. Once we obtain an offset solution, the results are compared to the reference factory motor offset values. For the case of the single PSD approach, a set of 3 experiments were performed and an average value was taken. This is because a relatively large standard deviation between offset values was previously found in the past after several experiments were performed with the same conditions. Once the solution is obtained from the calibration system, we proceed to compensate the offset value in the robot controller as follows:

$$\delta_i = \delta_o + \delta_n, i = 2, 3, 4, 5, 6 \tag{7.1}$$

Where δ_i is the new motor offset value, δ_o is the value from factory, and δ_n is the value found by our calibration system, all values in radians.

Typically one experiment alone will not be enough to achieve a high level of accuracy in the solution. Nevertheless, if we run the experiment multiple times compensating each time the joint offset, a higher level of accuracy in the solution can be achieve, similar to current reliable calibration systems in the market. Although, it is not necessary, a total of 6 experiments were performed using both methods in order to show the stability in finding the solution.

7.2 Single PSD Calibration Results

Using our single PSD calibration approach, the factory motor offset values were changed to a nominal value π in all joints initially. The idea was to be able to compute similar values to those found by the factory calibration system. Then, we run the calibration process for a total of 3 times. Afterwards, the average of joints offset found by the calibration system was taken and added to the original value of joint offset. For instance, in joint 4, let the average value found to be -1.843125 degrees, the initial value for that joint offset (introduced artificial value) is π radians. Therefore the new motor offset value should be as follows;

$$\delta_4 = \pi + (-1.843125) * \frac{\pi}{180} * 50 = 1.533162665 \tag{7.2}$$

Below are the results of 6 different sets (of 3 experiments each): For the first set shown in Table 7.2, we delete the factory motor offset values and introduce the value π into each motor side offset in the controller and run experiments to see the results:

Table 7.2: Experiment set 1

Joint	Experiment 1	Experiment 2	Experiment 3	Average	STD
δ_2	-1.144901	-1.271305	-1.177991	-1.198065667	0.0655495
δ_3	-1.26074	-1.26074	-1.26074	-1.26074	0.0
δ_4	-1.844231	-1.834079	-1.851065	-1.843125	0.00854684
δ_5	-1.963385	-1.954932	-1.955745333	-1.955745333	0.007267216
δ_6	1.899297	1.9057	1.903369667	1.903369667	0.003539265

Table 7.3 below shows the average offset values in the encoder motor side using equation 7.1:

Table 7.3: Offset values found by the calibration system

Joint	Offset found (deg)	Offset motor side (rads)
δ_2	-1.198065667	-2.530133055
δ_3	-1.26074	-2.222410465
δ_4	-1.843125	-1.608429989
δ_5	-1.955745333	-1.740843965
δ_6	1.903369667	1.661003378

Then, the average was use to compensate the offsets found by the single PSD calibration system. For instance,

$$\delta_2 = 3.141592654 + (-2.530133055) = 0.611459598 \tag{7.3}$$

where δ_2 is the new motor offset value for joint 2, 3.141592654 is the initial value, and -2.530133055 is the value found by our calibration system in radians. This is shown in Table 7.4 below as new motor side offset. This new motor side offset will then be used to be the new initial motor side offset for the next set of experiments.

Table 7.4: New motor side offset values

Joint	Initial motor side offset	New motor side offset
δ_2	3.141592654	0.611459598
δ_3	3.141592654	0.919182188
δ_4	3.141592654	1.533162665
δ_5	3.141592654	1.400748688
δ_6	3.141592654	4.802596032

Experiment set 2: at this point the value obtained above was added to the controller and a new set of experiments were conduct, as shown in Table 7.5 below:

Table 7.5: Experiment set 2

Joint	Experiment 1	Experiment 2	Experiment 3	Average	STD
δ_2	-0.26543	-0.180342	-0.197923	-0.214565	0.044918916
δ_3	-0.05583	0.001643	-0.00997	-0.021385667	0.030389546
δ_4	-0.079641	-0.080887	-0.082043	-0.080857	0.001201281
δ_5	0.269785	0.25727	0.227005	0.251353333	0.021995166
δ_6	-0.180412	-0.205178	-0.226991	-0.204193667	0.023305096

Once again, Table 7.6 below shows the average offset values in the motor side for the experiment set 2:

Table 7.6: Offset values found by the calibration system

Joint	Offset found (deg)	Offset motor side (rads)
δ_2	-1.198065667	-2.530133055
δ_3	-1.26074	-2.222410465
δ_4	-1.843125	-1.608429989
δ_5	-1.955745333	-1.740843965
δ_6	1.903369667	1.661003378

The average was use once again to compensate the offsets found by the calibration system and the values are shown in Table 7.7 below:

Table 7.7: New motor side offset values

Joint	Initial motor side offset	New motor side offset
δ_2	0.611459598	0.158330848
δ_3	0.919182188	0.881483909
δ_4	1.533162665	1.462601621
δ_5	1.400748688	1.624482794
δ_6	4.802596032	4.624403442

The same process was repeated for sets 3, 4, 5, and 6 and the results are summarized in Table 7.8 Below:

Table 7.8: Experiment summary using the single PSD calibration

Joint	Initial	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6
δ_2	3.14159	0.61146	0.158331	0.272266	0.214393	0.351835	0.34578
δ_3	3.14159	0.919182	0.881484	0.755835	0.753175	0.768786	0.71828
δ_4	3.14159	1.533163	1.462602	1.465335	1.456379	1.470641	1.46219
δ_5	3.14159	1.400749	1.624483	1.643717	1.575292	1.57465	1.64003
δ_6	3.14159	4.802596	4.624403	4.666855	4.657897	4.733736	4.72974

7.2.1 Remarks & Discussions

The experiments performed using the single PSD calibration system essentially verified the feasibility of the system and the ability to identify the joint offsets with some limitations.

Beyond experiment sets 4, 5, and 6 we notice that the calibration results start to oscillate on 0.0 deg offsets. Nevertheless, is at this point where we can analyze how reliable this calibration system is by calculating the percentage of errors as shown in Table 7.9.

Table 7.9: Error percentage using single PSD calibration

Joint	Encoder Errors	Joint Errors	Encoder Errors %	Joint Errors %
δ_2	0.059327	0.028093	0.944223024	0.007803496
δ_3	0.174061	0.098742	2.770273148	0.027428447
δ_4	0.019341	0.022164	0.30782922	0.006156584
δ_5	0.12787	0.143655	2.035116816	0.039904251
δ_6	0.060367	0.069175	0.960765892	0.019215318

The plot comparing the newly found offsets with the factory offsets is shown in Figure 7.2.

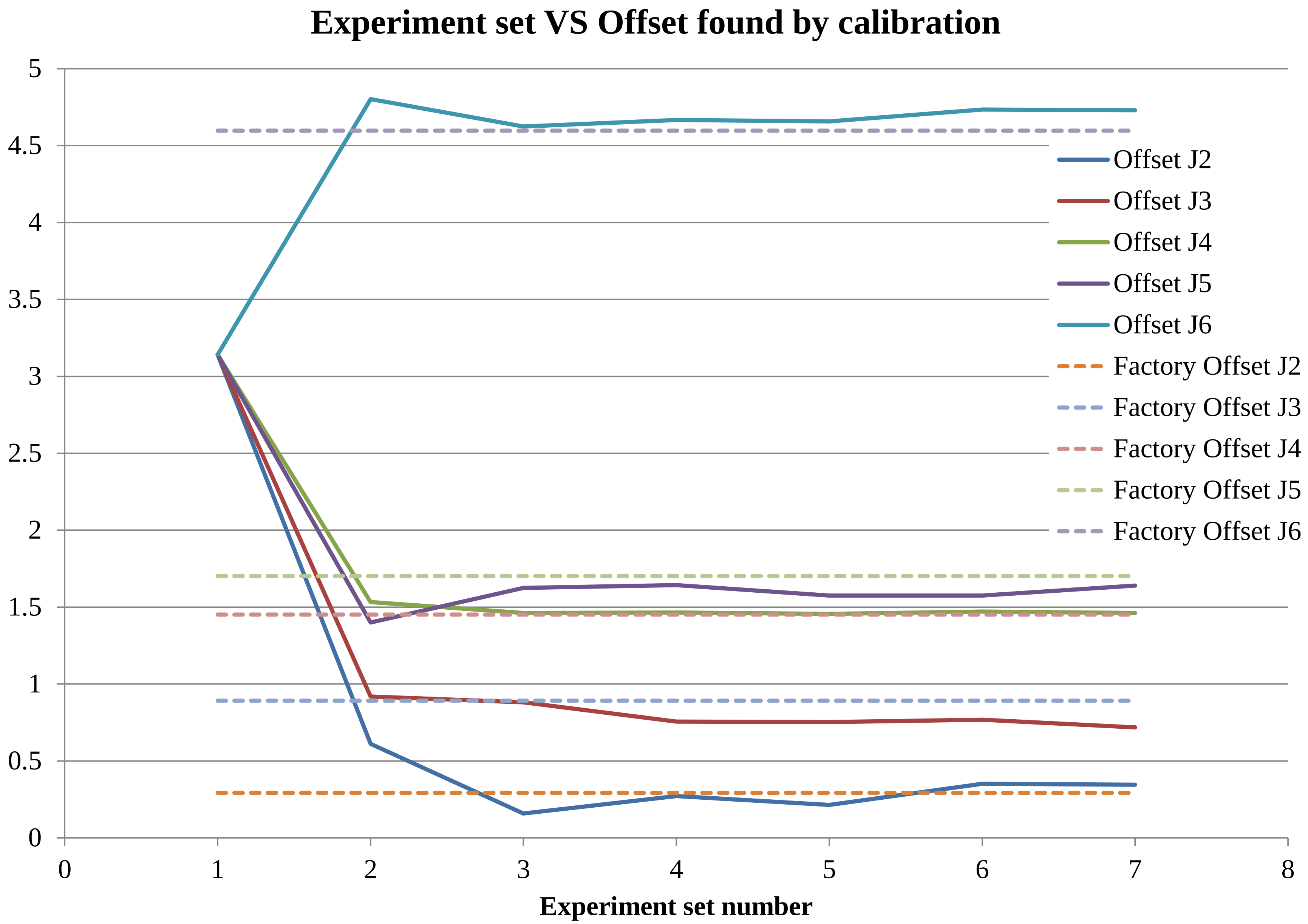


Figure 7.2: Single PSD calibration results

Note that the accuracy of the robot localization is limited to the control and feedback errors as well as robot accuracy itself. Additionally the sensitivity of variation of joint angle plays an important role on the accuracy and efficiency of the solution. The sensitivity also relies on robot configuration and PSD device placement as discussed in our previous papers. Moreover, the reliability and accuracy in the solution highly depends on the tolerance radius set for the experiments. The tolerance radius is the maximus distance from the center of the sensor that we allowed in order to record the joint data and to go from one pose to another. Even though decreasing the tolerance will give us better solutions, it will also make the calibration process to be slower. For instance, the experiments performed and reported in this dissertation where done with a tolerance equal to 0.5μm, this make the system to take up to 20 minutes per experiment. Decreasing the tolerance even further was not possible, since the battery died before one pose could be recorded. Therefore, decreasing the tolerance is not a reasonable option. Moreover, because the standard deviation between experiments with the same conditions was so large, we had to make experiments 3 times and use the average values in order to improve the solution. This will make the calibration process to reach up to one hour to just complete one set of experiments.

From Table 7.9 we can clearly argue that even the errors where acceptable, this method is still relatively far from being a reliable calibration system since the encoder errors in percentage can reach up to 2.78% relative to the offset reference.

7.3 Proposed Dual PSD Calibration Results

The same methodology described in section 7.1, and similar to section 7.2 was used to test the proposed dual calibration system. Similarly, we changed the factory motor offset values to a nominal value π in all joints initially. Then we run the calibration process only once. Afterwards, the joint offsets found by the calibration system were taken and added to the original values of joint offset.

Table 7.10: Offset values found by the calibration system

Joint	Offset found (deg)	Offset motor side (rads)
δ_2	-1.328229	-2.805018
δ_3	-1.257307	-2.216359
δ_4	-1.836191	-1.602379
δ_5	-1.646926	-1.465959
δ_6	1.82496	1.59258

Then, using equation 7.1 we calculate the new values that must be replaced in the robot controller, Table 7.11 show the results using the proposed robot calibration for the first time.

Table 7.11: New motor side offset values

Joint	Initial motor side offset	New motor side offset
δ_2	3.141592654	0.336574
δ_3	3.141592654	0.925233
δ_4	3.141592654	1.539214
δ_5	3.141592654	1.675634
δ_6	3.141592654	4.734171

The same process was repeated for experiments 2, 3, 4, 5, and 6 and the results are summarized in Table 7.12 Below:

Table 7.12: Experiment summary using the proposed dual PSD calibration

Joint	Initial	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6
δ_2	3.14159	0.336574	0.325773	0.278318	0.289165	0.303428	0.29738
δ_3	3.14159	0.925233	0.887535	0.89633	0.887619	0.895425	0.88697
δ_4	3.14159	1.539214	1.461959	1.456377	1.44742	1.461683	1.45323
δ_5	3.14159	1.675634	1.700322	1.672243	1.683818	1.683176	1.74855
δ_6	3.14159	4.734171	4.623761	4.620023	4.601065	4.606904	4.6029

7.3.1 Remarks & Discussions

The experiments performed using the proposed dual calibration system essentially verified the feasibility of the system and the ability to identify the joint offsets clearly much better than using our previous calibration system. Beyond experiment sets 3, 4, 5, and 6, we notice again that the calibration results start to oscillate on 0.0 deg offsets. Nevertheless, is at this point where we can analyze how reliable this calibration system is by calculating the percentage of errors as shown in Table 7.13.

Table 7.13: Error percentage using the proposed dual PSD calibration

Joint	Encoder Errors	Joint Errors	Encoder Errors %	Joint Errors %
δ_2	0.01092	0.005171	0.173796496	0.001436335
δ_3	0.005366	0.003044	0.085406713	0.000845611
δ_4	0.010383	0.011898	0.165255146	0.003305103
δ_5	0.030277	0.034015	0.481873205	0.009448494
δ_6	0.003535	0.004051	0.056259005	0.00112518

Errors were compute based on the maximum value of deviation from the factory motor offset values. Therefore, the reference is the factory motor offset value for each joint.

The plot comparing the newly found offsets with the factory offsets is shown in Figure 7.3.

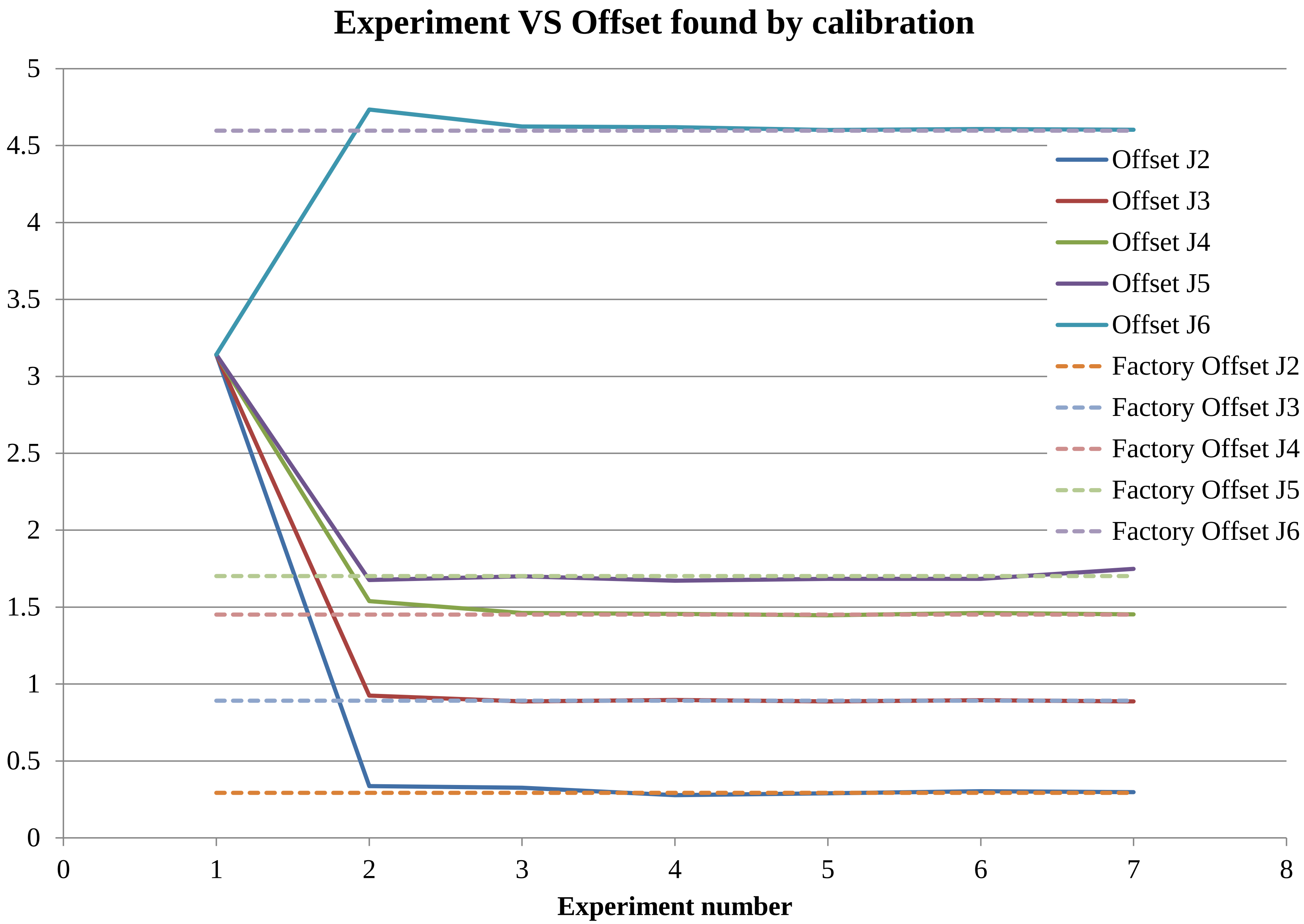


Figure 7.3: Proposed dual PSD calibration results

Note that again, the accuracy of the robot localization is limited to the control and feedback errors as well as robot accuracy itself. However the reliability and accuracy in the solution will no longer depends on the tolerance radius set for the experiments. Again, the tolerance radius is the maximum distance from the center of the sensor that we allowed in order to record the joint data and to go from one pose to another. In the case of the single PSD method, in order to compute the intersection of two lines by recording two poses intersecting the center of the sensor, we have to allowed certain tolerance as explained in section 7.2.1. In the case of the proposed dual PSD calibration system, the reflection made by the sensor itself will guarantee a perfect intersection between those lines and hence, improving the overall solution. Moreover, because the standard deviation between experiments with the same conditions was so small, we only perform the experiments once. This will essentially ensure that the speed of the calibration system will not be compromised. Furthermore, the values for the offset starts to oscillate sooner, so by performing the experiment once, we could achieve a similar level of accuracy in the solution than using the previous method 3 times.

From Table 7.13 we can clearly argue that the errors where acceptable, and with a better reliability compared to the single PSD calibration system, since the encoder errors in percentage can only reach up to 0.48% relative to the offset reference and can be as small as 0.05%. Furthermore, the experiments were made only once compared to the 3 sets and averages needed for the previous calibration system.

Chapter 8

Conclusions & Remaining Investigations

8.1 Summary of Contributions

In the modern era, the complexity of industrialization has played an important role in developing a strong economy, typically related to technological innovation in manufacturing. Manufacturing, in general, involves the development of large-scale productions utilizing industrial robots to create assembly lines. Generally, industrial robots reach high repeatability levels, and, for repetitive applications, they are able to perform such tasks successfully. Repeatability demonstrates the quality of modern robots and their precise positioning capabilities. However, it is also well-known in the robot industry that industrial robots have high repeatability but low accuracy. Nevertheless, the recent demand for high accuracy applications such as welding tasks, micro assembly operations, and surgery have increased the importance of and interests in robot calibration among researchers over the last few decades. Although there have been significant improvements in terms of accuracy of the newly designed industrial robot models, for such high accuracy applications the accuracy of the robot alone is not enough. While there are several sources of inaccuracies, the main source of inaccuracy lays in kinematic model parameter errors. According to researchers, around 90% of the inaccuracy in robot positioning is due to errors on assumed initial joint values of the robot. Without an appropriate robot calibration, any robotic system will experience accuracy degradation over time. Because of this, robot calibration has been used to improve position and orientation accuracy of industrial robots by identifying inaccuracies in the kinematic model parameters in order to create a more accurate model that better fits the real robot.

This dissertation proposed a new calibration system promising to be a fast, cost-effective, and reliable calibration solution. The proposed approach introduces a completely new technique for industrial robot calibration. The new calibration approach is not only able to compute the joint offset parameters of the robot, but is also capable of simultaneously calibrating the robot's workpiece relationship. It was also designed to be faster, simpler and cheaper than any other state of the art calibration systems. Throughout this dissertation our newly developed calibration device, the principle of our calibration system was presented and discussed. Also to take the system to the next level, and create an automated calibration process, we created a control system capable of guiding the robot's tool center point to a multi-position alignment. Simulations of the proposed controller were performed and successfully able to prove stability of the controlled system. The experimental results achieved in Chapter 5 essentially verified the feasibility of the proposed control system providing proof that the automation of the entire calibration system was possible. Simulations of the calibration algorithms in Chapter 6 demonstrated that the computational aspect of the system is robust and provides virtually perfect solutions to the system. Finally, experimental calibration results presented in Chapter 7 clearly demonstrated the advantages as well as the feasibility of the overall calibration system including device hardware, software and calibration algorithms.

8.2 Remaining Investigations

Thus far, our current investigations demonstrate that the overall calibration system is feasible, and that the system has some clear advantages over other calibration systems including our previous calibration system. However, this dissertation does not claim this system is suitable for the market at the moment, nor is it a final product, but rather emphasizes the big potential of this prototype, notwithstanding its limitations.

Therefore, based on this dissertation results, we propose the following remaining investigations:

1. Improve the implementation of the different control modules and stages required by the virtual linear constraints.

2. Expand the calibration algorithm to include robot workpiece frame calibration for experimental results.

3. Implement the calibration algorithms into the GUI so that once the process of reaching all poses required is completed, the results will be displayed automatically.

4. Perform more experiments to test the performance and reliability on the solution when using different PSCD configurations.

5. Apply a statistical method to analyze and verify deviations in the solution when running multiple experiments.

6. Compare experimental results using the proposed system with the current state of the art systems in terms of speed, reliability and accuracy on the solution.

REFERENCES

REFERENCES

[1] B. Mooring, Z. Roth, and M. Driels, *Fundamentals of manipulator calibration.* Wiley & Sons, Incorporated, John, 1991.

[2] P. Shiakolas and K. Conrad, "On the accuracy, repeatability, and degree of influence of kinematics parameters for industrial robots," *International journal of modelling and simulation*, vol. 22, no. 3, pp. 245–254, 2002.

[3] X. Zhong and J. Lewis, "A new method for autonomous robot calibration," *Robotics and Automation, 1995.*, no. Figure 2, pp. 1790–1795, 1995.

[4] J.-H. Borm and C.-H. Meng, "Determination of Optimal Measurement Configurations for Robot Calibration Based on Observability Measure," *The International Journal of Robotics Research*, vol. 10, pp. 51–63, Feb. 1991.

[5] A. Rauf, A. Pervez, and J. Ryu, "Experimental results on kinematic calibration of parallel manipulators using a partial pose measurement device," *Robotics, IEEE Transactions on*, vol. 22, no. 2, pp. 379–384, 2006.

[6] M. R. Driels and U. S. Pathre, "Robot calibration using an automatic theodolite," *The International Journal of Advanced Manufacturing Technology*, vol. 9, pp. 114–125, Mar. 1994.

[7] M. R. Driels, W. Swayze, and S. Potter, "Full-pose calibration of a robot manipulator using a coordinate-measuring machine," *The International Journal of Advanced Manufacturing Technology*, vol. 8, pp. 34–41, Jan. 1993.

[8] K. Lau, R. Hocken, and W. Haight, "Automatic laser tracking interferometer system for robot metrology," *Precision Engineering*, vol. 8, pp. 3–8, Jan. 1986.

[9] A. Nubiola and I. a. Bonev, "Absolute calibration of an ABB IRB 1600 robot using a laser tracker," *Robotics and Computer-Integrated Manufacturing*, vol. 29, pp. 236–245, Feb. 2013.

[10] M. Ikits and J. Hollerbach, "Kinematic calibration using a plane constraint," in *Proceedings of International Conference on Robotics and Automation*, vol. 4, pp. 3191–3196, IEEE, 1997.

[11] S. Motaghedi and Z. Roth, “Robot calibration with planar constraints,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, pp. 805–810, IEEE, 1999.

[12] A. Omodei, G. Legnani, and R. Adamini, “Calibration of a measuring robot: Experimental results on a 5 DOF structure,” *Journal of Robotic Systems*, vol. 18, pp. 237–250, May 2001.

[13] M. Meggiolaro, “Manipulator calibration using a single endpoint contact constraint,” in *2000 ASME Design Engineering Technical Conference*, pp. 1–9, 2000.

[14] S. Hutchinson, G. Hager, and P. Corke, “A tutorial on visual servo control,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[15] P. Corke, “Visual control of robot manipulators–a review,” *Visual servoing*, pp. 1–31, 1994.

[16] Y. Liu, Y. Shen, N. Xi, R. Yang, X. Li, and G. Zhang, “Rapid robot/workcell calibration using line-based approach,” *2008 IEEE International Conference on Automation Science and Engineering*, pp. 510–515, Aug. 2008.

[17] Y. Liu, N. Xi, G. Zhang, X. Li, H. Chen, C. Zhang, M. J. Jeffery, and T. a. Fuhlbrigge, “An automated method to calibrate industrial robot joint offset using virtual line-based single-point constraint approach,” *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 715–720, Oct. 2009.

[18] S. Blank, Y. Shen, N. Xi, C. Zhang, and U. C. Wejinya, “High precision PSD guided robot localization: Design, mapping, and position control,” *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 52–57, Oct. 2007.

[19] W. Newman, C. Birkhimer, R. Horning, and A. Wilkey, “Calibration of a Motoman P8 robot based on laser tracking,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 4, pp. 3597–3602, IEEE, 2000.

[20] T. Fuhlbrigge, “Practical industrial robot zero offset calibration,” *2008 IEEE International Conference on Automation Science and Engineering*, pp. 516–521, Aug. 2008.

[21] C. S. Gatla, R. Lumia, J. Wood, and G. Starr, “An Automated Method to Calibrate Industrial Robots Using a Virtual Closed Kinematic Chain,” *IEEE Transactions on Robotics*, vol. 23, pp. 1105–1116, Dec. 2007.

[22] R. P. Paul, "Robot Manipulators: Mathematics, Programming, and Control," in *The MIT Press*, Cambridge, 1981.

[23] S. A. H. S. A. Hayati, "Robot arm geometric link parameter estimation," *The 22nd IEEE Conference on Decision and Control*, vol. 22, pp. 1477–1483, 1983.

[24] C.-h. Wu, "A Kinematic CAD Tool for the Design and Control of a Robot Manipulator," *The International Journal of Robotics Research*, vol. 3, pp. 58–67, Mar. 1984.

[25] W. Veitschegger and C.-H. Wu, "Robot calibration and compensation," *IEEE Journal on Robotics and Automation*, vol. 4, no. 6, pp. 643–656, 1988.

[26] Z. Roth, B. Mooring, and B. Ravani, "An overview of robot calibration," *IEEE Journal on Robotics and Automation*, vol. 3, pp. 377–385, Oct. 1987.

[27] R. Z. S. Zhuang Hangi, *Camera-Aided Robot Calibration.* CRC Press, 1 ed., 2006.

[28] B. C. Jiang, J. Black, and R. Duraisamy, "A review of recent developments in robot metrology," *Journal of Manufacturing Systems*, vol. 7, pp. 339–357, Jan. 1988.

[29] D. E. Whitney, C. A. Lozinski, and J. M. Rourke, "Industrial Robot Forward Calibration Method and Results," *Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 1, p. 1, 1986.

[30] G. Puskorius and L. Feldkamp, "Global calibration of a robot/vision system," in *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, vol. 4, pp. 190–195, Institute of Electrical and Electronics Engineers, 1987.

[31] Y. Liu, N. Xi, J. Zhao, E. Nieves-Rivera, Y. Jia, B. Gao, and J. Lu, "Development and sensitivity analysis of a portable calibration system for joint offset of industrial robot," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3838–3843, IEEE, Oct. 2009.

[32] Hamamatsu Photonics, "PSD (Position Sensitive Detectors)," 2003.

[33] Y. Shen, Y. Liu, and K. Li, "Adaptive visual feedback control of manipulators in uncalibrated environment," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 1, pp. 7–12, IEEE, 2001.

[34] Y. Liu, N. Xi, Y. Shen, G. Zhang, and T. a. Fuhlbrigge, "High-accuracy visual/PSD hybrid servoing of robotic manipulator," *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 217–222, July 2008.

[35] F. Lewis, *Control of robot manipulators.* New York: Macmillan Publishing Company, 1993.

[36] J. D. R.S. Hartenberg R.S., "A kinematic Notation for Lower Pair Mechanisms Based on Matrices," *J. Appl. Mech. ASME*, pp. 215–221, 1955.