

Robot Control System for Multi-position Alignment Used to Automate an Industrial Robot Calibration Approach

Erick Nieves¹ and Ning Xi²

Abstract—Robot calibration is widely used in the manufacture industry to enhance and achieve a higher level of accuracy on industrial robot manipulators. Currently, there are many reliable calibration systems able to perform the calibration task. Those systems, however, are far from being a user friendly calibration tool; they are time consuming, very expensive, and usually requires a lot of human interaction. Therefore, we proposed a new calibration system able to overcome those problems with promising results. However, to take the system to the next level, and create an automated calibration process, we need to create a control system capable of guiding the robot's tool center point to a multi-position alignment. Throughout this paper the control approach needed to achieve automation of the entire system is presented and discussed. Simulations and experimental results demonstrated the feasibility of the overall calibration system including hardware, software and control algorithms.

I. INTRODUCTION

In our modern era, the complexity of industrialization had 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. Certainly repeatability demonstrates the quality of a modern robot and their precise positioning capabilities. However, it is also well-known that industrial robots possess high repeatability but low accuracy [1]. Nevertheless, the recent demand of high accuracy applications such as welding tasks, micro assembly operations, surgery, etc. have increased the importance and interests of robot calibration among researchers over the last decades. Although there have been significant improvements in terms of accuracy on 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, etc.), the main source of errors lies in the kinematic parameters defined within the robot's controller. According to [2] and [3], around 90% of the inaccuracy in robot positioning is mostly due to errors at the initial joint values of the robot. Without an appropriate robot calibration,

any robotic system will experience accuracy degradation over time. Due to this fact, robot calibration has been used to improve the position and orientation accuracy of industrial robots by identifying inaccuracies in the kinematic parameters in order to create a more accurate model that better fits the real robot.

There are many calibration approaches that have been designed by researchers with promising methodologies to calibrate 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], Inclometers [5], Theodolites [6], Coordinate Measurement Machines (CMMs) [7] and laser tracking systems [8]. Other methods impose some 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 to generate an equation system capable of determining a set of parameters, also known as self-calibration system. Due to this particular advantage, self-calibration systems are widely investigated and analyzed more than any other methods. Furthermore, additional expensive measuring devices are not required. In [9] the authors measured the position and poses of a robot by matching the pin of the TCP to an aperture on a dime. Additionally, vision-based systems have been developed to perform the calibration task. In those systems however, the lack of resolution under wide fields of view is a major problem, as well as the low frame rate cameras possess [10]. Because those devices are so expensive, or their procedures are time consuming, they are difficult to be used extensively in the manufacturing plants. For instance, a Laser Tracker System can cost up to \$100,000 US dollars. Therefore, it is particularly important to design a system which could be both cost-effective and easy to implement, while still be able to achieve a high level of accuracy.

This paper presents a robot control system for our optical approach based on 2 PSDs (i.e. Position Sensitive Detectors) to improve the calibration process in respect to time, reliability, and cost aspects that other methods still lack. Our focus will be concentrated in the control methodology that offers the solution to reduce or even eliminate the need for human interaction in the process. During the calibration process, the procedure to aim a laser beam at the center of each sensor will only repeat twice, so the approach is faster and simpler than our previous methods [11], [12]. Experiments and simulations were conducted on an ABB industrial robot (IRB120) to verify the feasibility of the proposed control method as well as the newly developed calibration system.

*Research partially supported by ABB University Research Program.

¹Erick Nieves is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA (e-mail:nieveser@msu.edu).

²Ning Xi is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA (e-mail: xin@egr.msu.edu).

II. AUTOMATED CALIBRATION SYSTEM OVERVIEW

A. Data Acquisition System

Our proposed calibration system approach [13] requires a laser beam to shoot into one PSD in a way that the reflection hits 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 find such constrained line. A picture of such device is shown in Figure 3 to the right.

Internally our device must be able to carry the signal information from the PSDs to the computer in order to control the robot to the desired state. Figure 1 highlights the basic internal components of the device as well as the interaction between the device and the robot controller.

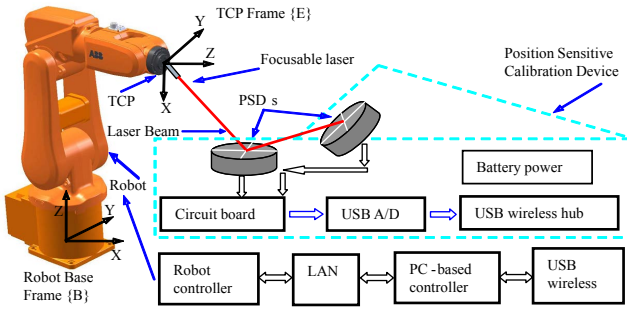


Fig. 1. Feedback data acquisition system.

After the processing circuit board gets the raw data from the two sensors, the signals are taken by a wireless USB hub through a data acquisition card. Detailed information about the data acquisition system is explained more extensively in [14]. Once the data reaches the computer we implement our PC-based controller so that the robot TCP move to the desired position relative to the sensors.

B. Calibration system setup

Figure 2 shows the schematic model of the calibration system, implemented and verified by an ABB robot under lab testing. This robotic system comprises of an ABB robot controller (IRC5 Compact) and a six degree of freedom (6-DOF) robot manipulator (IRB120).

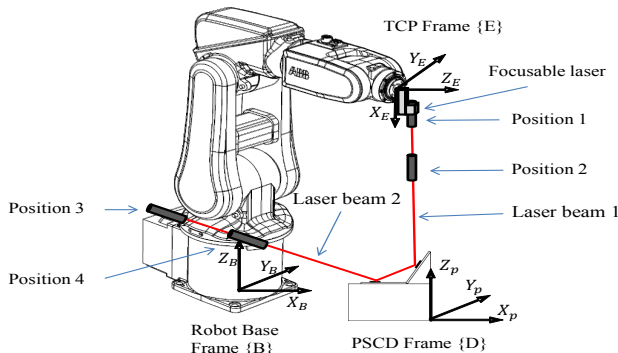


Fig. 2. Calibration system schematic.

In Figure 2, the robot calibration system mainly consists of a robot, a laser and camera fixture; and our position sensitive calibration device (PSCD). A laser pointer is mounted on its fixture and attached to the robot TCP; this is shown in details in Figure 3 to the left. The laser beam is tuned to align its orientation toward the X-axis of the TCP frame. Two segmented PSDs are mounted on a portable custom-built, high-precision fixture. In theory segmented PSDs have a higher resolution than $0.1\mu\text{m}$ and even under experimental conditions, its resolution may reach approximately $2\mu\text{m}$. 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.

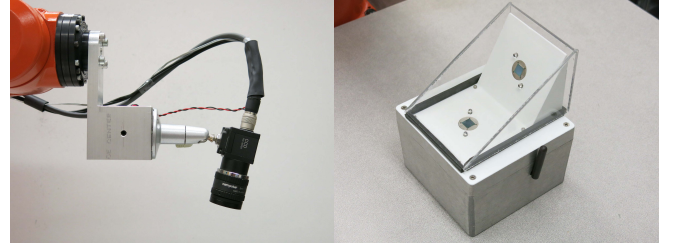


Fig. 3. Laser and camera fixture along with our newly developed position sensitive calibration device.

The calibration process, in Figure 2, is completed by locating the TCP 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. Details on how to control the robot to achieve such requirements are explained in Section III. Therefore, four sets of robot joint angles can be recorded. Based on the recorded joint angles and robot forward kinematics, a calibration algorithm is developed.

III. ROBOT CONTROL METHOD

The robot control system is among the most crucial and important elements of the proposed calibration system. 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. Afterwards, the reflection from such PSD sensor must also be controlled to reach a centered position of the other sensor. Indeed, it is a challenging problem that must be overcome in order to make the entire calibration system an automated and faster procedure.

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 introduction to this topic can be found in [10]. 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 [12], as shown below in Figure 4. The camera is mounted tilted towards the

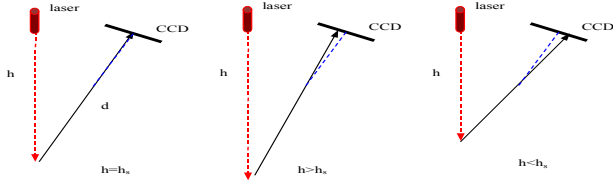


Fig. 4. Schematic of laser line length control.

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 hits the center of the first PSD, i.e. PSD1, then we proceed to the third and final stage. At this point, we should have the laser line length information from the first stage. Therefore, we use it to modify the kinematic model of the robot to be able to fix the point found in the second stage. Once this step is done, we can control only the orientation of that fixed point such that the reflection hit the center of the second PSD (PSD2) without changing its position in PSD1. At this stage, we used a similar approach to move it to the center of PSD2 but this time controlling only orientation. A block diagram of the entire process is shown in Figure 5.

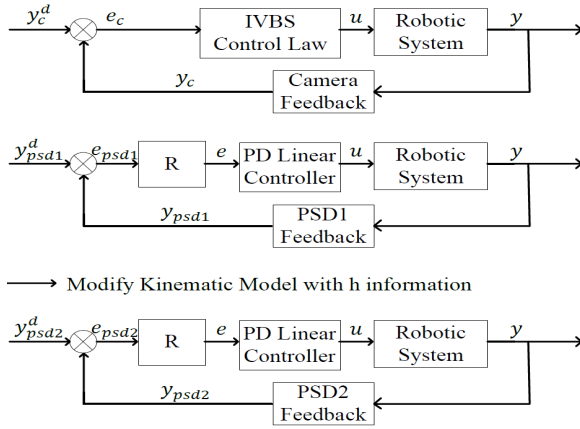


Fig. 5. Robot control system block diagram.

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

A. 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 [12], i.e.

$$\dot{f} = J(u, v, z) \dot{r} \quad (1)$$

where;

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

$f = (u, v)^T$ are the current image coordinate features, $\dot{f} = \dot{f}^d - \dot{f}$ 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;

$$\dot{U} = \dot{r} = \Gamma J^{-1}(u, v, z) \dot{f} \quad (3)$$

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

B. Laser Line Length control

To avoid the well-known problem of singularity in the image Jacobian, Equation (1) can be decomposed into its rotational and translational components, i.e.

$$\dot{f} = J(u, v)w + J(u, v, z)v \quad (4)$$

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;

$$w = \Gamma J^{-1}(u, v) \dot{f} \quad (5)$$

where $\dot{f} = \dot{f}^d - \dot{f}$ represents 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.

C. 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 PSD. For instance, let us define the homogeneous transformation matrix of the base frame $\{B\}$ as ${}^P T_B$, which can be written as;

$${}^P T_B = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \quad (6)$$

where $R \in \mathfrak{R}$ denotes the rotation matrix of the base frame $\{B\}$ relative to the PSD 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 PSD 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 ;

$$\xi = \begin{bmatrix} v & w \end{bmatrix}^T \quad (7)$$

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

$$\xi_P = \begin{bmatrix} R_1 & 0_{3 \times 3} \\ 0_{3 \times 3} & R_2 \end{bmatrix} \xi_B \quad (8)$$

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 (8) for ξ_B we have;

$$\xi_B = \begin{bmatrix} R_1^T & 0_{3 \times 3} \\ 0_{3 \times 3} & R_2^T \end{bmatrix} \xi_P \quad (9)$$

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

$$\dot{s}(t) = \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} \quad (10)$$

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 (9) and (10) we have the following;

$$\dot{s}(t) = \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = R_1 v_B \quad (11)$$

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;

$$e(t) = s(t) - s^d \quad (12)$$

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 (31) by v_B ;

$$v_B = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = R_1^{-1} \dot{s}(t) \quad (13)$$

Therefore, a proportional controller is designed such that;

$$\dot{e} = -K_p e \quad (14)$$

Finally, substituting Equation (14) into (13) we have;

$$v_B = -K_p R_1^{-1} e \quad (15)$$

D. 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;

$${}^P T_B = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \quad (16)$$

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

$$\xi = \begin{bmatrix} v & w \end{bmatrix}^T \quad (17)$$

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;

$$\xi_P = \begin{bmatrix} R_1 & 0_{3 \times 3} \\ 0_{3 \times 3} & R_2 \end{bmatrix} \xi_B \quad (18)$$

Here ξ_P represents 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 (18) for ξ_B we have;

$$\xi_B = \begin{bmatrix} R_1^T & 0_{3 \times 3} \\ 0_{3 \times 3} & R_2^T \end{bmatrix} \xi_P \quad (19)$$

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

$$\dot{s}(t) = \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} \quad (20)$$

Because of the position remaining constant, we can state that $v = 0$ and also $w_z = 0$: Therefore, if we combine this with Equations (19) and (20) we have the following;

$$\dot{s}(t) = \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = R_2 w_B \quad (21)$$

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;

$$e(t) = s(t) - s^d \quad (22)$$

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 (21) by v_B ;

$$w_B = \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = R_2^{-1} \dot{s}(t) \quad (23)$$

Therefore, a proportional controller is designed such that;

$$\dot{e} = -K_p e \quad (24)$$

Finally, substituting Equation (14) into (13) we have;

$$w_B = -K_p R_2^{-1} e \quad (25)$$

IV. CALIBRATION SYSTEM APPROACH

The calibration system approach is one of the most crucial components to achieve an accurate robot calibration. At this stage, we are looking to use the robot's encoder information after the controller stage is successfully done, to be able to accurately determine the real TCP position. Nieves et al. [13] introduces the algorithm derivations in more detail. There are two main and separate ways to calibrate an industrial robot, both equally useful in the calibration task, i.e. joint offset calibration, and robot workpiece frame calibration. Yet another advantage of our system is the ability to perform both at the same time.

A. Analysis of the Kinematics Error Model

The Denavit-Hartenberg [15] is a commonly used convention to represent frame references in the forward kinematic model of a robot manipulator as follows,

$${}^B T_E = \prod_{i=1}^n A_i \quad (26)$$

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,

$$\tilde{A}_i = \begin{bmatrix} c\tilde{\theta}_i & -s\tilde{\theta}_i c\alpha_i & s\tilde{\theta}_i s\alpha_i & a_i c\tilde{\theta}_i \\ s\tilde{\theta}_i & c\tilde{\theta}_i c\alpha_i & -c\tilde{\theta}_i s\alpha_i & a_i s\tilde{\theta}_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

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 (27) into (26), forward kinematics with the offset is written as,

$${}^B T_E = \prod_{i=1}^6 \tilde{A}_i = \tilde{A}_1 \tilde{A}_2 \tilde{A}_3 \tilde{A}_4 \tilde{A}_5 \tilde{A}_6 \quad (28)$$

Note that joint 1 depends on the robot base frame. So in (28) there are five unknown parameters, which are the last five offsets $\tilde{\delta}_i (i = 2, 3, 4, 5, 6)$.

B. Joint Offset Calibration

Joint Offset Calibration is the process of calculating the individual error contribution of robot joints, so that they can be compensated later in the kinematic model.

In our proposed system, this process is accomplished by locating the TCP and laser pointer several times at four different locations. Therefore, four sets of joint angles can be recorded by the robot controller. The main idea is to find a squared error for each two laser lines. Therefore, the unknown parameters, i.e. the last offsets $\tilde{\delta}_i (i = 2, 3, 4, 5, 6)$, are found by minimizing the total sum of squared errors, i.e.

$$\Psi = \operatorname{argmin} \sum_{k=1}^2 \psi_k \quad (29)$$

C. 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, and it is usually in the form of a transformation matrix so that the entire kinematic model can be compensated later on.

Let R and t be the rotation matrix and the translation vector of ${}^B T_D$, respectively. Then following the detailed derivations found in our previous paper [13], we have;

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad (30)$$

where,

$$\begin{aligned} R_{11} &= k_x^2(1 - \cos(\theta)) + \cos(\theta) \\ R_{12} &= k_x k_y(1 - \cos(\theta)) - k_z \sin(\theta) \\ R_{13} &= k_z k_x(1 - \cos(\theta)) + k_y \sin(\theta) \\ &\vdots \end{aligned} \quad (31)$$

In the relationships defined in our previous paper, the values of $[t_x \ t_y \ t_z]^T$ are unknown and can be computed. Therefore the calibration matrix will be given by;

$${}^B T_D = \begin{bmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (32)$$

V. SIMULATION AND EXPERIMENTAL RESULTS

A. Robot Control 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 [12]. 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 PSD. 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 6 to the left. Similarly, in Figure 6 to the right, the simulations show stability as well as for the third stage of the controller using orientation control over PSD2 while keeping position errors over PSD1 equal to zero.

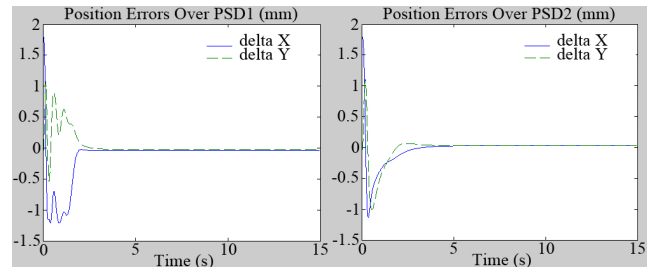


Fig. 6. Robot control simulation results for PSD1 and PSD2.

B. Robot control experimental results using IRB120 Robot

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 7 shows the image features of the two LEDs along with the features of the laser spot. Figure 7 demonstrates the convergence between the center of the two LEDs and the laser spot feature. 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.

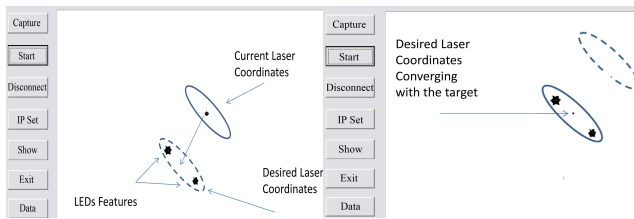


Fig. 7. Image features before and after IBVS control.

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 8 shows how PSD-based servo control was able to reach the center of the PSD1, and 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 and the eventual automation of the entire calibration system eliminating the need of human interaction.

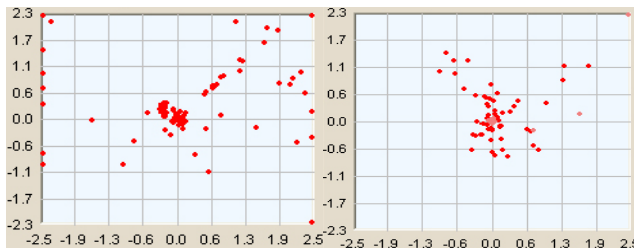


Fig. 8. Robot controller results after all stages were completed.

VI. CONCLUSIONS

Robot calibration is widely used in the manufacture industry to enhance and achieve a higher level of accuracy

on industrial robots. We have proposed in the past a new calibration system promising to be a fast, cost-effective, and reliable calibration solution. However, 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 by this paper essentially verified the feasibility of the proposed control system providing proof that the eventual automation of the entire calibration system is possible. Further research on different types of controllers is ongoing as well as experiments on the whole calibration system performance.

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. Pervaz, 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] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] E. Nieves, N. Xi, B. Du, and Y. Jia, "A reflected laser line approach for industrial robot calibration," in *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, (Kaohsiung, Taiwan), pp. 610–615, IEEE, July 2012.
- [14] E. Nieves, N. Xi, Y. Jia, C. Martinez, and G. Zhang, "Development of a Position Sensitive Device and Control Method for Automated Robot Calibration," in *IEEE International Conference on Automation Science and Engineering (IEEE CASE 2013)*, (Wisconsin, USA), p. 1133–1138, IEEE, 2013.
- [15] 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.