# A COMPUTER SYSTEM TO CALIBRATE INDUSTRIAL ROBOTS USING A MEASUREMENT ARM

**Luciano Selva Ginani, lucianoginani@unb.br**
**José Maurício S. T. Motta, jmmotta@unb.br**
University of Brasilia – Department of Mechanical Engineering, Faculty of Technology, Brasília-DF, Brazil, 70910-900

***Abstract.*** *One of the greatest challenges in today's industrial robotics is the development of off-line programming systems that allow drastic reduction in robots' reprogramming time, improving productivity. Among the problems that hold back the development of robots' off-line programming is the lack of accuracy in static and dynamic positioning. The intention here is to improve the robot positioning accuracy so that off-line programming is viable. A computer system was built for developing and implementing a calibration system that involves the joint work of computer and measurement systems, including all processes of a robot static calibration, from kinematic modeling to model identification and evaluation. Each step of this system's development is presented together with its theoretical basis. With the development of a remote maneuvering system based on ABB S3 controller experimental tests have been carried out using an IRB2000 robot and a measurement arm (ITG ROMER) with a position measurement accuracy of 0.087mm. The robot model used by its controller was identified and the robot was calibrated and evaluated in different workspaces resulting in an average accuracy improvement from 1,5mm to 0,3mm.*

***Keywords****: Robot Calibration, Off-Line Programming, Parameter Identification, Robot Position Accuracy.*

## 1. INTRODUCTION

For decades robots have been used in manufacturing industries to replace men work in simple, repetitive and dangerous tasks. Investments on the robotic research and technological achievements in computer sciences and electronics has brought about to the arena new possibilities to make robots more accurate and precise, pushing the field of robotics towards an enormous amount of applications, from industry to service, entertainment to marketing robotics. However, one of the greatest challenges in today's industrial robotics is still the mismatch between control models and the physical robots, making the so desired robot off-line programming largely used in industry an achievement quite far away to be reached. That means, robots have a very good repeatability but still a poor accuracy.

In addition to improving robot accuracy through software (rather than by changing the mechanical structure or design of the robot), calibration techniques can also minimize the risk of having to change application programs due to slight changes or drifts (wearing of parts, dimension drifts or tolerances, and component replacement effects) in the robot system. This is mostly important in applications that may involve a large number of task points.

Robot calibration is an integrated process of modeling, measurement, numeric identification of actual physical characteristics of a robot, and implementation of a new model (Schröer, 1993, Motta, 2007).

The proposal of this article is to present a robot calibration system that has been developed aiming at improving robot position accuracy. Mathematical basics, experimental procedures and results are presented and discussed. The system was conceived to be used with an ABB IRB2000 robot model, however it can easily be adapted and used with any type of industrial robots.

## 2. THE ROBOT CALIBRATION SYSTEM

Robot calibration is the process of improving the robot accuracy by modifying its control software (Bernhardt and Albright, 1993). General calibration systems can be divided into two main groups: static and dynamic (Schröer, 1993). While static calibration systems focus on the correction of parameters such as joint/axis geometries that affects static positioning of the robot, the dynamic calibration focus on the determination of parameters that affects dynamic characteristics such as torque and speed (Bernhardt and Albright, 1993).

The calibration system described here involves the joint work of a measurement system, an off-line robot calibration model and the robot controller as shown in Fig. 1.

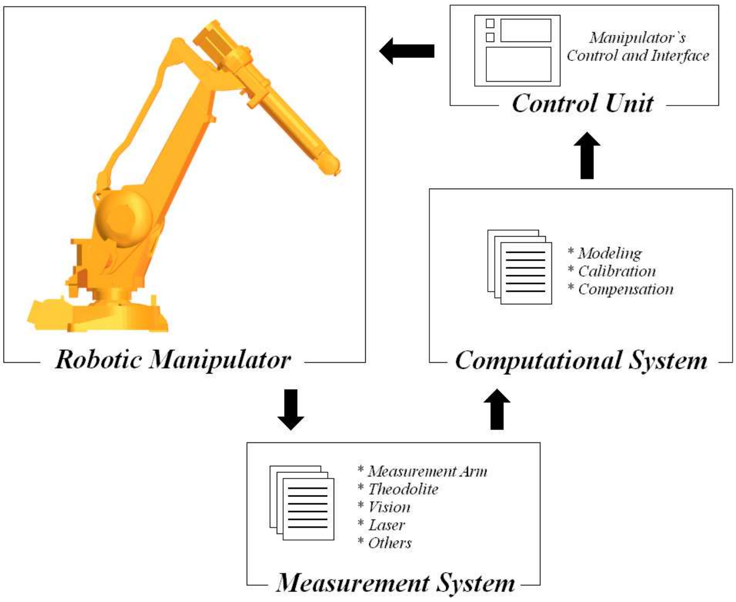


Figure 1. Block Diagram of the developed calibration system

In the next sections each part of the calibration system will be discussed such that the entire system can be fully understood. The robot calibration model outputs geometric parameters that describe the robot geometry (links lengths, joint offsets and axis misalignments). The robot calibration procedures can be divided into four main steps:

- Kinematic Modeling;
- Position Measurements;
- Parameter Identification;
- Position Compensation.

Kinematic modeling is a subject that has been widely studied for a long time, and together with dynamics it is the topic in robotics that has produced the largest number of publications up to date (Goldenberg and Emami, 1999). Kinematic modeling for robot calibration has to include an error model to fit the actual robot errors.

The measurement step is the most critical in the shop-floor since measurement data have to be many times more accurate than the robot accuracy expected after the calibration procedures. There is a wide range of measurement systems available with different levels of accuracy (Kyle, 1993), including contact and non-contact systems. The measurement system used here can only measure end-effector positions, since orientation measuring is not possible with the type of measuring device used. Only few measuring systems have this capacity and some of them are usually based on vision or optical devices. The price of the measuring system appears to be a very important issue for medium size or small companies.

Parameter identification is the step where data acquired with the measurement system are processed within a mathematical model specific for error searching, producing a corrected robot kinematic model. The errors calculated are used to fit the robot model to the experimental data.

Position compensation refers to using the robot geometrical errors obtained from the parameter identification step after the robot kinematic model is corrected to modify the robot’s control commands, compensating joint positions as needed to improve the robot position accuracy.

### 2.1. Measurement System

The robot calibration system constructed for this work gives support to different measurement systems, thanks to its modular construction. In this work, the measurement system used was a Measurement Arm ITG ROMER with an accuracy reported by the manufacturer of 0.087mm. The system can be seen in details in Fig. 2.


Figure 2. ITG ROMER measurement arm

The measurement arm was used to measure the end-effector positions of the ABB IRB2000 robot.

**2.2. Mathematical Basis**

Concerning mathematics, robot calibration is basically a problem of fitting a non-linear model to experimental data. The results are error parameters that are identified using a proper cost function.

A robot kinematic model can be seen as a function that relates kinematic model parameters and joint variables to coordinate positions of the robot end-effector. A kinematic model following the Denavit-Hartenberg convention (McKerrow, 1991) can be formulated as:

$$P = f(\theta,\alpha,d,l) = R_Z(\theta)T_Z(d)T_X(l)R_X(\alpha) \tag{1}$$

where $P$ represents position and orientation coordinates of the manipulator end-effector (Tool Center Position – TCP) and $\theta$, $\alpha$, $d$ and $l$ are the four parameters that define the transformation from the robot base frame to the TCP-frame.

The first derivative of Eq. (1) can be seen as the positioning and orientation error equation of the robot TCP coordinates (Hollerbach and Benett, 1988),

$$\Delta P = \partial P/\partial\theta\,\Delta\theta + \partial P/\partial\alpha\,\Delta\alpha + \partial P/\partial d\,\Delta d + \partial P/\partial l\,\Delta l \tag{2}$$

where $\Delta P$ is a measure of error and can be measured. Considering the full transformation $P$, from the robot's base frame to the TCP-frame, the measured robot position M with reference to the measurement system origin and the transformation $B$ that identifies the robot base frame in the measurement system, $\Delta P$ is the vector illustrated in Fig. 3.

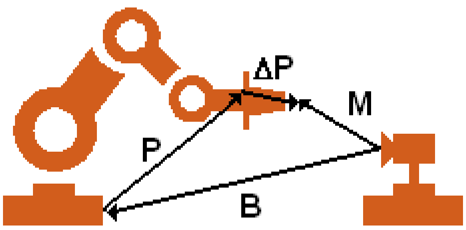


Figure 3. Calibration transformations

The transformation $B$ can still be treated as a link that makes part of the robot model such that Fig. 3 becomes Fig. 4 and the error value $\Delta P$ can be calculated using Eq. (3).

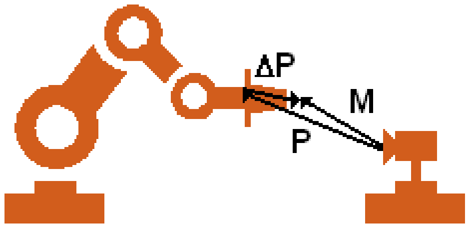


Figure 4. Simplified representation of the calibration transformations

$$\Delta P = M - P \tag{3}$$

The transformation $P$ is then iteratively modified as the error parameters of the robot model are upgraded, and by the end of the calibration process the transformation $P$ will represent the real robot and locate this robot within the measurement system coordinate frame.

Rewriting Eq. (2) in a matricial form for various measured positions and orientations of the robot end-effector, Eq. (4) is obtained, where $J$ is the Jacobian matrix containing the partial derivatives from Eq. (1) and $\Delta x$ is the modeling parameters error vector,

$$\begin{bmatrix} \Delta P_1 \\ \Delta P_2 \\ \vdots \\ \Delta P_n \end{bmatrix} = \begin{bmatrix} \partial P_1/\partial\theta & \partial P_1/\partial\alpha & \partial P_1/\partial d & \partial P_1/\partial l \\ \partial P_2/\partial\theta & \partial P_2/\partial\alpha & \partial P_2/\partial d & \partial P_2/\partial l \\ \vdots & \vdots & \vdots & \vdots \\ \partial P_n/\partial\theta & \partial P_n/\partial\alpha & \partial P_n/\partial d & \partial P_n/\partial l \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta\alpha \\ \Delta d \\ \Delta l \end{bmatrix} = \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_n \end{bmatrix} \Delta x \Rightarrow \mathbf{J}\Delta x = \mathbf{\Delta P} \tag{4}$$

Thus the calibration problem is reduced to the solution of a non-linear system of the type $\mathbf{A}x = \mathbf{b}$.

There are many different methods to solve this type of system and one of those that is widely used is the Squared Sum Minimization (SSM). Many authors (Jacoby et al, 1972 and Dennis et al, 1983) discuss extensively those methods and algorithms are easily found in the literature (Press et al, 1992).

One method to solve non-linear least-square problems proved to be very successful in practice and then recommended for general solutions is the algorithm proposed by Levenberg-Marquardt (LM algorithm) (Dennis & Schnabel, 1983). Several algorithms versions of the L.M. algorithm have been proved to be successful (globally convergent). It turns to be an iterative solution method by introducing few modifications in the Gauss-Newton method in order to overcome some divergence problems. (Jacoby et al, 1972)

Each algorithm iteration has three steps, where $x_k$ represents the parameter list of the mathematical model in the k$^{th}$ iteration and $\Delta x_k$ the alterations to be introduced in the model.

1. Calculation of the robot`s jacobian ( $\mathbf{J}(x_k)$ );
2. Calculation of the vector $\Delta x_k$ using the relation $\Delta x_k = -\left\{[\mathbf{J}(x_k)]^T \mathbf{J}(x_k) + \mu_k \mathbf{I}\right\}^{-1} [\mathbf{J}(x_k)]^T \mathbf{\Delta} P(x_k)$;
3. Upgrade $x_{k+1} = x_k + \Delta x_k$ and $k = k+1$.

, where $\mu_k$ is obtained from the formation law in Eq. (5).

$$\begin{cases} \mu_0 = 0.001 \\ \mu_{k+1} = \begin{cases} 0.001\lambda & se \left\|\Delta\hat{T}(x_{k+1})\right\| \geq \left\|\Delta\hat{T}(x_k)\right\| \\ 0.001/\lambda & se \left\|\Delta\hat{T}(x_{k+1})\right\| < \left\|\Delta\hat{T}(x_k)\right\| \end{cases} \\ 2.5 < \lambda < 10 \end{cases} \tag{5}$$

### 2.3. Computer System

In order to implement the calibration procedures in an user-friendly environment a computer program was developed. More than robot calibration results the implemented software also offers a series of resources from robot modeling to Jacobian analysis.

The software was developed in C/C++ with the help of the Borland C++ Builder compiler using the 3D library OPEN-GL for visualization. One important aspect considered was to make calibration procedures an easy and simple task that can be carried out in a fast and precise routine.

Each of the software features can be easily accessed with the help of its main menu, as can be seen in Fig. 5. The calibration screen can be observed in Fig. 6

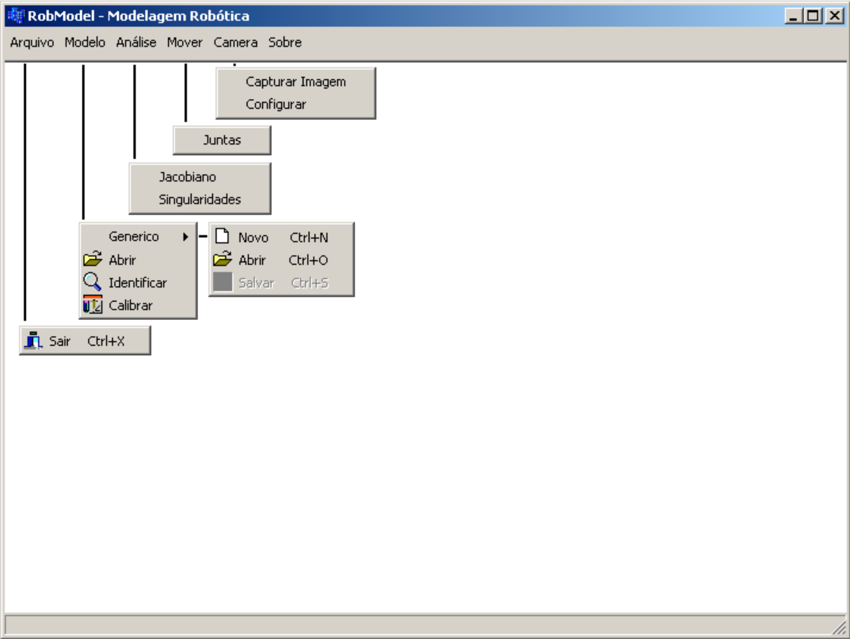


Figure 5. Calibration software main screen

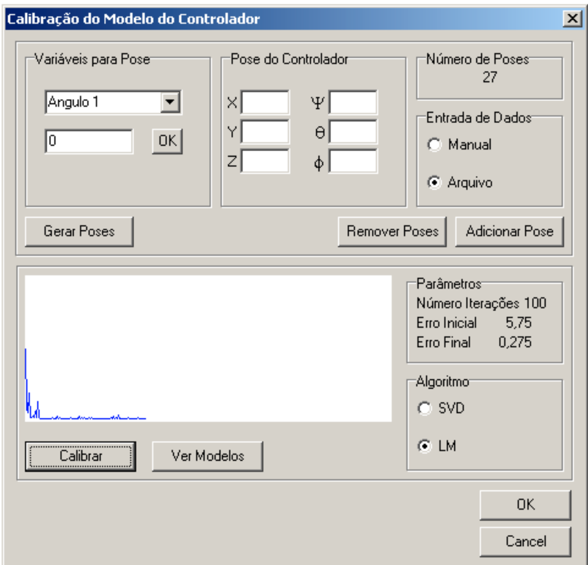


Figure 6. Calibration screen

## 3. EXPERIMENTAL PROCEDURES

The robot calibration system implemented was evaluated on an ABB IRB2000 robot using an ITG ROMER measurement arm. The robot was calibrated within different workspaces, and the accuracy improvement could be assessed in various robot configurations. The system was also used to validate the correct matching between the nominal robot kinematic model in the off-line calibration software and the nominal robot kinematic model in the control unit. The results and procedures are presented and discussed to show up the performance of the developed system and the robot accuracy improvements.

### 3.1. IRB2000 Modeling

The first step to calibrate a robotic manipulator is kinematic modeling. The IRB2000 robot is an industrial robot with six degrees of freedom used in a wide range of tasks that can be remotely controlled by using its RS232 interface and the communication protocols ADPL10 and ARAP. (ABB, 1993)

The IRB2000 robot (Fig. 7 and Fig. 8) has perpendicular and parallel axes. However, the well known Denavit-Hartemberg convention, shown in Eq. (6), largely used for kinematic modeling cannot be used in error parameter models when modeling parallel axes due to singularities that come about in the Jacobian matrix of Eq. (4). This issue is discussed in details in Motta (2005) and Schröer (1997). A possible convention for parallel axes is the Hayati-Mirmirani, that cannot be used in perpendicular axes for the same reason. The Hayati-Mirmirani is as four-parameter convention that describe the transformation between two parallel axes as shown in Eq. (7).

$$f(\theta,\alpha,d,l) = R_Z(\theta)T_Z(d)T_X(l)R_X(\alpha) \tag{6}$$

$$f(\theta,\alpha,\beta,l) = R_Z(\theta)T_X(l)R_X(\alpha)R_Y(\beta) \tag{7}$$

So, the kinematic model was constructed using the Denavit-Hartenberg convention for perpendicular axes and the Hayati-Mirmirani convention for parallel axes. The model parameters used are shown in Tab. 1, where $\delta$ are the error parameters between the nominal model and the actual robot model to be identified by the calibration system.

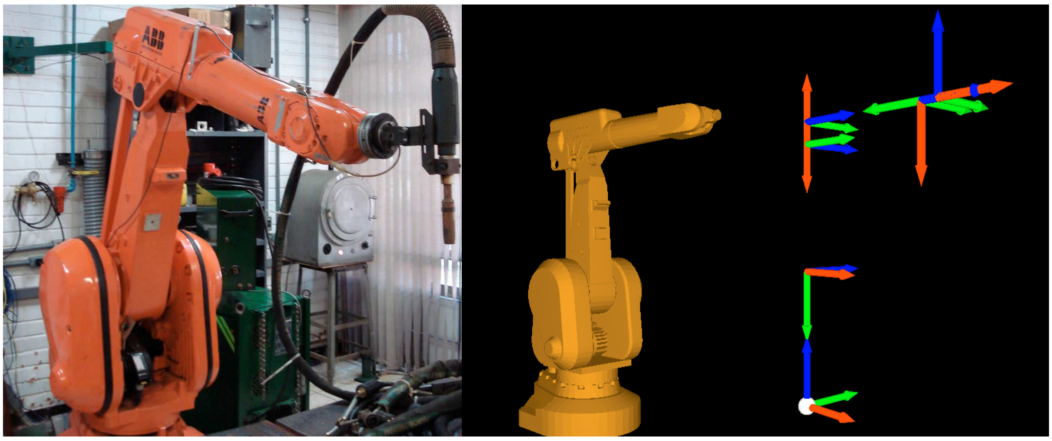

Figure 7. IRB2000 and its 3D representation

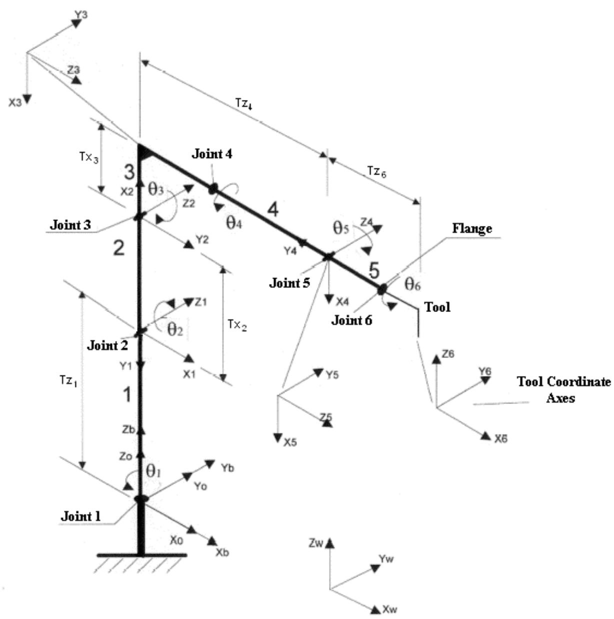


Figure 8. IRB2000 modeling

Table 1. Joints and links modeling for IRB2000 (units in mm and degrees)

| Variable | Model |
|---|---|
| **Link B** | |
| $T_{x_b} + \delta_{Tx_b}$ | 0.00 |
| $T_{y_b} + \delta_{Ty_b}$ | 0.00 |
| $T_{z_b} + \delta_{Tz_b}$ | 0.00 |
| $R_{x_b} + \delta_{Rx_b}$ | 0.00 |
| $R_{y_b} + \delta_{Ry_b}$ | 0.00 |
| $R_{z_b} + \delta_{Rz_b}$ | 0.00 |

| Variable | Model |
|---|---|
| **Link 0** | |
| $T_{x_0}$ | 0.00 |
| $R_{z_0}$ | 0.00 |
| $T_{z_0}$ | 0.00 |
| $R_{x_0}$ | 0.00 |

| Variable | Model |
|---|---|
| **Link 1** | |
| $T_{x_1} + \delta_{Tx_1}$ | 0.00 |
| $T_{z_1} + \delta_{Tz_1}$ | 750.00 |
| $R_{x_1} + \delta_{Rx_1}$ | -90.00 |
| $R_{z_1}$ | 0.00 |

| Variable | Model |
|---|---|
| **Link 2** | |
| $T_{x_2} + \delta_{Tx_2}$ | 710.00 |
| $R_{y_2} + \delta_{Ry_2}$ | 0.00 |
| $R_{x_2} + \delta_{Rx_2}$ | 0.00 |
| $R_{z_2} + \delta_{Rz_2}$ | -90.00 |

| Variable | Model |
|---|---|
| **Link 3** | |
| $T_{x_3} + \delta_{Tx_3}$ | -125.00 |
| $T_{z_3} + \delta_{Tz_3}$ | 0.00 |
| $R_{x_3} + \delta_{Rx_3}$ | 90.00 |
| $R_{z_3} + \delta_{Rz_3}$ | 180.00 |

| Variable | Model |
|---|---|
| **Link 4** | |
| $T_{x_4}$ | 0.00 |
| $T_{z_4} + \delta_{Tz_4}$ | 850.00 |
| $R_{x_4} + \delta_{Rx_4}$ | -90.00 |
| $R_{z_4} + \delta_{Rz_4}$ | 0.00 |

| Variable | Model |
|---|---|
| **Link 5** | |
| $T_{x_5}$ | 0.00 |
| $T_{z_5}$ | 0.00 |
| $R_{x_5} + \delta_{Rx_5}$ | 90.00 |
| $R_{z_5} + \delta_{Rz_5}$ | 0.00 |

| Variable | Model |
|---|---|
| **Link 6** | |
| $T_{x_6} + \delta_{Tx_6}$ | 0.00 |
| $T_{y_6} + \delta_{Ty_6}$ | 0.00 |
| $T_{z_6} + \delta_{Tz_6}$ | 100.00 |
| $R_{x_6}$ | 0.00 |
| $R_{y_6}$ | -90.00 |
| $R_{z_6}$ | 0.00 |

The error parameters are included in the model in such links that there will be no redundancies in the model. Motta (2005) and Motta and Mcmaster (1999) discuss about the choice of those parameters and about strategies to analyze the conditioning of the resultant system.

### 3.2. The IRB2000 Remote Control

Robot calibration procedures require the access to the robot TCP position coordinates and the correspondent joint values. However, the IRB2000 control unit does not show this information on the teach-pendant screen (few industrial robots will do so), but only when expensive off-line programming software produced by the manufacturer is available. Fortunately, the IRB2000 has a remote control interface that complies with those requirements (ABB, 1993). Thus, by using the remote control interface, software for remote manipulation of the IRB2000 was developed. The computer program communicates with the robot using the ADPL10 and ARAP protocols and the RS232 interface. With the help of this software the robot can be commanded to any position within its workspace and joint variables can be read from its control unit and recorded together with TCP coordinates. The software interface can be seen in Fig. 9.

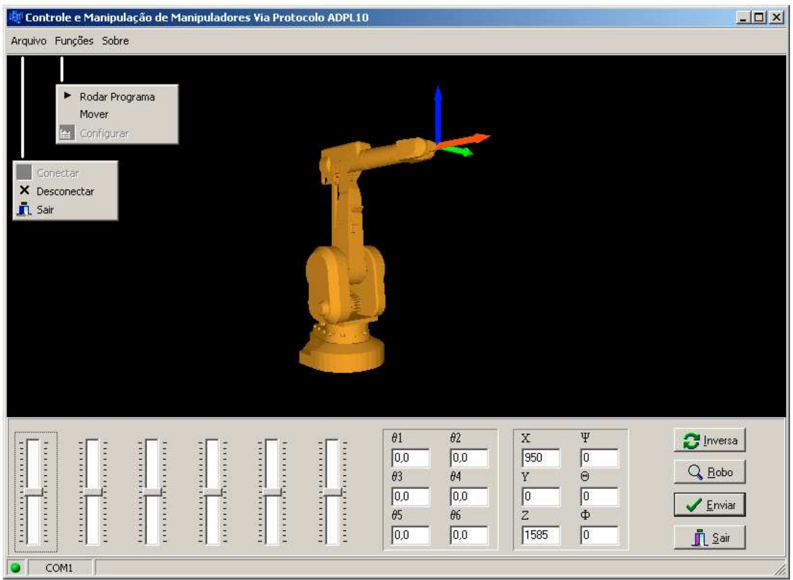


Figure 9. Remote control software

### 3.3. Control Unit Model Identification

When a robot is to be calibrated for the first time it is very important to check if the kinematic model of the off-line program is exactly the same as the robot control unit. Doing so, the kinematic model used in the off-line program can be corrected to fit exactly the nominal model used by the control unit. This procedure is needed because the nominal model used by the control unit is not accessible.

The procedure to identify the control unit kinematic model requires the robot to be moved to several positions within the workspace and that its joint variables and TCP positions are recorded. The value of $\Delta P$, in Eq. (3) (Fig. 3), can be so fully determined, where $B$ is set to null and $M$ is the TCP positions obtained from the control unit. The error parameters in the kinematic model are then modified to include only error variables related to the link dimensions in the nominal kinematic model, that is, not all error parameters in Tab. 1 are identified. Table 2 shows the identified link parameters.

Table 2. Identified error parameters for the IRB2000 controller model (units in mm)

| Error Parameter | Value |
|---|---|
| $\delta_{Tz_1}$ | 0.08 |
| $\delta_{Tx_2}$ | 0.20 |
| $\delta_{Tx_3}$ | -0.05 |
| $\delta_{Tz_4}$ | 0.05 |
| $\delta_{Tz_6}$ | 0.08 |

Those results, as expected, do not represent a considerable change in the robot model, and can be considered as numerical error due to the low resolution of the TCP position obtained from the control unit (0.125mm). Therefore those results were not incorporated to the nominal model.

### 3.4. Robot Calibration

With the mathematical model used by the robot’s control unit identified, the next step is the identification of the mathematical model that best represents the real robot. In this stage the robot is maneuvered to different positions, and those positions are measured using the measurement system. The value of each joint variable is obtained from the control unit and the position of the TCP is measured using the measurement system. So the value of all vectors, shown in Eq. (3) and Fig. 3, are known and $\Delta P$ is fully determined.

For the calibration of the IRB2000 model shown in Tab. 1 five different regions were chosen. The five chosen regions are cubes within the robot’s workspace. The central cubes have 27 positions and the external cubes have nine. These regions and the robot’s positions used for the model calibration are illustrated in Fig. 10 and Fig. 11. In those experiments the orientation of the TCP was not measured. Only TCP position data was used in the calibration process.

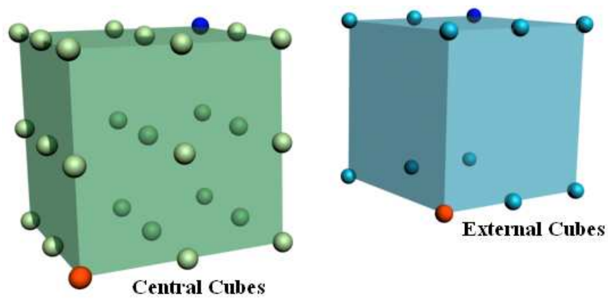


Figure 10. Cubic regions chosen and robot positions within

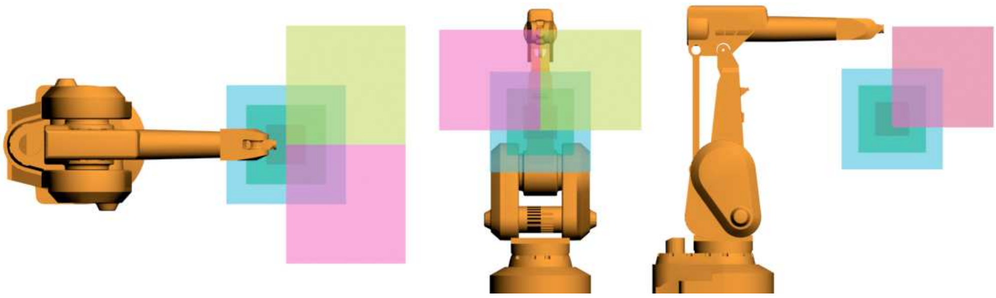

Figure 11. Calibration regions within robot's workspace

In Fig. 12 and Fig. 13 the accuracy improvement obtained through the robot calibration is shown. The values in Fig. 12 and Fig. 13 were calculated according to Eq. (8) and Eq. (9).

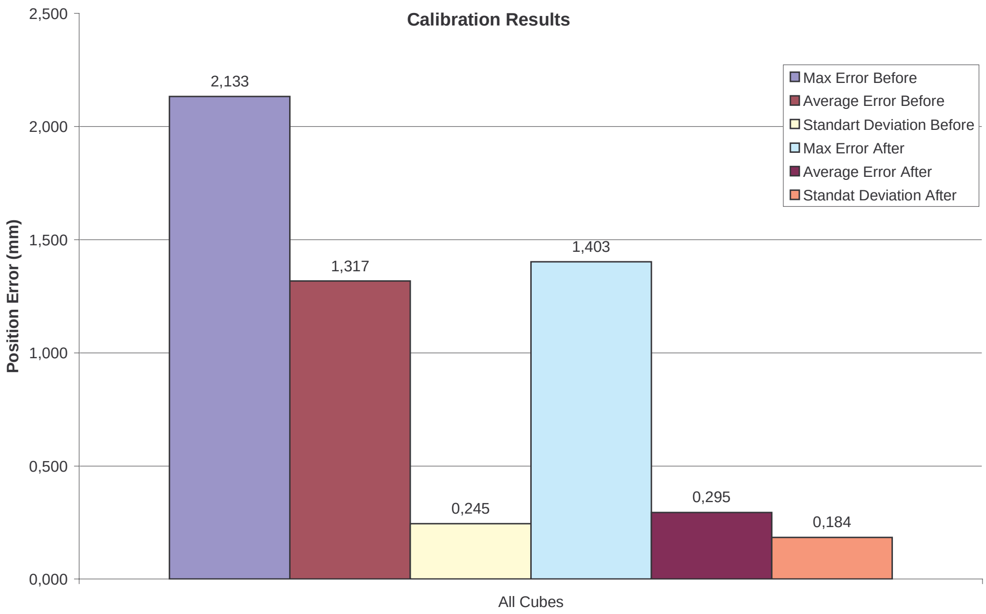


Figure 12. Errors and standard deviation after and before the calibration

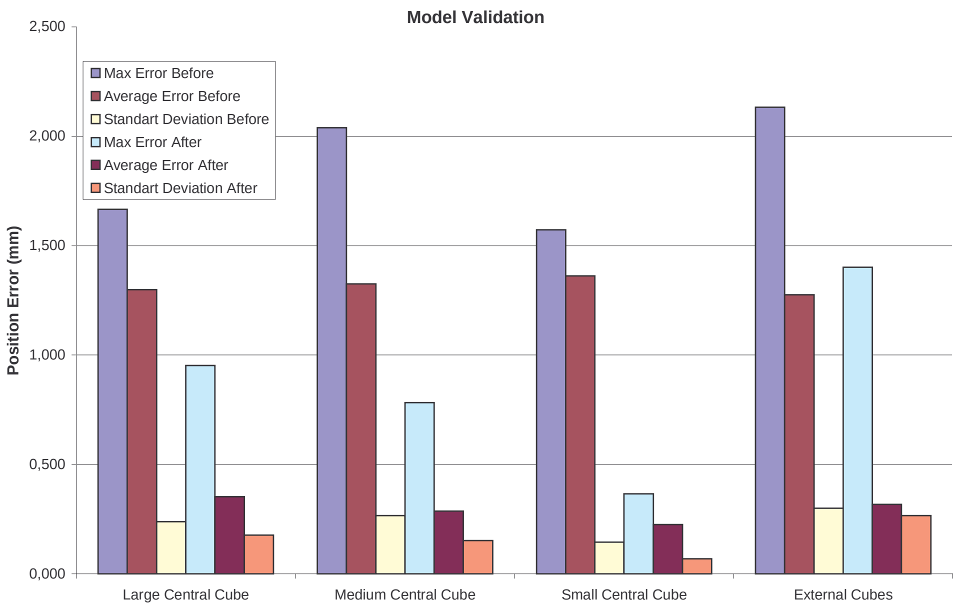


Figure 13. Evaluation of the calibrated model in each of the calibration regions

$$\overline{E} = \frac{\sum_{i=1}^{n}\left|\Delta\hat{T}_i\right|}{n} = \frac{\sum_{i=1}^{n}\left|M_i - P_i\right|}{n} \tag{8}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}\left(\left|M_i - P_i\right| - \overline{E}\right)^2}{n}} \tag{9}$$

## 4. CONCLUSIONS

A robot calibration system capable of improving the robot position accuracy has been presented. The development of this system with its mathematical background and practical considerations focusing the IRB2000 robot were discussed.

The presented system allows easy and fast calibration procedures, making the implementation of off-line programming on the shop-floor of industrial robot applications a more viable alternative.

The system was tested on an ABB IRB2000 by using an ITG ROMER measurement arm, and results showed an average improvement of the robot accuracy from 1.5 to 0.3mm. The system allows a large variation in robot configurations, which is essential to proper calibration.

## 5. REFERENCES


ABB, 1993, “Robotics Products: IRB2000”.

Bernhardt, R., Albright, S.L., 1993, “Introduction”, Robot Calibration, Ed. Chapman & Hall, pp. 3-11.

Dennis, J.E., Schnabel, R.B., 1983, “Numerical Methods for Unconstrained Optimization and Nonlinear Equations”, Ed. Prentice Hall, New Jersey, USA.

Goldenberg, A. A., Emami, M. R., “Kinematics and Dynamics of Robot Manipulators”, Handbook of Industrial Robotics, John Wiley & Sons, 2 ed., 1999, pp- 79-98.

Hollerbach, J. M., Benett, D. J., 1988, “A Survey of Kinematic Calibration”, Robotics Review, 207-242.

Jacoby, S.L.S, Kowalik, J.S., Pizzo, J.T., 1972, “Iterative Methods for Nonlinear Optimization Problems”, Ed. Prentice Hall, New Jersey, USA.

Kyle, S.A., 1993, “Non-Contact Measurement for Robot Calibration”, Robot Calibration, Ed. Chapman & Hall, pp. 79-100.

McKerrow, P.J., 1991, “Introduction to Robotics”, Ed. Addison-Wesley.

Motta, J.M.S.T., McMaster R.S., 1999, “Modeling, Optimizing and Simulating Robot Calibration with Accuracy Improvement”, J. of The Brazilian Soc. Mechanical Sciences, Vol. 21, No. 3, pp. 384-401.

Motta, J.M.S.T., 2005, “An Investigation of Singularities in Robot Kinematic Chains Aiming at Building Robot Calibration Models for Off-line Programming”, J. of The Brazilian Soc. Mechanical Sciences, Vol. 2, No. 2, pp. 200-204.

Motta, J.M.S.T., 2007, “Robot Calibration: Modeling, Measurement and Applications”. In: Low Kin-Huat. (Org.). Industrial Robotics: Programming, Simulation and Applications. 1 ed. Mammendorf - Germany: Verlag Robert Mayer-Scholz, v. 1, p. 107-130.

Press, W.H., Teukolsky, S.A., Vetterling, S.A., Flannery, B.P., 1992, “Numerical Recipes in C: The Art of Scientific Computing”, Ed. Cambridge University Press.

Schröer, K., 1993, “Theory of Kinematic Modeling and Numerical Procedures for Robot Calibration”, Robot Calibration, Ed. Chapman & Hall, pp. 157-193.


## 6. RESPONSIBILITY NOTICE