

Inverse Kinematics and Model Calibration Optimization of a Five-D.O.F. Robot for Repairing the Surface Profiles of Hydraulic Turbine Blades

Regular Paper

Jose Mauricio S.T. Motta^{1*}, Carlos Humberto Llanos-Quintero¹ and Renato Coral Sampaio¹

¹ University of Brasilia, Brasilia, DF Brazil

*Corresponding author(s) E-mail: jmmotta@unb.br

Received 25 May 2015; Accepted 13 April 2016

DOI: 10.5772/63673

© 2016 Author(s). Licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

This paper presents and discusses the results of an ongoing R&D project aiming to design and build a fully automated prototype of a specialized spherical robotic welding system for repairing hydraulic turbine surfaces eroded by cavitation pitting and/or cracks produced by cyclic loading. The system has an embedded vision sensor built to acquire range images by laser scanning over the blade's surface and produce 3D models to locate the damaged spots to be registered in a 3D coordinate system into the robot controller, enabling the robot to repair the flaws automatically by welding in layers. The paper is focused on the robot kinematic model and describes an iterative algorithm to process the inverse kinematics with only five degrees-of-freedom. The algorithm makes use of data collected from a vision sensor to ensure that the welding gun axis is perpendicular to the blade's surface. Besides this, it proposes a modelling and optimization mathematical routine for more efficient robot calibration, which can be used with any type of robot. This robot calibration optimization scheme finds the optimal error parameter vector based on the condition number of the manipulator trans-

formation composed from the partial derivatives of the error parameters. Experimental results proved both the iterative algorithm to perform the inverse kinematics and the technique to optimize robot calibration to be very efficient.

Keywords Robot Calibration, Inverse Kinematics, Turbine Blade Repairing, Welding Robots, FPGA Applications

1. Introduction

A consolidated process for repairing the blade surface of hydraulic turbines eroded by cavitation or damaged by fatigue cracks is the recovery of flaws using electric arc welding [1, 2, 3]. The welding process is performed manually after visual inspection of the blade's surface, which requires a turbine halt. These are very unfavourable conditions for human labour, with air temperatures around 40° C and 99% relative air humidity for dozens of hours.

The prototype constructed is a specialized robotic welding system for repairing damage on the surface of hydraulic

turbine blades eroded by cavitation and/or fatigue cracks under cyclic loading by welding in layers, reducing risks and increasing the efficiency of the repairing process. The robot is designed to improve the quality of the cavitation and/or fatigue damage repair, reducing the incidence of welding defects, welding material consumption, repairing time and overall repair costs. In addition, the proposed technology is expected to remove welding personnel from a harsh environment, achieve a better blade surface profile and improve welding consistency.

A robot designed with those requirements must have the ability to weld within a large range of torch orientation angles; it has to be light, small, accurate and resistant to loads from any direction on its wrist; rigid to deflections and with the potential to be fixed at any position. Some of those characteristics have opposite effects, suggesting that some of them have to be compensated by the others. Nonetheless, although a few dedicated robots have already been constructed for this type of task [4, 5, 6], this robot features improvements made to the previous ones, since it was designed to be fully automated, with minimal intervention by human operators. Additionally, this robot is capable of operating in conditions comprising large distances between blades, since the turbines in which it is to operate are 8 m in diameter and there is at least 500 mm between blades [7, 8, 9].

It is generally accepted that manipulators designed for dexterity and kinematic models' simplicity must have at least six degrees-of-freedom (d.o.f.). However, it was decided that this robot's dexterity could be limited to five d.o.f. in order to reduce weight and size and to increase portability. This solution can be justified considering that the welding torch does not need three orientation angles within the 3D space, since the torch has a cylindrical symmetry.

It is well known that welding tasks are generally performed with a quasi-constant torch orientation during long displacements when welding. Thus, the robot can track a seam for welding the beads with a small variation in one of its Euler angles, mostly rotation around the torch symmetry axis. Several approaches for finding inverse kinematic solutions for a five-d.o.f. serial manipulator have been published in the past few years [10, 11, 12, 13, 14], making use of either analytical or numerical methods, normally specific to a certain robot model or topology.

Analytical solutions for the inverse kinematics for high-degree-of-freedom robots are very difficult or impossible. A complete analytical formulation for the inverse kinematics of the five-d.o.f. Pioneer 2 robotic arm (P2Arm) was presented in [10], but the correct solution could only be found after testing some partial solution alternatives. Some of the former authors [11] proposed a different analytical solution for the same robot but using some geometrical restriction during operation for tracking a given trajectory, while keeping the orientation of one axis in the end-effector

frame. However, different partial solutions had still to be checked before the correct solution was found. More recently, it was presented in [13] an analysis of the inverse kinematics of a five-d.o.f. robot, the Mitsubishi Melfa RV-2AJ industrial robot, aiming at controlling the z-axis position only.

Other authors [14] presented a generalized solution for the five-d.o.f. revolute joint variables of a machine comprising 2-links and a spade-like three-d.o.f. end-effector obtained by solving a set of algebraic equations emerging from series of transformation matrices.

An iterative approach can be found in [15] for solving inverse kinematics by adding a virtual joint to the five-d.o.f. robot, expressing all joints by one variable and applying the one-dimensional iterative Newton-Raphson method to minimize the tip-position error.

In [12], an iterative algorithm was proposed for a five-d.o.f. welding robot with a wrist offset. The algorithm showed good results when the position and three orientation angles were given and tracks were performed, keeping that orientation constant.

The algorithm proposed here for our spherical robot, with no wrist offsets but the inclusion of a shoulder, requires a torch orientation vector, expressed only from two angles, as a 3D vector parallel to the torch symmetry axis assigned to every track point.

Other than the inverse kinematics problem, one very important procedure when building a complex robot from scratch is to perform a full robot calibration routine to assure that the kinematic model corresponds to the actual robot, since there is no previous nominal model. As with every robot, every time it is necessary to disassemble parts of that robot, the calibration procedure has to be re-executed. Once a robot has been completely calibrated, it is not necessary to recalibrate all the model parameters but only the ones related to the link that has been disassembled. For example, if a torch is to be changed or disassembled, it is necessary to calibrate only the torch parameters. Another important issue in our robot is that the vision system may be considered as another link belonging to the robot's kinematic chain, since all measurements have to be assigned to the robot base coordinate system.

Our robot can operate with online or offline programming modes. Every time the vision system is used to generate a map and transmit point coordinates to the robot controller, it works in an offline programming mode. However, one of the obstacles that makes the viability of offline programming difficult is the poor accuracy of robot static positioning, turning robot calibration into an important procedure.

During the past few decades many different robot calibration methods have been published. The large majority of them compensate for position errors in the kinematic model [16, 17, 18] and few are considered modelless methods that use mathematical regression modelling of the errors or neural networks, but with the disadvantage of

needing a large number of sample measurements [19, 20, 21]. Currently, robot calibration is still an active area of research [22].

The robot calibration optimization method presented in this article comprises kinematic parameter modelling, since the control system is open and easily changed. An important contribution of this article is to propose a method to define which geometric parameters have to be included in the model and, more importantly, to improve the mathematical conditioning of the system. The results of the application of this optimization technique are to reduce the number of measurement points and to improve the positional accuracy compared to a non-optimized calibration model. There are no works presented to date showing results of a method like this one with experimental verification.

This article presents initially and briefly, in Sections 2 and 3, the entire robotic system and its kinematic model. Section 4 presents the iterative algorithm developed to perform the inverse kinematics, making use of data collected from a vision sensor to ensure that the welding gun's axis is perpendicular to the blade surface and to collect the results of point tracking. In Section 5, a brief revision of robot calibration theory is discussed. Section 6 proposes and discusses the mathematical routine to determine the optimal parameter set to be identified by the robot calibration process, based on the analysis of the condition number of the manipulator transformation partial derivatives of the error parameters. Section 7 presents experimental results and proves the success of the solutions.

2. Robot Characteristics

The robotic system prototype constructed has a spherical topology with five d.o.f., a pan-tilt wrist, electric stepper motors, rotary and linear actuators. The robot arm carries an embedded vision sensor for acquiring range images and modelling 3D structured surface maps. The maps are used to plan welding trajectories in 3D coordinates automatically, according to the welding strategy on the surface blade region to be welded. The measurement system is implemented via a mini-PC that is embedded in the system. The robot controller and welding controller systems are built on a reconfigurable architecture based on FPGA (Field Programmable Gate Arrays). The welding process type is the GMAW (Gas Metal Arc Welding), with a pulsed arc welding machine and tubular metal cored electrodes. The robot was designed to have easy assembly and fixation on the blades (either belts or magnetic paddles), high rigidity mechanics and trouble-free operation. Portability, low cost, lightweight, good repeatability and high positioning accuracy are features of the resulting robot [8].

2.1 Robot prototype

The robot constructed weights 30kg, has a spherical topology and a 2.0m outer diameter workspace with dimensions of 30 x 25 x 60-100cm without the welding torch. Figure 1 shows the constructed system.

The robot was designed to embody an integrated control system for managing several tasks to be carried out automatically. The control system was designed to manage the robot's actions, such as (a) environment recognition, (b) calculation of volumes and location to be filled with weld beads, (c) slicing the volumes by using welding strategies to weld in layers, (d) calculating trajectories for the welding torch and (e) managing the robot motion according to the welding needs.

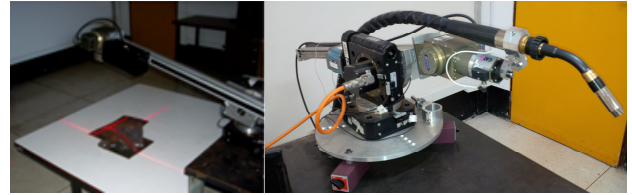


Figure 1. Robot prototype

3. Kinematic System

3.1 Assignment of joint coordinate frames

For the construction of the forward kinematic model, joint coordinate systems were assigned in such a way that the Z_n axes point towards the joint axis of motion, and that the vectors X_n are all parallel when the robot is in its zero position. The Y_n axes are orthogonal to the former ones, creating an orthonormal system of axes.

The origin of the base frame coincides with the origin of the first joint, whose axis is perpendicular to the x-y plane. The origins of the other joint frames are placed as follows: (a) if the joint axes of a link intersect, then the origin of the frame fixed to the link is placed at the intersection of the joint axes; (b) if the joint axes of a link are parallel or do not intersect, then the origin of the frame fixed to the link is placed at the distal joint. Thus, the coordinate frame i is placed at joint $i + 1$, i.e., the joint that connects link i to link $i+1$; (c) if a frame origin is described relative to another coordinate frame including more than one direction, then it must be moved in order to have its relative position described by only one direction, if possible. Thus, the origins of the coordinate frames are described using a minimum number of link parameters [23].

A coordinate frame is attached to the end-effector such that the z-axis of the frame has the same direction as the z-axis of the frame placed at the last joint (n-1). For this robot, it is also necessary to have a separate transformation for this last frame, which is fixed at the welding torch tip, to ensure that its z-axis is parallel to the axis of symmetry of the torch nozzle, ensuring a proper control of the welding torch's orientation.

The nomenclature used to represent link length parameters has an index for joints and direction. The p_{ki} length is the distance between the origins of the coordinate frames $i-1$ and i , and k is the axis of the coordinate frame $i-1$ that is

parallel to the length direction. Figure 2 shows the previous rules applied to the robot constructed with all coordinate frames, geometric parameters and link variables.

3.2 Forward kinematic model

Homogeneous transformation matrices relating coordinate frames from the base (b) to the torch/tool (t) can be derived as follows:

$${}^b_iT = {}^b_0T * {}^0_1T * {}^1_2T * {}^2_3T * {}^3_4T * {}^4_5T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where ${}^i_{i+1}T$ is the homogeneous transformation between two successive joint coordinate frames. The welding torch is attached to Link 4 and its geometric parameters are shown in Figure 3.

The transformations shown in Eq. (1) can be described using the Denavit-Hartenberg (D-H) convention, as below:

$${}^{i-1}_iT = Rz(\theta) * Tz(d) * Tx(l) * Rx(\alpha) = \begin{bmatrix} \cos(\theta) & -(\cos(\alpha) * \sin(\theta)) & \sin(\alpha) * \sin(\theta) & l * \cos(\theta) \\ \sin(\theta) & \cos(\alpha) * \cos(\theta) & -(\cos(\alpha) * \sin(\theta)) * l * \sin(\theta) & \\ 0 & \sin(\alpha) & \cos(\alpha) & d \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad (2)$$

where θ and α are rotation parameters and d and l are translation parameters. The application of Eq. (2) to each of the consecutive robot joint frames by using the geometric parameters shown in Figures 2 and 3 produces the following homogeneous transformations:

$${}^0_1T = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & \cos(\theta_1) & 0 \\ 0 & -1 & 0 & pz1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$${}^1_2T = \begin{bmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) & -px2 * \cos(\theta_2) \\ \sin(\theta_2) & 0 & -\cos(\theta_2) & -px2 * \sin(\theta_2) \\ 0 & 1 & 0 & pz2 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$${}^2_3T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & pz3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$${}^3_4T = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$${}^4_5T = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & px5 * \cos(\theta_5) \\ \sin(\theta_5) & \cos(\theta_5) & 0 & px5 * \sin(\theta_5) \\ 0 & 0 & 1 & pz5 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

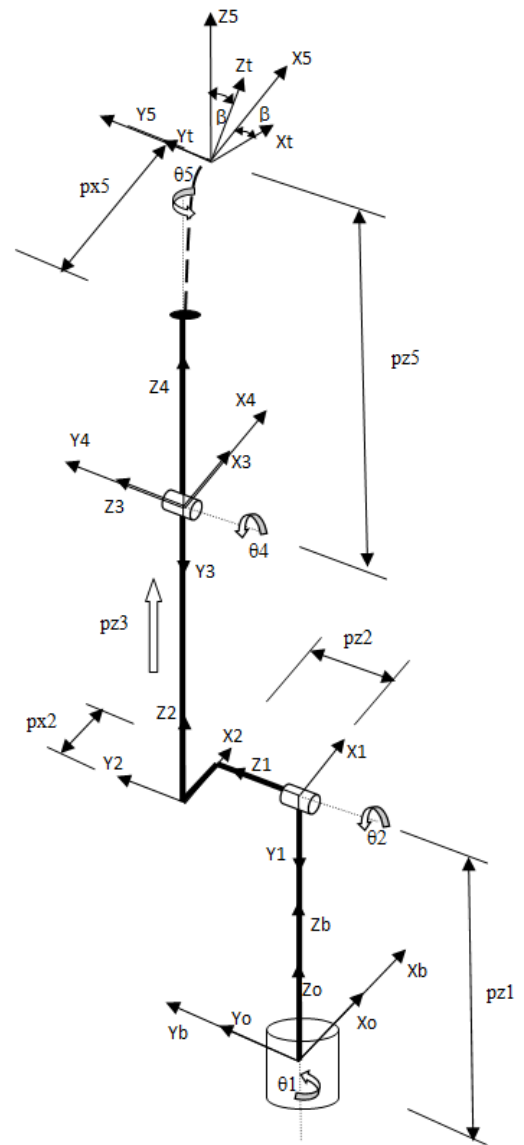


Figure 2. Robot at zero position with joint coordinate systems and link variables

The next and final homogeneous transformation rotates the Coordinate System 5 around the y_5 vector, consistent with

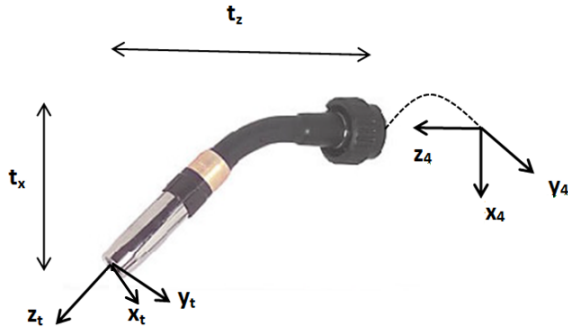


Figure 3. Welding torch coordinate frame (x_t, y_t, z_t) and related geometric parameters (t_x, t_z)

the geometry of the torch used. Its application is justified only for the Z_t axis to be aligned with the torch axis of symmetry.

$${}^5_tT = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

The elements of the general manipulator transformation, 0_5T , according to Eq. (1), excluding the rotation of the torch tip coordinate frame by the angle β , are formulated below, as for the robot forward kinematic equations.

$$nx = -\sin(\theta_1) \cdot \sin(\theta_5) + \cos(\theta_5) \cdot \cos(\theta_1) \cdot \cos(\theta_2 + \theta_4) \quad (9)$$

$$ox = -\sin(\theta_1) \cdot \cos(\theta_5) - \sin(\theta_5) \cdot \cos(\theta_1) \cdot \cos(\theta_2 + \theta_4) \quad (10)$$

$$ax = \cos(\theta_1) \cdot \sin(\theta_2 + \theta_4) \quad (11)$$

$$\begin{aligned} px &= pz3 \cdot \cos(\theta_1) \cdot \sin(\theta_2) + \\ &+ px2 \cdot \sin(\theta_1) \cdot \sin(\theta_2) - pz2 \cdot \sin(\theta_1) + \\ &+ pz5 \cdot \cos(\theta_1) \cdot \sin(\theta_2 + \theta_4) + \\ &+ px5 \cdot [\cos(\theta_1) \cdot \cos(\theta_2 + \theta_4)] - \sin(\theta_1) \cdot \sin(\theta_5) \end{aligned} \quad (12)$$

$$ny = \cos(\theta_1) \cdot \sin(\theta_5) + \cos(\theta_5) \cdot \sin(\theta_1) \cdot \cos(\theta_2 + \theta_4) \quad (13)$$

$$oy = \cos(\theta_1) \cdot \cos(\theta_5) - \sin(\theta_5) \cdot \sin(\theta_1) \cdot \cos(\theta_2 + \theta_4) \quad (14)$$

$$ay = \sin(\theta_1) \cdot \sin(\theta_2 + \theta_4) \quad (15)$$

$$\begin{aligned} py &= pz3 \cdot \sin(\theta_1) \cdot \sin(\theta_2) - \\ &- px2 \cdot \sin(\theta_1) \cdot \cos(\theta_1) + pz2 \cdot \cos(\theta_1) + \\ &+ pz5 \cdot \sin(\theta_1) \cdot \sin(\theta_2 + \theta_4) + \\ &+ px5 \cdot [\cos(\theta_1) \cdot \sin(\theta_5) + \cos(\theta_5) \cdot \sin(\theta_1) \cdot \cos(\theta_2 + \theta_4)] \end{aligned} \quad (16)$$

$$nz = -\sin(\theta_2 + \theta_4) \cdot \cos(\theta_5) \quad (17)$$

$$oz = \sin(\theta_2 + \theta_4) \cdot \sin(\theta_5) \quad (18)$$

$$az = \cos(\theta_2 + \theta_4) \quad (19)$$

$$\begin{aligned} pz &= pz1 + pz5 \cdot \cos(\theta_2 + \theta_4) + pz3 \cdot \cos(\theta_2) + \\ &+ px2 \cdot \sin(\theta_2) - px5 \cdot \sin(\theta_2 + \theta_4) \cdot \cos(\theta_5) \end{aligned} \quad (20)$$

3.3 Inverse kinematic model

It is supposed that the vision sensor provides the normal vector to the surface to be welded, Z , at the point to be reached by the welding torch, represented in the robot base coordinate frame. Assuming that the Z_t vector is parallel to the torch axis of symmetry and points outwards, as is shown in Figures 2 and 3, one wants to establish the triplet $\Omega_t = [X_t, Y_t, Z_t]$ on the welding torch tip so that the torch reaches the desired position coordinate point, $P=[P_x, P_y, P_z]$, with the torch tip Z_t axis oriented in the direction of the surface normal vector, Z .

According to the kinematic model in Figure 2, with θ_5 at its zero position, the vectors X_t and Z_t belong to a plane π , shown in Figure 4a, passing through the torch tip and Links 4 and 3.

It is assumed that Z is a unit vector. Thus, $Z_t = -Z$ and, according to Figures 4a and 4b, one can calculate the vector V , belonging to the plane π , as:

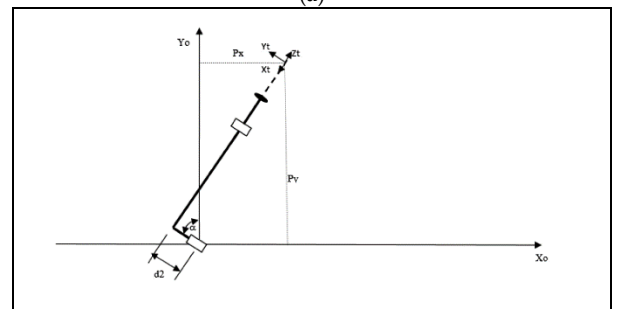
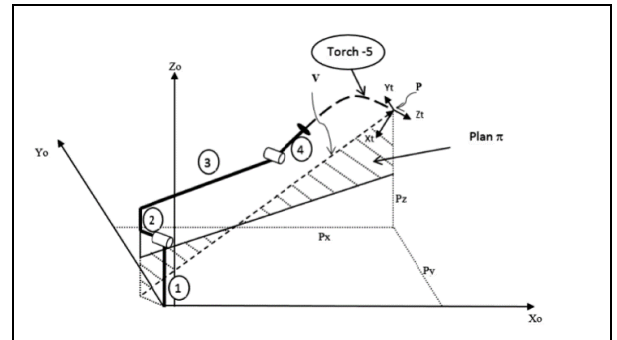


Figure 4. The robot position with the torch coordinate system oriented when the angle $\theta_5 = 0$

$$\mathbf{V} = [\mathbf{Px} + d2 \cdot \sin(\alpha), \mathbf{Py} - d2 \cdot \cos(\alpha), \mathbf{Pz}],$$

in which $\alpha = \text{atan}(\mathbf{Py} / \mathbf{Px}) - \text{asin}(d2 / \|\mathbf{P}\|)$.

The triplet $\mathbf{\Omega t} = [\mathbf{Xt}, \mathbf{Yt}, \mathbf{Zt}]$ can be calculated from:

$$\mathbf{Yt} = (-\mathbf{Z})x(-\mathbf{V}); \mathbf{Xt} = (-\mathbf{Z})x(-\mathbf{Yt}); \mathbf{Zt} = -\mathbf{Z} \quad (21)$$

The vectors \mathbf{Xt} and \mathbf{Yt} are therefore calculated assuming that \mathbf{Xt} belongs to the plane π , i.e., $\theta5 = 0$. However, for a solution for the triplet $\mathbf{\Omega t}$ for any non-zero value of $\theta5$, \mathbf{Xt} and \mathbf{Yt} calculated from Eq. (21) are close to the solution and will be used as initial values in a simple iterative method to be described later in this article.

A general orientation transformation \mathbf{Hp}_0 can be formulated as:

$$\mathbf{Hp}_0 = \begin{bmatrix} Xt_x & Yt_x & Zt_x & P_x \\ Xt_y & Yt_y & Zt_y & P_y \\ Xt_z & Yt_z & Zt_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix}, \quad (22)$$

whose first three columns determine the components of the vectors \mathbf{Xt} , \mathbf{Yt} , \mathbf{Zt} in the robot base coordinate frame, respectively.

The robot's inverse kinematics provide all the joint values in Eqs. (9) to (20). However, Eq. (1) is not equal to Eq. (22), except in the case of $\theta5 = 0$.

Particularly, the equations of the robot's inverse kinematics can be found geometrically. For that purpose, the torch coordinate frame should be rotated by an angle $-\beta$ around \mathbf{Yt} , in such a way that the \mathbf{Zt} axis retrieves its orientation collinear to the $\mathbf{Z5}$ vector. The vector system $(\mathbf{X5}, \mathbf{Y5}, \mathbf{Z5})$ at the torch tip becomes parallel to the vector system $(\mathbf{X4}, \mathbf{Y4}, \mathbf{Z4})$ of the flange. Therefore,

$${}^0T = \mathbf{Hp}_0 * {}^5T^{-1} = \mathbf{Hp}_0 * {}^tT \quad (23)$$

The transformation tT produces an elementary rotation around the \mathbf{Yt} axis.

Equations (17) and (18) can be manipulated algebraically to make $\theta5$ explicit. There are two solutions for $\theta5$:

$$\theta5_1 = \tan^{-1}\left(\frac{Yt_z}{Xt_z}\right); \theta5_2 = \tan^{-1}\left(-\frac{Yt_z}{Xt_z}\right) \quad (24)$$

If the component of the product $(\mathbf{V}) \times (\mathbf{Z})$ in the \mathbf{Zb} direction is positive, the solution chosen for $\theta5$ is negative; if it is negative, the solution chosen is positive.

From Eqs. (11) and (15), and by using the orthonormality of the rotation matrix below:

$$n_z^2 + o_z^2 + a_z^2 = 1 \text{ and } a_x^2 + a_y^2 + a_z^2 = 1 \quad (25)$$

$\theta1$ can be made explicit as:

$$\theta1_1 = -2 * \tan^{-1} \left[\frac{\sqrt{2 * (n_z^2 + o_z^2) - (a_x^2 + a_y^2)} - \sqrt{n_z^2 + o_z^2}}{a_x + a_y + \sqrt{n_z^2 + o_z^2}} \right] \quad (26)$$

$$\theta1_2 = 2 * \tan^{-1} \left[\frac{\sqrt{2 * (n_z^2 + o_z^2) - (a_x^2 + a_y^2)} - \sqrt{n_z^2 + o_z^2}}{a_x + a_y + \sqrt{n_z^2 + o_z^2}} \right], \quad (27)$$

The equations above present a singularity for $a_z = \mp 1$, i.e., $n_z = o_z = a_x = a_y = 0$, where the flange $\mathbf{Z4}$ axis is collinear with the robot base \mathbf{Zb} axis. This particular robot configuration can be easily avoided.

The correct solution for $\theta1$ must be chosen from the sign of the z components of the normal vector to the plane π formed by the vectors \mathbf{P} and \mathbf{Z} .

Closed equations for the joint variables $\theta4$, $\theta2$ and $pz3$ cannot be easily formulated from the robot's general transformation matrix, but they can be expressed using geometry. From Eqs. (23) and (7), one can determine the position of the origin of the coordinate frame 4, $(\mathbf{Px4}, \mathbf{Py4}, \mathbf{Pz4})$, through the equation:

$${}^0T = {}^0T * {}^4T^{-1} = {}^0T * {}^5T, \quad (28)$$

where the origin position is described from the entries of the fourth column of the 0T transformation.

From Figures 4a and 4b, $pz3$ can be obtained as:

$$pz3 = \sqrt{l^2 + (Pz4 - pz1)^2 - px2^2} \quad (29)$$

in which $l = \sqrt{(Px4^2 + Py4^2 - pz2^2)}$

The variable $\theta2$ can be formulated from:

$$\theta2 = \pi - \sin^{-1} \left(\frac{pz3}{\sqrt{l^2 + (Pz4 - pz1)^2}} \right) - \sin^{-1} \left(\frac{Pz4 - pz1}{\sqrt{l^2 + (Pz4 - pz1)^2}} \right) \quad (30)$$

The variable $\theta4$ can be obtained from:

$$\theta4 = \cos^{-1}(a_z) - \theta2 \quad (31)$$

It is clear from the equations above that θ_5 , θ_1 and $(\theta_4+\theta_2)$ only carry information from the torch orientation, while p_{z3} , θ_2 and θ_4 define its position.

4. Algorithm for Searching the Correct Inverse Kinematics Solution

Equations (24), (26), (27), (29), (30) and (31) provide an explicit solution for the inverse kinematics, assuming proper values for the welding-torch orientation vectors, \mathbf{Xt} and \mathbf{Yt} . However, a robot with five d.o.f. cannot move its end-effector/tool to any specified pose within the 3D space, which means that one of the three orientation angles cannot be chosen freely. Equation (1) cannot be solved for any robot's pose chosen in Eq. (21). In other words, once a torch orientation vector, \mathbf{Zt} , is chosen, the first two columns of the rotation transformation matrix, \mathbf{R} , in Eq. (22), cannot be known in order to produce the inverse kinematics solution that will move the welding torch to the correct pose $\mathbf{P}=(\mathbf{Px}, \mathbf{Py}, \mathbf{Pz})$.

A solution to this problem is the use of an iterative method that varies \mathbf{Xt} and \mathbf{Yt} in Eq. (22) until the solution of Eq. (1) converges to the correct pose with the desired \mathbf{Zt} .

4.1 Algorithm description

Since a welding torch has a cylindrical geometry, one can assume that the rotation around its symmetry axis does not need to be specified (around \mathbf{Zt}). Thus, the orientation angle of the orthonormal vectors, \mathbf{Xt} and \mathbf{Yt} , cannot be specified, but it can be found numerically as a function of the other components of the robot's kinematic model. Then, an initial value for the angle ϕ (around \mathbf{Zt}) can be specified as a function of the joint angles θ_5 and θ_1 (they depend only on the general orientation), which are independent of the torch position, \mathbf{P} . Variables θ_2 , θ_4 and p_{z3} depend on the torch orientation and also on position.

The algorithm principle is based on the calculation of the relative angle, ϕ , between two 2D rotation matrices. An arbitrary reference angle is the guess for the triplet $\mathbf{\Omega t}$, and the other is calculated from the equations of the inverse and forward kinematics. The algorithm starts with the choice of an initial value for the \mathbf{Xt} and \mathbf{Yt} vectors in Eq. (21), assuming \mathbf{Zt} is provided by the vision sensor (normal to the surface). The second matrix is calculated from the forward kinematics using the joint variable values obtained from the inverse kinematics (Eqs. (9-20)). Therefore, the equation below can be formulated:

$$\begin{bmatrix} Xt_x & Yt_x \\ Xt_y & Yt_y \\ Xt_z & Yt_z \end{bmatrix} = \begin{bmatrix} n_x & o_x \\ n_y & o_y \\ n_z & o_z \end{bmatrix} * \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix} \quad (32)$$

The matrices above are orthonormal. Two properties of orthonormal matrices ensure that the scalar product

between two columns/rows is null and that the norms of the columns/rows are unitary. Then, it can be easily demonstrated that:

$$Xt_x.o_x + Xt_y.o_y + Xt_z.o_z = \sin\phi \quad (33)$$

$$Xt_x.n_x + Xt_y.n_y + Xt_z.n_z = \cos\phi \quad (34)$$

$$\phi = \tan^{-1} \frac{Xt_x.o_x + Xt_y.o_y + Xt_z.o_z}{Xt_x.n_x + Xt_y.n_y + Xt_z.n_z}, \quad (35)$$

where ϕ is the relative rotation angle of the pair $[\mathbf{Xt}, \mathbf{Yt}]$ around the \mathbf{Zt} axis calculated from the two matrices. The proposed algorithm can be established in the following steps:

1. The torch orientation, $\mathbf{\Omega t} = [\mathbf{Xt}, \mathbf{Yt}, \mathbf{Zt}]$, is initially supposed according to Eq. (21). The torch position vector, $\mathbf{P}=[\mathbf{X}, \mathbf{Y}, \mathbf{Z}]$, represented in the robot base coordinate frame is known and \mathbf{Zt} is obtained from the vision sensor.
2. Equation (23) is used to rotate the triplet $\mathbf{\Omega t}$ around \mathbf{Yt} , aligning the torch tip coordinate frame with the coordinate system 5, $\mathbf{\Omega}_5$, linked to the robot flange. This keeps the system $\mathbf{\Psi}_t = [\mathbf{P}, \mathbf{\Omega}_5]$ on the torch tip.
3. Equations (24-31) are used to determine the joint variable values, $[\mathbf{\Theta}]$, corresponding to the torch pose $\mathbf{\Psi}_t$.
4. Equations (9-20) are applied to calculate the forward kinematics, finding the robot position, \mathbf{Pt} , with the joint coordinates, $[\mathbf{\Theta}]$, calculated in Step 3.
5. The difference between the position specified, \mathbf{P} , in Step 1, and the position calculated in Step 4, \mathbf{Pt} , is found. This difference establishes a position error $\Delta\mathbf{Pt}$. The difference between the torch orientation $\mathbf{\Omega t}$ and the new orientation $\mathbf{\Omega t'}$, around \mathbf{Zt} , is the angle ϕ , which can be calculated from Eq. (35). This difference shall be null with this proposed robot.

Once the vectors \mathbf{P} , \mathbf{Pt} , $\Delta\mathbf{Pt}$ and $\mathbf{\Omega t}$ are available, a routine is applied to update the orientation triplet, $\mathbf{\Omega t}$, in such a way that the values of $\Delta\mathbf{Pt}$ are reduced to a minimum specified value, ϵ , upon which $\mathbf{\Omega t}$ would provide the correct orientation of the torch. The rule for updating $\mathbf{\Omega t}$ is to consider that the deviation of ϕ from its correct value is proportional to the Euclidean error, $e_k = |\mathbf{E}_k|$, in which $\mathbf{E}_k = \Delta\mathbf{Pt} = \mathbf{P} - \mathbf{Pt}_k$, between the torch tip position and its desired position, \mathbf{P} , in each iteration k , as shown in Figure 5.

It is assumed that each value of ϕ_k in an iteration corresponds to a \mathbf{E}_k position error. Figure 6 represents, graphically, each value of ϕ_k in successive iterations until the solution is achieved. The proportional relationship be-

tween ϕ_k and e_k can be established graphically as shown in Figure 6 as:

$$\frac{\beta}{\beta + \phi_1} = \frac{e_0}{\mp e_1} \Rightarrow \frac{\phi_{k+1}}{\phi_{k+1} + \phi_k} = \frac{e_{k-1}}{\pm e_k} \quad (36)$$

Therefore,

$$\phi_{k+1} = \frac{e_{k-1}}{e_{k-1} - \text{sign}[E_{k-1}, E_k] e_k} \phi_k \quad (37)$$

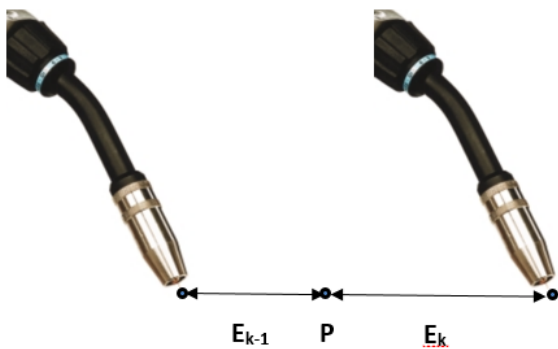


Figure 5. Positioning error of the welding torch in each iteration k

Equation (36) provides a new rotation value ϕ_{k+1} , $\forall k$: ($1 \leq k \leq n$), in each iteration k, until the error e_{n+1} between P and $P_{t_{n+1}}$ is smaller than ϵ . The final value for ϕ , the angle γ , is measured between the orientation of the torch coordinate system proposed by Eq. (1) and the correct one. For this robot, as soon as the algorithm stops at the solution, Eq. (35) provides a value for γ that is numerically approximate to the solution for θ_5 .

4.2 Results of the inverse kinematics algorithm

To check the algorithm's efficiency in order to calculate the inverse kinematic iteratively, different robot poses were inputted. All cases have demonstrated that the algorithm converges to the solution with three to four iterations. The input variables are the position of the welding point to be welded and the normal vector to the surface obtained from the vision sensor (Z), both represented in the robot base frame. The torch orientation must be the same as the normal vector.

The robot dimensions were determined as shown in Table 1, after calibration, discussed in subsequent sections.

| pz1 (mm) | pz2 (mm) | pz5 (mm) | px2 (mm) | px5 (mm) | β (°) |
|----------|----------|----------|----------|----------|----------------|
| 275 | 36.31 | 476.9 | 35.94 | 18.98 | 45 |

Table 1. Robot geometric parameters, according to Figure 2, after model calibration

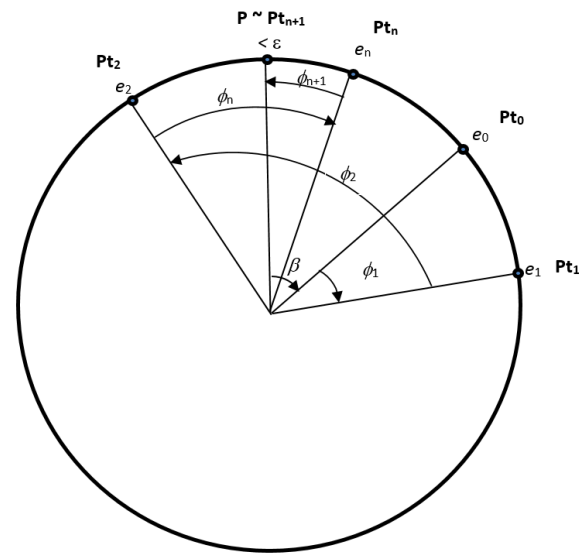


Figure 6. Graphical demonstration of the rotation of X_t and Y_t around the Z_t axis by an angle ϕ , in successive iterations. For each position P_t attained, the error and the angle ϕ are shown. The value of ϕ is initially unknown ($= \beta$).

Three cases of position data in three different quadrants of the robot workspace and torch orientation were inputted into the iterative algorithm as shown in Table 2. Results show the convergence of the joint values and the algorithm parameters.

The chosen value for ϵ was 0.15 mm, but it can of course be higher. This error threshold most of the time interrupts the algorithm typically in the third iteration. Of course, the poses were chosen to be feasibly reached by the torch. Some indetermination could be found when θ_1 was close to 45° , 135° , 235° and 315° , but the singularity can be easily avoided with a forward and backward coordinate transformation. The robot was tested at many different positions and orientations during welding and no singularities were found. Z is a unitary vector, normal to the plate surface and pointing from the plate outwards.

The results of the inverse kinematics were used in the forwards kinematic model to check the accuracy of the algorithm, and the results are shown in Table 3. Physical measurements of the tracking process are not always feasible and were not performed, furthermore the position accuracy depends on that of the kinematic model after calibration. This matter will be discussed in the following sections. Figure 7 shows a weld bead welded by the robot, with a speed of 10mm/s in a straight line. Details of the control system will not be presented in this article.

5. The Robot Calibration System

Robot calibration is the process of improving the robot's accuracy by modifying its nominal kinematic model in the robot controller. Robot calibration can be divided into two main groups: static and dynamic calibration [24].

| Track Point 1 – P1 | | | | |
|--------------------|----------------------|---------|---------|-------------------------------|
| | $P=[560 \ 350 \ 10]$ | | | $Z=[-0.099 \ -0.099 \ 0.990]$ |
| Iteration | 0 | 1 | 2 | 3 |
| e(mm) | 21.095 | 31.359 | 0.356 | 0.110 |
| θ_1 (deg) | 30.597 | 31.493 | 28.793 | 28.754 |
| θ_2 (deg) | 89.462 | 89.508 | 89.377 | 89.377 |
| d_3 (mm) | 290.452 | 290.057 | 291.895 | 291.930 |
| θ_4 (deg) | 37.773 | 37.691 | 37.928 | 37.928 |
| θ_5 (deg) | 2.823 | 2.653 | 3.168 | 3.173 |
| ϕ (deg) | -0.998 | 3.030 | 0.026 | -0.019 |

| Track Point 2 – P2 | | | | | |
|--------------------|------------------------|--------|--------|-----------------------------|--------|
| | $P=[-500 \ -180 \ 50]$ | | | $Z=[0.318 \ 0.424 \ 0.848]$ | |
| Iteration | 0 | 1 | 2 | 3 | 4 |
| e(mm) | 88.317 | 138.79 | 11.95 | 0.166 | 0.002 |
| θ_1 (deg) | -155.44 | -150.4 | -163.8 | -165.1 | -165.1 |
| θ_2 (deg) | 142.98 | 137.68 | 132.21 | 128.11 | 128.05 |
| d_3 (mm) | 101.49 | 129.92 | 88.571 | 92.01 | 92.076 |
| θ_4 (deg) | -34.80 | -31.14 | -20.51 | -15.82 | -15.75 |
| θ_5 (deg) | 20.947 | 17.434 | 26.779 | 27.609 | 27.619 |
| ϕ (deg) | -7.04 | 19.359 | 1.826 | 0.027 | 0.002 |

| Track Point 3 – P3 | | | | |
|--------------------|------------------------|---------|---------|------------------------------|
| | $P=[900 \ -550 \ -10]$ | | | $Z=[-0.577 \ 0.577 \ 0.577]$ |
| Iteration | 0 | 1 | 2 | 3 |
| e(mm) | 38.970 | 27.951 | 0.211 | 0.065 |
| θ_1 (deg) | -30.998 | -34.704 | -33.146 | -33.156 |
| θ_2 (deg) | 122.883 | 123.284 | 123.154 | 123.157 |
| d_3 (mm) | 665.745 | 671.615 | 668.926 | 668.939 |
| θ_4 (deg) | -40.636 | -41.954 | -41.473 | -41.479 |
| θ_5 (deg) | -16.224 | -11.909 | -13.721 | -13.709 |
| ϕ (deg) | 5.351 | -2.234 | 0.017 | 0.009 |

Table 2. Variable joint values and geometric parameters at each iteration of the algorithm, showing convergence

Static calibration systems calibrate geometric parameters, such as joint positions and axis misalignments, and also non-geometric parameters, such as link and joint elasticity, gear errors (eccentricity and transmission errors), gear backlash and thermal expansion. Both geometric and non-geometric parameters can be included in static calibration models because these parameters can be identified only using robot pose data. Although there are some publications [25, 26] related to non-geometric parameter calibration, these extra parameters increase significantly the complexity of the model. It has been reported [27, 28] that the different types of parameters follow an order of quantitative importance for improving the accuracy of kinematic models: geometrical parameters, joint elasticity and link elasticity. Transmission and coupling (i.e., gear

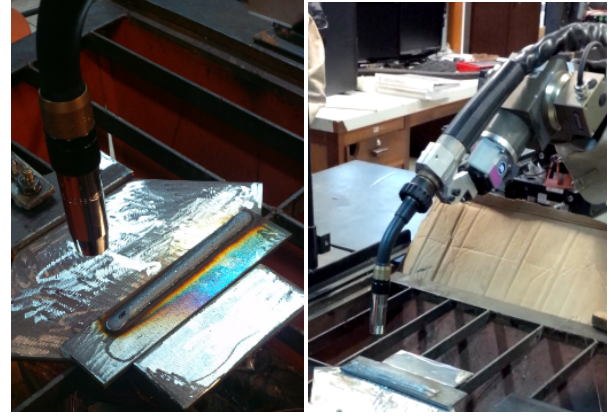


Figure 7. Weld beads, welded by the robot at a speed of 10mm/s

| Position and Orientation Errors | | | |
|---------------------------------|--------------|-----------|-----------|
| Position | Track Points | | |
| | P1 | P2 | P3 |
| Px (mm) | 560.0931 | -560.0016 | 899.9818 |
| Py (mm) | 349.9404 | -179.9998 | -549.9381 |
| Pz (mm) | 9.9991 | 49.9997 | -9.9994 |
| ΔP (mm) | -0.0473 | -0.001431 | 0.0478 |

| Orientation | Track Points | | |
|------------------|--------------|---------|---------|
| | P1 | P2 | P3 |
| Ztx | 0.0991 | -0.3179 | 0.5773 |
| Zty | 0.099 | -0.424 | -0.5773 |
| Ztz | -0.9901 | -0.848 | -0.5774 |
| ΔZ (deg) | 1.1140 | 0.9998 | 0.9997 |

Table 3. Position and orientation errors of the inverse kinematics for each of the three track points within three different quadrants

parameters) have little influence on increasing the model accuracy. This article will consider only static calibration with geometric parameter errors, since this type of parameter is the main source ($\approx 90\%$) of pose errors in industrial robots [28].

Dynamic robot calibration has importance only in large robots at high speeds and accelerations, requiring extensive and difficult experimental procedures [29, 30], which does not correspond to this case.

In general, a robot calibration system consists of an external measurement system, a calibration model including the parameters to be calibrated and the robot controller, as shown in Figure 8 [31].

In this work, the measurement system used was a Measurement Arm ITG ROMER, with an accuracy reported by the manufacturer of 0.087 mm. The measuring arm was used to measure the positions of the welding torch tip mounted on the robot flange, in several locations within its workspace. The system can be seen in Figure 9.

The procedures for robot calibration have four steps: (a) kinematic modelling, (b) position measurements, (c) parameter identification and (d) kinematic parameter updating or model compensation.

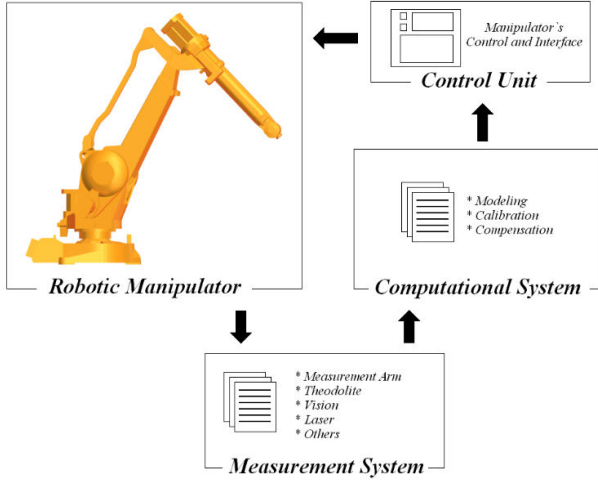


Figure 8. Block diagram of a robot calibration system



Figure 9. ITG ROMER measurement arm

5.1 Parameter identification modelling

Robot calibration consists basically in the problem of fitting a nonlinear model to experimental data. As a result, error parameters are identified by minimization of an error function.

A robot kinematic model is a complex non-linear function that relates link geometric parameters and joint variables to the robot end-effector pose, such as in:

$$P = T_1 \cdot T_2 \cdot \dots \cdot T_n, \quad (38)$$

where P is the manipulator transformation, T_i are the link transformations defined in Eq. (1) and n is the number of links. A kinematic model using the D-H convention can be expressed as (from Eq. (2)):

$$\Delta P = \frac{\partial P}{\partial \theta} \Delta \theta + \frac{\partial P}{\partial \alpha} \Delta \alpha + \frac{\partial P}{\partial d} \Delta d + \frac{\partial P}{\partial l} \Delta l, \quad (39)$$

where θ , α , d and l are parameters defining the transformation from a robot joint frame to the next joint frame, where d and l are translation parameters and θ and α are the rotation parameters.

The partial derivatives shown in Eq. (39) represent the contribution of the pose errors produced by each of the geometric error parameters of each joint, producing the total pose deviation of the robot's end-effector, which can be measured physically. Considering the measured robot pose (M) and the transformation from the robot base frame to the measurement system (B), then ΔP is the vector shown in Figure 10.

The transformation B can also be seen as a link that belongs to the robot model and has to be identified. Then the error value ΔP can be calculated using Eq. (40).

$$\Delta P = M - P - B = M - C \quad (40)$$

The transformation P can be updated iteratively each time a new set of geometric parameters are identified, and the calibration process ends up with P minimizing the deviations from the measured poses.

Rewriting Eq. (39) in a matrix form for various measured poses of the robot end-effector, Eq. (40) can be rewritten as the Jacobian matrix containing the partial derivatives of P , since Δx is the vector of the model parameter errors as:

$$\begin{bmatrix} \Delta P_1 \\ \Delta P_2 \\ \vdots \\ \Delta P_n \end{bmatrix} = \begin{bmatrix} \frac{\partial P_1}{\partial \theta} & \frac{\partial P_1}{\partial \alpha} & \frac{\partial P_1}{\partial d} & \frac{\partial P_1}{\partial l} \\ \frac{\partial P_2}{\partial \theta} & \frac{\partial P_2}{\partial \alpha} & \frac{\partial P_2}{\partial d} & \frac{\partial P_2}{\partial l} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial P_n}{\partial \theta} & \frac{\partial P_n}{\partial \alpha} & \frac{\partial P_n}{\partial d} & \frac{\partial P_n}{\partial l} \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta \alpha \\ \Delta d \\ \Delta l \end{bmatrix} = \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_n \end{bmatrix} \Delta x \Rightarrow J \Delta x = \Delta P \quad (41)$$

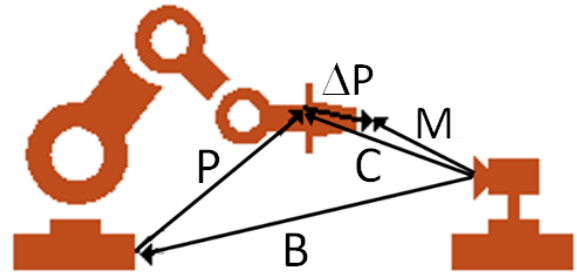


Figure 10. Calibration transformations

The size of the Jacobian matrix is a function of the number of measured points measured in the workspace (m) and the number of error parameters in the model (n). The matrix order is $(\eta m \times n)$, where η is the number of space degrees

of freedom (three positions and three orientation parameters). Thus, the calibration problem is reduced to the solution of a non-linear system of the type $\mathbf{J} \cdot \mathbf{x} = \mathbf{b}$.

There are many methods available to solve this type of system and one that is widely used is the Squared Sum Minimization (SSM). Numerous publications that discuss these methods and related algorithms extensively can be easily found [32]. A widely used method for solution of nonlinear least squares problems, which is successful in practice, is the algorithm proposed by Levenberg-Marquardt (L-M algorithm). Several versions of this algorithm have proved to be globally convergent. The algorithm turns out to be an iterative solution method by introducing few alterations in the Gauss-Newton method to overcome numerical divergence problems.

Each iteration of the algorithm follows three steps, where \mathbf{x}_k is the parameter vector of the kinematic model in the k^{th} iteration and $\Delta \mathbf{x}_k$ are the corrections to be inserted into the model [23], described by the following:

- Calculation of the robot's Jacobian ($\mathbf{J}(\mathbf{x}_k)$);
- Calculation of the vector $\Delta \mathbf{x}_k$ using the relation

$$\Delta \mathbf{x}_k = -\left[\mathbf{J}(\mathbf{x}_k)\right]^T \mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I} \Big)^{-1} \left[\mathbf{J}(\mathbf{x}_k)\right]^T \Delta \mathbf{P}(\mathbf{x}_k);$$

- Update $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x}_k$ and $k = k + 1$.

where μ_k is obtained from the relations as expressed in Eq. (42).

$$\begin{cases} \mu_0 = 0.001 \\ \mu_{k+1} = \begin{cases} 0.001\lambda & \text{if } \|\Delta \mathbf{P}(\mathbf{x}_{k+1})\| \geq \|\Delta \mathbf{P}(\mathbf{x}_k)\| \\ 0.001/\lambda & \text{if } \|\Delta \mathbf{P}(\mathbf{x}_{k+1})\| < \|\Delta \mathbf{P}(\mathbf{x}_k)\| \end{cases} \\ 2.5 < \lambda < 10 \end{cases} \quad (42)$$

5.2 Kinematic modelling for robot calibration

In parameterized kinematic models for robot calibration, three properties are the most important: completeness, continuity and minimality [24]. Completeness is regarded as the ability of the model to describe every possible geometrical variation in the links and joints of a robot. Continuity and minimality characterize proportion and redundancies in parameters of the calibration model, respectively.

In our case, the robot has only perpendicular axes. The D-H convention, shown in Eq. (2), can be safely used in parameterized error models when modelling perpendicular axes, which is not true for parallel axes due to singularities that occur in the Jacobian matrix. This topic is discussed in detail in [23].

The transformations shown in Eq. (40) have to include parameterized errors in the kinematic model for the robot calibration. The geometric parameters included in the identification model of this robot can be seen in Table 4, where δs are the error parameters that model the geometric differences between the robot nominal model and the corrected model, which is closer to the actual robot. The parameters can be identified by the calibration system algorithm and are all initialized to zero. The transformations between the world coordinate frame (at the frame origin of the measurement system) and the robot base coordinate frame (W-B), joint to joint transformations (J_R or $P - J_R$ or P) (R: rotative; P: prismatic) and last joint to the TCP coordinate frame ($J_R - J_{TCP}$) include all the geometric parameters of rotation and translation. This first model is complete, but there is no guarantee that it is minimal.

The first (W-B) and the last ($J_R - J_{TCP}$) transformations shown in Table 4 are the transformations that locate the coordinate system of the robot base with respect to the world coordinate system, and the TCP with respect to robot flange, respectively. As both systems can vary in position and orientation that cannot be measured, it is necessary that their elementary transformations belong to the Euclidean group, with six parameters. However, if only position data are measured in the TCP with the external measurement system, there is no need to include orientation error parameters in the last transformation.

| Joint Type | D-H Parameterization |
|-----------------|--|
| W - B | $[T_x(\text{pxb}), T_y(\text{pyb}), T_z(\text{pzb}), R_x(\delta \text{xb}), R_y(\delta \text{yb}), R_z(\delta \text{zb})]$ |
| $J_R \perp J_R$ | $[R_z(\theta_1 + \delta \theta_1), T_z(\text{pz}_1 + \delta \text{pz}_1), T_x(\text{px}_1 + \delta \text{px}_1), R_x(\alpha_1 + \delta \alpha_1)]$ |
| $J_R \perp J_P$ | $[R_z(\theta_2 + \delta \theta_2), T_z(\text{pz}_2 + \delta \text{pz}_2), T_x(\text{px}_2 + \delta \text{px}_2), R_x(\alpha_2 + \delta \alpha_2)]$ |
| $J_P \perp J_R$ | $[R_z(\theta_3 + \delta \theta_3), T_z(\text{pz}_3 + \delta \text{pz}_3), T_x(\text{px}_3 + \delta \text{px}_3), R_x(\alpha_3 + \delta \alpha_3)]$ |
| $J_R \perp J_R$ | $[R_z(\theta_4 + \delta \theta_4), T_z(\text{pz}_4 + \delta \text{pz}_4), T_x(\text{px}_4 + \delta \text{px}_4), R_x(\alpha_4 + \delta \alpha_4)]$ |
| $J_R - J_{TCP}$ | $[R_z(\theta_5 + \delta \theta_5), T_x(\text{px}_5 + \delta \text{px}_5), T_y(\text{py}_5 + \delta \text{py}_5), T_z(\text{pz}_5 + \delta \text{pz}_5)]$ |

Table 4. Initial robot kinematic model parameters and transformation equations. (\perp = perpendicular, R = rotary, P = prismatic, W = world, B = base)

A proper choice of the error parameters to be included in the identification model is very important to ensure the minimality and continuity of the model, such that the error parameters do not impose parametric redundancies that produce ill-conditioned solutions into the nonlinear identification routines. Therefore, a finite number of parameters may have to be excluded from the complete model shown in Table 4. The mathematical basis and the strategies to optimize the model parameterization and ensure a good model fitting are discussed in the next section.

6. Kinematic Model Optimization

A very useful tool to analyse, evaluate and optimize kinematic models and parameter identification routines is

the Singular Value Decomposition (SVD) [32] of the J matrix, which is an algorithm equivalent to a linearization in the least-square sense. The SVD is a powerful tool to deal with equations or matrices that are either singular or numerically quasi-singular.

The Jacobian matrix in Eq. (41) can be written as ($h = 3.m$ if only positions are measured):

$$J_{h \times n} = U_{h \times h} \cdot S_{n \times n} \cdot V_{n \times n}^T \quad (43)$$

where the number of rows, h , is larger than or equal to the number of columns, n ; S is a square diagonal matrix with positive or null entries; U is an orthogonal matrix and V^T is the transpose of an orthogonal matrix. This matrix decomposition is always possible, no matter how ill-conditioned the J matrix is.

If J has rank r , the singular values of S can be ordered to be non-increasing such that all values are non-negative and exactly r of them are positive.

6.1 Condition number

In numerical analysis, the condition number can be viewed as an observability index [33] of model parameters to be identified. It can also be considered as an amplification factor in numerical perturbation and error analysis [32].

The condition number of the matrix J can be calculated from:

$$k(J) = \|J^T \cdot J\| \cdot \|(J^T \cdot J)^{-1}\|, \quad (44)$$

where $\|\cdot\|$ is a given norm.

If the previous norm is Euclidean, the condition number can be directly calculated from the largest, S_l , and the smallest, S_r , non-zero singular values as:

$$k(J) = \frac{S_l}{S_r} \quad (45)$$

6.2 Jacobian column scaling

A procedure to improve the Jacobian matrix condition can be implemented by performing column scaling. Scaling factors can be valuated from the expected error of the robot ($\cong 1$ mm) [24, 34]. These factors can be calculated using the model function $P(x)$ (Eq. 38), by using the first-order approximation in x_k :

$$\frac{\partial P(q, x)}{\partial x_k} = \frac{P(q, x + \Delta x_k) - P(q, x)}{\Delta x_k}, \quad (46)$$

where $x = [x_1^T, x_2^T, \dots, x_n^T]^T$, n is the number of parameters and q is the robot pose.

For the robot pose q , the value

$$\sigma_{x_k}(q) = \frac{1^{-3}}{\left\| \frac{\partial P(q, x)}{\partial x_k} \right\|} \quad (47)$$

is the parameter variation that, using a first-order approximation, produces a TCP's position displacement of 1 mm. The denominator is the norm of the k^{th} column of the Jacobian.

If the $\sigma_{x_k}(q)$ values are calculated for a large set of joint positions, $q = [q_1, \dots, q_m]$, without restriction of position and orientation, the values

$$\sigma x_k = \min \{ \sigma x_k(q) \} \cong \min \{ \sigma x_k(q) \} \quad (48)$$

$$q \in \{q_1, \dots, q_m\} \quad q \in \mathbb{R}^N$$

are very close to the minimum deviation among all possible positions [24]. The σx_k values are denominated as *extreme values* and are to be employed for column scaling of the linearized least-square system in Eq. (41).

6.3 Optimization scheme

The main purpose of calibration model optimization is to reduce the number of parameters in the model in order to eliminate dependencies or quasi-dependencies between them, not to the point of restricting the model accuracy but far enough so that the condition number of the scaled Jacobian is less than 100. The experience of research groups in mathematics demonstrates that a condition number smaller than 100 is required for reliable results [24, 32].

Therefore, the procedures for model optimization can follow the following steps:

- Model-based scaling (Eq. 48) using extreme scaling values must be computed to reduce the condition number $k(J)$ many times.
- Quasi-dependencies and non-identifiabilities are pinpointed by investigating the column vector V_r corresponding to the smallest singular value S_r of the SVD as follows:

$$J \cdot V = U \cdot S \quad (49)$$

and so

$$\|J \cdot V_r\| = S_r \quad (50)$$

If J has rank r , then $S_{r+1} = \dots = S_n = 0$. The optimization procedure is then performed by following the two previous

steps. The first step improves the condition $k(J)$. If $k(J)$ is higher than 100 then the next step determines the model parameters that produce rank deficiencies. Thus, an optimal model is obtained by excluding a small number of parameters from the complete model successively, until $K(J)$ is less than 100 during the parameter identification routine. However, in practice, most of the available pose measurement systems used in robot metrology (ultrasound, contact, CCD cameras, laser systems, theodolites, etc.) have an accuracy ranging from 0.05 mm to 0.5 mm, depending on its type and complexity. If measurement noise is increased by using measuring systems with lower accuracies, then usually $k(J)$ is also increased, and the threshold of $k(J) = 100$ will not be achieved by excluding only a few parameters from the model. If that is the case, the routine of the kinematic model optimization stops if $k(J)$ has only a small reduction on the exclusion of one more parameter. It is important to note that the optimization procedure described aims to exclude the most redundant parameter from the parameterized model at a time. In this way, regardless of the number of parameters in the model, there will always be an improvement in the condition number with the exclusion of another parameter. However, if a parameter is removed from the parameterized model that has not been identified by the optimization routine then there will be no guarantee that the condition number will improve, or that there will be an improvement in the kinematic model accuracy.

7. Experimental Results

The model optimization method aiming at eliminating redundant parameters from the calibration model was tested in an experimental procedure, where the actual robot had been measured at 24 positions that were selected to cover a large range of the robot's joint positions. The results of the experiment are shown in Table 5.

The calibration process was initially carried out with the maximum number of parameters shown in Table 5. The algorithm stopped when the Euclidean norm of the parameter error vector did not change significantly. Equations (49) and (50) were then used to calculate the condition number, $K(J)$, and to identify the parameters corresponding to the smallest singular value in a model, easily observed from the largest entry in the last column of the scaled V matrix. Frequently, two parameters have approximately the same largest value in the interval $0 < v_i < 1$, where v_i is the entry in the i^{th} line of the last column of V , which means that they are mutually redundant. The parameter to be eliminated from the model is the one with a null value in the nominal model (not necessary for the nominal model).

The results showed in Table 5 show clearly that as long as a parameter is removed from the model, the condition number of the Jacobian reduces as well. The model with 18 parameters presented a condition number below 100 in the

| Number of parameters to be identified | | | | | | |
|---------------------------------------|-------------------|-------------------|------------------|-----------------|------------------|------------------|
| | 26 | 25 | 24 | 23 | 22 | 21 |
| E (mm) | 0.310 | 0.342 | 0.346 | 0.338 | 0.336 | 0.343 |
| Ave (mm) | 0.585 | 0.576 | 0.913 | 0.801 | 0.832 | 0.843 |
| Std (mm) | 0.240 | 0.229 | 0.307 | 0.302 | 0.298 | 0.308 |
| K | 530×10^6 | 254×10^8 | 14×10^7 | 1×10^6 | 60×10^3 | 16×10^3 |
| Iter. | 13 | 15 | 12 | 17 | 16 | 14 |
| Par. | pz1 | t1 | pz4 | px3 | t5 | px4 |
| Number of parameters to be identified | | | | | | |
| | 20 | 19 | 18 | 17 | 11 | |
| E (mm) | 0.373 | 0.387 | 0.328 | 0.124 | 0.084 | |
| Ave (mm) | 0.741 | 0.719 | 0.704 | 0.799 | 0.888 | |
| Std (mm) | 0.334 | 0.315 | 0.274 | 0.327 | 0.421 | |
| K | 7×10^3 | 198 | 27 | 19 | 14 | |
| Iterat. | 6 | 6 | 5 | 5 | 4 | |
| Par. | a4 | a2 | t2 | - | - | |

E : Euclidean norm of the parameter error vector after calibration;

Ave : Average of the position errors after calibration;

Std : Standard deviation of the position errors after calibration;

K : Condition number calculated by Eq. (28) in the last iteration;

Iter.: Number of iterations;

Par.: Parameter identified as redundant after calibration.

Table 5. Statistics of the calibration process as a function of the number of parameters to be identified in the model

sequence. That coincides with the smallest average position error.

However, the precision of the calibrated model has to be checked moving the robot to other points than the ones at which the measured points were collected. Therefore, the accuracy evaluation step is to correct the nominal model using the parameter errors obtained from the calibration routine and to proceed the measurement process at other locations, but calibrating only the first six parameters of the model, i.e., only the robot base parameters.

The robot was evaluated at another quadrant of the workspace, far from the region at which it had been previously calibrated. The results shown in Table 6 demonstrate that the highest accuracy had been obtained using the model with 18 parameters. One can realize that the robot's position accuracy had improved each time that a parameter had been removed, and started decreasing when the number of parameters was below 18. This behaviour clearly agrees with the mathematical concepts of completeness and minimality, which means that too many parameters produce redundancies in the kinematic model and using fewer parameters than is necessary cannot model the robot's geometry completely.

| CALIBRATION EVALUATION RESULTS | | | | | |
|--------------------------------|-----------|---------------|------------------|-----|---------|
| No. Param. | E (mm) | Ave (mm) | Std Dev. (mm) | K | Iterat. |
| 26 | 0.428 | 110.45 | 42.16 | 7.7 | 5 |
| 25 | 0.186 | 0.923 | 0.413 | 7.7 | 2 |
| 24 | 0.214 | 1.166 | 0.349 | 7.7 | 2 |
| 23 | 0.212 | 1.104 | 0.344 | 7.7 | 2 |
| 22 | 0.170 | 1.079 | 0.355 | 7.7 | 2 |
| 21 | 0.230 | 1.140 | 0.308 | 7.7 | 2 |
| 20 | 0.164 | 0.847 | 0.404 | 7.7 | 2 |
| 19 | 0.167 | 0.872 | 0.389 | 7.7 | 2 |
| 18 | 0.121 | 0.750 | 0.406 | 7.7 | 2 |
| 17 | 0.220 | 1.073 | 0.589 | 7.7 | 2 |
| 11 | 0.156 | 1.200 | 0.656 | 7.7 | 2 |

Abbreviations have the same meanings as in Table 5

Table 6. Statistics of the calibration evaluation procedure, with robot joint positions other than the ones used to calibrate the model, within another quadrant of the workspace.

The parameter optimization scheme had been proved to be successful, pinpointing the optimal error parameter vector to be identified in order to improve the robot's accuracy.

8. Conclusions

This article presented a robot constructed to repair hydraulic turbine defects by welding in layers automatically by using a 3D surface map acquired from a specialized vision sensor. The discussion focused on the kinematic model, the algorithm to perform the iterative inverse kinematics and the mathematical procedure for parameter-identification optimization of the robot's calibration system.

Experimental results showed that the iterative algorithm for inverse kinematics is very efficient, converging to the solution in three to four iterations and showing very good tracking. Experimental work with welding in straight lines also proved the reliability of the solution. The mathematical scheme to find the optimal set of geometric parameters for robot calibration proved to be efficient and simple, able to find the best set of geometrical parameters to be identified by the robot's calibration procedures, leading to the best accuracy for the robot model among all possible parameter sets.

9. Acknowledgements

This work had been supported by ELETRONORTE (Electrical Power Plants of the North of Brazil) and FINATEC (Foundation for Scientific and Technological Enterprises).

10. References

[1] Parmar R S (1997) Welding processes and technology. Delhi: Khanna Publishers. xx+808.

[2] Ecober X, Egusquiza E, Farhat M, Avellan F, Coussirat M (2006) Detection of cavitation in hydraulic turbines. *Mech. Syst. and Signal Proc.* 20: 983-1007.

[3] Hammit F G (1979) Cavitation erosion: the state-of-the-art and predicting capability. *Applied Mechan. Reviews.* 32: 665-675.

[4] Hazel B, Coté J, Laroche Y, Mongenot P (2012) A portable, multiprocess, track-based robot for in situ work on hydropower equipment. *J. Field Robotics.* 29: 69-101.

[5] Chen Q, Sim Z, Zhang W, Gui Z (2008) A robot for welding repair of hydraulic turbine blade. In: *Conference on Robotics, Automation and Mechatronics*, Chengdu, China.

[6] Xiang K, Sun Z, Dai H, et al. (2010) CAN-Bus based distributed control system for hydraulic turbine blade repairing robot. In: Liu H, Ding H, Xiong Z, et al., editors. *3rd International Conference on Intelligent Robotics and Applications (ICIRA)*; 2010 Nov 10-12; Shanghai, China. *Lecture Notes in Artificial Intelligence.* 6425: pp. 695-704.

[7] Motta J M S T, Llanos C H, Carvalho G C, Absi-Alfaro S C (2010) A prototype of a specialized robotic system for repairing hydraulic turbine blades. In: *1st International Conference on Applied Robotics for the Power Industry (CARPI)*, Montreal, Canada.

[8] Motta J M S T, Llanos C H, Carvalho G C, Absi-Alfaro S, Idrobo-Pizo G, Sampaio R (2014) A specialized robotic system prototype for repairing surface profiles of hydraulic turbine blades. In: *3rd International Conference on Applied Robotics for the Power Industry (CARPI)*, Foz do Iguassu, Brazil.

[9] Motta J M S T, Carvalho G C, Llanos-Quintero C H, Vidal-Filho W B, Vasconcelos-Filho, E P (2010) A specialized welding robot for repairing hydraulic turbine blades. In: *Symposium Series in Mechatronics*, Vol. 4. Victor J De Negri Editor. Brazil: ABCM. pp. 700-709.

[10] Gan J Q, Oyama E, Rosales E M, Hu H (2005) A complete analytical solution to the inverse kinematics of the Pioneer 2 robotic arm. *Robotica.* 23: 123-129.

[11] Xu D, Calderon C A A, Gan J Q, Hu H (2005) An analysis of the inverse kinematics for a 5-DOF manipulator. *Intern. J. Autom. and Comp.* 2: 114-124.

[12] Park J, Kim J-M, Park H-H, Kim J-W, Kang G-H, Kim S-H (2012) An iterative algorithm for inverse kinematics of 5-DOF manipulator with offset wrist. *World Academy Of Science, Eng. and Tech.* 6: 144-149.

[13] Ayob M A, Zakaria W N W, Jalani J, Tomari M R M (2015) Inverse kinematics analysis of a 5-axis

- RV-2AJ robot manipulator. *ARPN J. Eng. App. Sciences*. 10: 8388-8394.
- [14] Olunloyo V, Ibidapo-Obe1 O, Olowookere D, Ayomoh M (2014) Inverse kinematics analysis of a five jointed revolute arm mechanism. *J. Control Science Eng.* 2: 7-15.
- [15] Liu Z-Z, Liu H-Y, Luo Z (2012) inverse kinematics analysis of 5-DOF robot manipulators based on virtual joint method. *Applied Mech. Mater.* 143: 265-268.
- [16] Roth Z S, Mooring B, Ravani B. (1987) An overview of robot calibration. *IEEE J Robot Automat.* 3: 377-85.
- [17] Zhuang H Q, Roth Z S, Hamano F (1992) A complete and parametrically continuous kinematic model for robot manipulators. *IEEE Trans Robot Automat.* 8: 451-63.
- [18] Nubiola A, Bonev I A (2013) Absolute calibration of an ABB IRB 1600 robot using a laser tracker. *Robot Comput-Integr Manuf.* 29: 236-45.
- [19] Bai Y, Wang D (2004) Improve the robot calibration accuracy using a dynamic online fuzzy error mapping system. *IEEE Trans. Syst. Man Cybern. Part B.* 34: 1155-1160.
- [20] Tian W, Mei D, Li P, Zeng Y, Hong P, Zhou W (2015) Determination of optimal samples for robot calibration based on error similarity. *Chin. J. Aeronaut.* 28: 946-953.
- [21] Zhou W, Liao W H, Tian W (2013) Theory and experiment of industrial robot accuracy compensation method based on spatial interpolation. *Chin J Mech Eng.* 49: 42-8.
- [22] Wang H, Lu X, Hu Z, Li Y (2015) A vision-based fully-automatic calibration method for hand-eye serial robot. *Indust. Robot: An Inter. J.* 42: 64-43.
- [23] Motta J M S T (2006) Robot calibration: modeling measurement and applications. In *Industrial Robotics: Programming, Simulation and Applications*. Low Kin Huat Editor. Croatia: InTech. pp. 107-130.
- [24] Schröer K (1993) Theory of kinematic modelling and numerical procedures for robot calibration. In: *Robot Calibration*. Bernhardt and Albright. UK: Chapman & Hall. pp. 157-193.
- [25] Jang J H, Kim S H, Kwak Y K (2001) Calibration of geometric and non-geometric errors of an industrial robot. *Robotica.* 19: 311-321.
- [26] Majarena A C, Santolaria J, Samper D, et al. (2010) An overview of kinematic calibration models using internal/external sensors or constraints to improve the behaviour of spatial parallel mechanisms. *Sensors.* 10: 10256-10297.
- [27] Santolaria J, Aguilar J-J, Yague J-A, Pastor J (2008) Kinematic parameter estimation technique for calibration and repeatability improvement of articulated arm coordinate measuring machines. *Prec. Engin.* 32: 251-268.
- [28] Stark G, Benz E, Hüttenhofer M (1993) Calibration experiences in industry. In: *Robot Calibration*. Bernhardt and Albright. UK: Chapman & Hall. pp. 283-295.
- [29] Wu J, Wang J, You Z (2010) An overview of dynamic parameter identification of robots. *Robot. and Comp.-Integ. Manufacturing.* 26: 414-419.
- [30] Surdilovic D, Lizama E, Lück R (1993) Identification of dynamic parameters of large manipulator arms with closed kinematic loops. In: *Robot Calibration*. Bernhardt and Albright. UK: Chapman & Hall. pp. 233-269.
- [31] Ginani L S, Motta J M S T (2011) Theoretical and practical aspects of robot calibration with experimental verification. *J Braz Soc Mech Sci & Eng.* 33: 15-21.
- [32] Press W H, Teukolsky S A, Vetterling S A, Flannery B P (2007) *Numerical recipes 3rd edition: the art of scientific computing*. UK: Cambridge University Press. 1256.
- [33] Driels M R, Pathre U S (1990) Significance of observation strategy on the design of robot calibration experiments. *J. Rob. Systems.* 7: 197-223.
- [34] Hollerbach J, Khalil W, Gautier M (2008) Model identification. In: *Siciliano B, Khatib O Editors. Springer Handbook of Robotics*. Berlin: Springer - Verlag. pp. 321-344.