ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE
PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE
DE LA PRODUCTION AUTOMATISÉ
M.Ing.

PAR
Albert NUBIOLA

CALIBRATION OF A SERIAL ROBOT USING A LASER TRACKER

MONTRÉAL, LE 15 JUILLET 2011

©Tous droits réservés, Albert NUBIOLA, 2011

©Tous droits réservés

Cette licence signifie qu'il est interdit de reproduire, d'enregistrer ou de diffuser en tout ou en partie, le présent document. Le lecteur qui désire imprimer ou conserver sur un autre media une partie importante de ce document, doit obligatoirement en demander l'autorisation à l'auteur.

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE

M. Ilian Bonev, directeur de mémoire
Département de génie de la production automatisée à l'École de technologie supérieure

M. Vincent Duchaine, président du jury
Département de génie de la production automatisée à l'École de technologie supérieure

Pascal Bigras, membre du jury
Département de génie de la production automatisée à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 15 JUILLET 2011

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

I would like to thank my supervisor Prof. Ilian Bonev for his constant support and motivation to keep me working on my studies. He gave me all the necessary means and equipment to complete my project.

I would like to thank Dr. Torgny Brogardh, from ABB headquarters, for his comments and suggestions on our results. I would also like to thank Dr. Mohamed Slamani for his help.

I thank my family for their support. My mother, Judith, and my father, Jordi, always took care of me and gave me the right advice at the right time. Thanks to my parents I ended up studying in the field of robotics.

Last but not least I thank Lauren, for her loving support and patience helping me to improve my English. Thanks to Lauren it was much easier for me to write this thesis.

ÉTALONNAGE D'UN ROBOT SÉRIEL AVEC UN SYSTÈME DE LASER DE POURSUITE

Albert NUBIOLA

RÉSUMÉ

La performance en positionnement d'un robot industriel ABB IRB 1600-6/1.45 a été étudiée avec un système de laser de poursuite (« laser tracker »). En faisant l'analyse axe par axe, on trouve que les axes 2, 3 et 6 ont un comportement non géométrique. Un modèle à 34 paramètres d'erreur a été utilisé pour modéliser le robot réel. Ce modèle d'erreur tient en compte les défauts géométriques de fabrication ainsi que quatre paramètres d'erreur concernant la rigidité (provenant des axes 2 et 3) et quatre autres paramètres d'erreur pour modéliser le comportement non linéaire du sixième axe avec une série de Fourier de deuxième ordre. L'algorithme d'optimisation non linéaire Nelder-Mead a été utilisé pour trouver les paramètres d'erreur qui correspondent aux mesures prises du robot.

Une fois les 34 paramètres identifiés, on ne peut pas appliquer une solution algébrique pour calculer la cinématique inverse du modèle à 34 paramètres d'erreur. On propose une solution numérique itérative à la cinématique inverse. Au maximum trois itérations sont nécessaires pour obtenir les angles des moteurs correspondants à une pose de l'outil.

Pour comparer la précision entre le modèle nominal et le modèle corrigé (à 34 paramètres d'erreur) on a fait des tests similaires à ceux proposés dans la norme ISO 9283. La validation de l'amélioration de la précision absolue est faite avec de nombreuses mesures. Pour le modèle à 34 paramètres, l'erreur de position moyenne/maximale est réduite de 0.979 mm / 2.326 mm à seulement 0.329 mm / 0.916 mm (vérification faite avec environ 1000 mesures arbitraires), avec une charge de l'outil de 6 kg (soit, le maximum), pour huit points sur l'outil et pour tout l'espace de travail du robot (ou presque, car il y avait quelques obstacles proches du robot à éviter).

On a fait des analyses avec l'erreur attendue, qui permettent de « prévalider » les modèles sans devoir prendre plus de mesures. On trouve que cette « prévalidation » est proche à la validation réelle.

Mots-clés : étalonnage robotique, laser tracker, erreur de position, erreur d'orientation, cinématique inverse, paramètres d'erreur, optimisation, IRB1600-6/1.45, ISO 9283.

CALIBRATION OF A SERIAL ROBOT USING A LASER TRACKER

Albert NUBIOLA

ABSTRACT

The positioning performance of an industrial robot ABB IRB 1600-6/1.45 has been studied with a laser tracker. Performing some axis-by-axis analyses, we found that axes 2, 3 and 6 have a non-geometrical behavior. A 34-parameter model was used to represent the real robot. This error model takes into account the geometrical errors due to fabrication as well as four error parameters related to stiffness (in axes 2 and 3) and four other error parameters used to fit a second-order Fourier series to the non-linear behavior of axis 6. The Nelder-Mead non linear optimization technique was used to find the error parameters that best fit the measures acquired with the laser tracker.

An algebraic solution for the inverse kinematics is not possible for the 34-parameter model. We therefore propose a numerical and iterative inverse algorithm to recalculate the robot targets into so-called fake targets. No more than three iterations are needed to accurately obtain the joint angles corresponding to a given pose of the end-effector.

Similar tests to the ones proposed by the ISO 9283 norm are performed to compare the accuracy of the nominal and improved robot models. The validation of the accuracy is done with a large number of measures. For the 34-parameter model the mean / maximum position errors are reduced from 0.979 mm / 2.326 mm to 0.329 mm / 0.916 mm (verification performed with around 1000 measurements), at a 6 kg payload, for eight points on the end-effector and for the complete robot workspace (or almost complete, since we had to avoid some obstacles).

Analyses were performed with the expected errors. They allow to “pre-validate” the models without having to take extra measurements. It was found that this pre-validation is very close to the real validation.

Keywords: robot calibration, laser tracker, position error, orientation error, inverse kinematics, error parameters, optimization, IRB1600-6/1.45, ISO 9283.

TABLE OF CONTENTS

Page

INTRODUCTION 1
1. Absolute vs. relative calibration 3
2. Open-loop vs. closed-loop calibration 4
3. Thesis organisation 4

CHAPTER 1 LITERATURE REVIEW 7
1.1 Robot calibration process..7
1.2 Tool calibration..7
1.3 Robot calibration...8
    1.3.1 Level-1 models.. 8
    1.3.2 Level-2 models.. 9
    1.3.3 Level-3 models.. 9
1.4 Kinematic modeling..10
1.5 Inverse kinematics computation 10
    1.5.1 Algebraic.. 11
    1.5.2 Iterative 11
1.6 Optimization algorithms 12
    1.6.1 Line-search methods 13
    1.6.2 Trust-region method... 13
    1.6.3 Nelder-Mead 13
1.7 Commercial solutions for robot calibration 14
    1.7.1 Dynalog.. 14
    1.7.2 Nikon Metrology.. 15
    1.7.3 Teconsult.. 16
    1.7.4 Wiest AG 16
    1.7.5 American Robot Corporation.. 17
1.8 Recent calibration results reported in the literature 18

CHAPTER 2 OBJECTIVES AND METHODOLOGY 21
2.1 Thesis objectives...21
2.2 Methodology...22

CHAPTER 3 KINEMATIC CALIBRATION 25
3.1 Kinematic parameter extraction from identified joint axes 25
    3.1.1 Method to find the axis of each joint 26
    3.1.2 Placing the link frames... 27
    3.1.3 Dealing with parallel axes.. 29
    3.1.4 Error parameters needed for a full calibration 29
    3.1.5 Homogeneous transformation.. 30
3.2 Base frame definition..31
3.3 Finding the base from three points...33

3.4 End-effector calibration 34
  3.4.1 End-effector design 35
  3.4.2 End-effector calibration for unknown dimensions 36
  3.4.3 End-effector calibration for known dimensions 38
  3.4.4 Experimental values 39
3.5 Kinematic parameter optimization 40

CHAPTER 4 ANALYSES OF THE BEHAVIOR OF EACH JOINT 43
4.1 Axis error analysis 44
4.2 Detailed analysis of joint 6 46
4.3 Detailed analysis of the 2nd axis 48
  4.3.1 Effect of the tool weight 48
  4.3.2 Robot arm stiffness effect 51

CHAPTER 5 ERROR MODELS 53
5.1 Nominal direct kinematics 53
5.2 Axis 6 model 55
5.3 Stiffness model 57
5.4 Direct kinematic models 61
  5.4.1 Nominal model 61
  5.4.2 11-parameter model 61
  5.4.3 16-parameter model 62
  5.4.4 Kinematic calibration 62
  5.4.5 34-parameter model 63

CHAPTER 6 INVERSE KINEMATICS 65
6.1 Nominal inverse kinematics 65
  6.1.1 Finding q_1, q_2 and q_3 65
  6.1.2 Case with C=0 67
  6.1.3 Case with $C\neq 0$ 67
  6.1.4 Finding q_4, q_5 and q_6 69
  6.1.5 Wrist singularity (q_5=0) 70
6.2 Iterative inverse calculation 71

CHAPTER 7 CALIBRATION METHODS 75
7.1 Procedure for taking measurements 75
7.2 Points needed for calibration 77
7.3 Calibration procedure 79
7.4 Verification procedure 80
7.5 Error vs. iterations 81
7.6 Expected error vs. real position error 83
7.7 Verification results 84
  7.7.1 Results from nominal kinematics model 85
  7.7.2 Results from entire kinematic calibration model 86
  7.7.3 Optimisation test results from 80 to 1000 identification measurements... 87

7.7.4 Results found for 34-parameter calibrations from 80 to 200 measurements.......... 99
7.7.5 Worst random 34-parameter procedures from 80, 120, 180 and 200 measurements.......... 102

CONCLUSION 109

ANNEX I POSITION REPEATABILITY OF THE ABB IRB 1600-6/1.45 113

ANNEX II FARO ION LASER TRACKER ACCURACY 117

ANNEX III FARO ION LASER TRACKER REPEATABILITY 121

ANNEX IV FARO ION LASER TRACKER 24-HOUR TEST 123

ANNEX V ABB IRB 1600-6/1.45 POSITION STABILIZATION TIME 127

ANNEX VI CIRCLE PATH ACCURACY TESTS WITH A TELESCOPIC BALLBAR 129

REFERENCES 137

LIST OF TABLES

Page

Table 3.1 *xyz* coordinates of the 9 targets (CAD + real points)................................40

Table 4.1 Range of motion for each joint 45

Table 4.2 Tests for 6th axis analysis..46

Table 4.3 Maximum TCP errors 51

Table 4.4 Tests for stiffness analysis 51

Table 5.1 Nominal DHM parameters for the ABB IRB 1600 robot..........................54

Table 5.2 Coefficients of the Fourier fit 56

Table 5.3 Nominal robot model...61

Table 5.4 Robot model for 11 parameters..62

Table 5.5 Robot model for 16 parameters..62

Table 5.6 Full robot kinematic model..63

Table 5.7 Robot model for 34 parameters..63

Table 6.1 Stabilization of the iterative inverse kinematics 74

Table 7.1 Expected position error for the nominal kinematic model for 8 targets. ...86

Table 7.2 Expected position error for the kinematic calibration model (3 kg)..........86

Table 7.3 Expected position error for the kinematic model (6 kg). 87

Table 7.4 Position errors for a 34-parameter calibration from 200 identification measurements (3 kg)..89

Table 7.5 Position errors for a 34-parameter calibration from 200 identification measurements (6 kg)..89

Table 7.6 Position errors for a 11-parameter calibration from 400 identification measurements (3 kg)..90

Table 7.7 Position errors for a 11-parameter calibration from 400 identification measurements (6 kg)..90

Table 7.8 ISO 9283 position and orientation errors for the four models (3 kg) 93

Table 7.9 Position and orientation errors found for the four models (3 kg) 95

Table 7.10 Relative position errors (from virtual frame, 3 kg).................................. 98

Table 7.11 Relative position errors (from 3 targets, 3 kg).. 99

Table 7.12 Position and orientation errors for the 34-parameter procedures (3 kg). 100

Table 7.13 Worst expected errors from pose measurements for the 34-parameter procedures (3 kg) 103

Table 7.14 Position statistics for the five 34-parameter procedures chosen (6 kg) .. 104

Table 7.15 Position statistics reduced to the 6th axis for the five 34-parameter procedures (3 kg) 105

Table 7.16 Orientation statistics for the five 34-parameter procedures (3 kg) 105

LIST OF FIGURES

Page

Figure 3.1 Axis-by-axis rotation for identification 26

Figure 3.2 Placing frame *i* 28

Figure 3.3 Position of the nominal and real robot base frame 32

Figure 3.4 Three points that determine the base frame 33

Figure 3.5 End-effector used for holding the nine SMRs 35

Figure 3.6 Identifying 5th and 6th axes....................36

Figure 3.7 Placing the 6th frame 37

Figure 3.8 Geometry to find the 6th frame....................38

Figure 4.1 The IRB 1600 robot of which the positioning performance was analyzed using a Faro laser tracker 43

Figure 4.2 Axis error analysis 44

Figure 4.3 Position and angular errors at the TCP when moving each joint one by one....................45

Figure 4.4 Position representation for each of the five tests. 47

Figure 4.5 Joint 6 angle errors....................47

Figure 4.6 Moving axis 2 to evaluate stiffness....................48

Figure 4.7 Picture of the real measurement points....................49

Figure 4.8 Stiffness effect with fully extended arm and a payload of 1.8 kg....................50

Figure 4.9 Stiffness effect with fully extended arm and a payload of 4.8 kg....................50

Figure 4.10 Equivalent θ_2 error 52

Figure 5.1 Frames corresponding to the DHM notation for the ABB IRB 1600 robot 55

Figure 5.2 Fourier series fit for the nonlinear motion pattern of joint 6 56

Figure 5.3 Robot arm representation 57

Figure 5.4 Estimating ΔL 58

Figure 5.5 Equivalent θ_2 error 60

Figure 5.6 Equivalent θ_2 error for a different combination of parameters 60

Figure 6.1 Solutions for q_3 68

Figure 6.2 Block diagram of the algorithm 72

Figure 6.3 Iterative inverse kinematics representation for iteration i 72

Figure 7.1 Random ISO cube points (left) and all range points (right) with respect to the robot and laser tracker. 79

Figure 7.2 Evolution of calibration step 82

Figure 7.3 Analysis of the expected error 83

Figure 7.4 Position error in ISO cube area (3 kg) 91

Figure 7.5 Position error in all robot range area (3 kg) 91

Figure 7.6 Position error in ISO cube area (6 kg) 92

Figure 7.7 Position error in all robot range area (6 kg) 92

Figure 7.8 ISO 9283 position errors reduced to the 6th axis (3 kg) 94

Figure 7.9 ISO 9283 orientation errors (3 kg) 94

Figure 7.10 Position errors reduced to the 6th axis for the four models (3 kg) 96

Figure 7.11 Orientation errors for the four models (3 kg) 96

Figure 7.12 Position of the 434 sphere points chosen for relative analysis 97

Figure 7.13 Relative position errors from a virtual frame (3 kg) 98

Figure 7.14 Relative position errors from 3 targets (3 kg) 99

Figure 7.15 Position errors reduced to the 6th axis for the four 34-parameter procedures (3 kg) 101

Figure 7.16 Orientation errors for the four 34-parameter procedures (3 kg) 101

Figure 7.17 Position errors for the five 34-parameter procedures chosen (6 kg)........104

Figure 7.18 Position errors reduced to the 6th axis for the five 34-parameter procedures (3 kg) 106

Figure 7.19 Orientation errors for the five 34-parameter procedures (3 kg)..............106

LIST OF ABBREVIATIONS AND ACTONYMS

CPC	Complete and parametrically continuous
DH	Denavit-Hartenberg
DHM	Denavit-Hartenberg modified, as defined in (Craig, 1986)
DBB	Double-ball bar
DOF	Degree of freedom
TCP	Tool center point
SMR	Spherically-mounted reflector
$s(\theta)$	$\sin(\theta)$
$c(\theta)$	$\cos(\theta)$
s_i	$\sin(q_i)$
c_i	$\cos(q_i)$
$\mathrm{Trans}(x,y,z)$	Geometrical translation $\{x,y,z\}$
$\mathrm{Rot}(\mathbf{v},\theta)$	Geometrical rotation of θ around unit vector $\mathbf{v}$
RMS	Root mean square
LAN	Local area network

LIST OF SYMBOLS AND UNITS OF MEASUREMENT

kg	Kilogram
g	Gram
N	Newton
Hz	Herz (measures/second)
s	Second
ms	Millisecond
º	Degree (“deg” is used in some figures)
rad	Radian
m	Meter
mm	Millimeter
µm	Micrometer
Nm	Newton-meter (torque)
mm/s	Millimeter per second

INTRODUCTION

Nowadays, two measures are commonly used for describing the positioning performance of industrial robots: *repeatability* and *accuracy*.

Loosely speaking, *pose repeatability* is the ability of a robot to repeatedly return to the same pose. In robotics, repeatability, as defined in ISO 9283 and used by all industrial robot manufacturers, actually refers to *unidirectional repeatability* only, *i.e.*, the ability to return to the same pose from the same direction, thus minimizing the effect of backlash. Multidirectional repeatability can be twice the unidirectional repeatability or even worse.

Repeatability can be improved by either using direct-drive motors (as in some SCARA robots), high-precision gear trains (as in most Staübli robots) or by placing high-resolution encoders at the output of the gear trains. All of these solutions, however, considerably raise the manufacturing cost of an industrial robot.

Loosely speaking, *volumetric* (also called *absolute*) *accuracy* is the ability of the robot to attain a command pose with respect to a fixed reference frame. Since identifying such a reference frame is not always simple, accuracy is most typically tested in relative measurements, *e.g.*, *distance accuracy* is the ability of the robot to displace its tool center point (TCP) a prescribed distance.

Accuracy is obviously affected by the same factors as multidirectional repeatability; actually it is lower bounded by the multidirectional repeatability of the robot. Accuracy is influenced mostly by geometric inaccuracies and elasticity, present in both the links and the transmissions. Fortunately, these two types of errors can be modeled to some extent in a process known as robot *calibration* (Abderrahim *et al.*, 2007).

The demand for industrial robots having better repeatability and higher volumetric accuracy has been constantly growing in the past decade, especially in the aerospace sector (Summers, 2005). Today, most industrial robot manufacturers and a few service providers (such as Dynalog in the USA) offer robot calibration services. Furthermore, most industrial manufacturers now adopt the ISO 9283 norm, which was not the case a decade ago (Greenway, 2000; Schröer, 1999). Nevertheless, the only upfront information regarding the positioning performance of an industrial robot continues to be a single measure specified as "positioning performance according to ISO 9283", which actually refers to the average unidirectional position repeatability and accuracy at five poses obtained from thirty cycles. A few additional performance measures might sometimes be obtained from the robot manufacturer (*e.g.*, found in the product manual of the robot), such as *linear path repeatability* and *linear path accuracy*, but even this information is highly insufficient and impossible to use for comparison purposes.

The absolute accuracy of a given robot is virtually never specified by its manufacturer. The accuracy of a robot is not important as long as poses of the robot end-effector are manually taught. In this case we only want the robot to be repeatable. However, in offline programming the accuracy becomes an important issue since positions are defined in a virtual space from an absolute or relative coordinate system. There are also some industrial applications where a robot is used as a measurement system; in this case, the accuracy of the robot is the accuracy of the measurement system.

Improvement of the robot accuracy requires a study of the direct kinematic model. Using the nominal kinematic model of a robot and adding error parameters we can find a mathematical model that represents the robot better than the nominal kinematic model. This improved model must reduce position and orientation errors, i.e., improve the robot accuracy.

In this thesis, we propose different mathematical models which take into account different combinations of error parameters to represent the positioning behavior of the robot. These calibration methods are applied and tested on an ABB IRB 1600-6/1.45 industrial robot using a Faro laser tracker. The main objective of this project is to improve the accuracy of our robot.

As summarized by (Andrew Liou *et al.*, 1993; Karan and Vukobratovic, 1994) there are five factors that cause robot errors: environmental (such as temperature or the warm-up process), parametric (for example, kinematic parameter variation due to manufacturing and assembly errors, influence of dynamic parameters, friction and other nonlinearities, including hysteresis and backlash), measurement (resolution and discretisation of joint position sensors), computational (computer round-off and steady-state control errors) and application (such as installation errors).

Although robot calibration has been studied for more than two decades, the theory remains quite the same as in the early 1980s (Barker, 1983). What is different nowadays is that robots are built better (i.e., their repeatability is greater) and the sources of errors (with respect to their nominal models) are somewhat different. Measurement equipment is also better, i.e., more accurate, though certainly not much more affordable. The mathematical models that used to work for robots a decade or two ago are no longer optimal for today's robots. Furthermore, the accuracy required today in some potential robot applications is much higher than a decade ago.

Robot calibration can be divided into several categories and subcategories: absolute and relative calibration, open-loop and closed loop calibration, with or without feedback, etc.

1. Absolute vs. relative calibration

An absolute calibration takes into account where the robot base is placed whereas a relative calibration disregards the actual location of the robot base. In other words, if we want more than one robot to share the same coordinate system they need to be "absolute" calibrated to agree with the same "absolute" reference frame (also called world frame). A relative calibration is of interest when we are positioning the robot relatively to a local frame (also called object or user frame), so we need a tool, such as a touch probe, which allows us to locate objects in the robot working space. An absolute calibration needs six more parameters than a relative calibration because we need to represent the relative frame with respect to an absolute frame.

2. Open-loop vs. closed-loop calibration

Whenever we use a measurement system to directly measure the pose of the robot tool, such as a laser tracker, we apply an open-loop calibration. We can find several methods used for measuring robot position as the measuring technology has improved a lot in the past two decades.

Some examples of open-loop methods are acoustic sensors (Stone and Sanderson, 1987), visual systems such as cameras (Meng and Zhuang, 2001; Puskorius and Feldkamp, 1987), coordinate measuring machines (CMM) (Driels *et al.*, 1993; Lightcap *et al.*, 2008; Mooring and Padavala, 1989) and, of course, laser tracking systems (Shirinzadeh, 1998). There has also been some research work that allows a laser tracking system to identify the 6 parameters of the tool pose (Vincze *et al.*, 1994).

On the other hand, a closed-loop method is used if the robot tool is constrained to lie on a reference object of precisely known geometry. This method only needs a switch such as a touch probe to detect the contact with an obstacle. When the robot is placed at the contact position the joint values given by the encoders are registered.

One example of closed-loop calibration is the MasterCal commercial product from American Robot, where the constraints are the diameter of two spheres and the distance between their centers. Other examples are the use of planar constraints (Ikits and Hollerbach, 1997), or point constraints (Meggiolaro *et al.*, 2000) or (Houde, 2006).

3. Thesis organisation

This project is organized into seven chapters. Chapter 1 presents a literature review on robot calibration. Chapter 2 defines the project objectives and methodology. Chapter 3 describes a preliminary kinematic robot calibration performed by moving each axis individually. It also presents the theory of the full kinematic calibration making reference to the work done before using Sklar's method (Mooring *et al*, 1991, p. 177). Once the robot is calibrated, we need to set the base frame and the tool frame to establish a relationship with the external world. Chapter 3 also specifies how to establish an arbitrary and/or absolute base and proposes two tool calibrations: one in case we do not know the end-effector's geometry and another in case we perfectly know this geometry (e.g., by measuring the end-effector on a CMM).

Chapter 4 discusses the axis-by-axis tests performed to the robot. These tests are based on axis identification and angle offset analysis for each joint of the robot, i.e., level 2 calibration. In this chapter, we find that the 6th axis has a peculiar non-linear behavior and that the robot arm weight is high enough to affect the linearity of axes 2 and 3.

Chapter 5 describes the kinematic and non-kinematic error models proposed in this thesis. We show how the compliances of axes 2 and 3 are modeled and how the motion pattern of axis 6 is fitted to a second order Fourier function.

Chapter 6 shows the nominal inverse kinematics and a slightly novel method to iteratively find the inverse kinematic solution of a fully calibrated and generic 6-revolute-axes serial robot.

In Chapter 7 we show all tests we performed to reach satisfactory calibration results. We describe a generic calibration procedure which corresponds to 120 measurements detailing the measurement acquisition, calibration method and verification tests.

CHAPTER 1

LITERATURE REVIEW

In this chapter we describe the generic calibration methods established in the literature, more precisely the robot calibration process, the three levels of robot calibration, the kinematic representation used for calibrated robots and the optimization methods used for parameter identification. We also mention the most relevant commercial solutions for robot calibration and some recent robot calibration results reported in literature.

The kinematic representation allows modeling the robot with parameters that define the geometry of the robot. Once we obtain a model to represent the behavior of the robot it is necessary to find a way to calculate the inverse kinematics for a given pose.

1.1 Robot calibration process

A typical robot calibration process consists of four sequential steps (Roth *et al*, 1987): modeling, measurement, identification and correction. The modeling step consists of representing the real robot through its direct kinematics equations. It is the mathematical model that takes into account the various error parameters. Data from the real robot allows generating the equations that the identification algorithm will use to find an improved robot model, better than the nominal kinematic model.

We could divide a full and complete robot calibration solution in three main steps: tool calibration, robot calibration and inverse kinematics computation.

1.2 Tool calibration

We may usually calibrate the tool at the same time as the robot is being calibrated. However, a separate tool calibration must be taken into account when the tool which we want to be precisely positioned is not the one that we use during calibration.

1.3 Robot calibration

By robot calibration researchers usually mean finding a new direct kinematics model that can represent the real robot better than the nominal model. The nominal model is the one used in the robot controller, and for robots with so-called inline wrists (the axes 4, 5 and 6 intersect at one point), the inverse kinematics of the nominal model are relatively simple and can be solved analytically. A robot calibration implies error parameters inserted to design parameters (nominal model) that represent the real source of errors. These parameters are called error parameters which must be found by the calibration method.

Although optimization algorithms are not primordial when calibrating a robot they can be very helpful improving precision if they are used appropriately. Some optimization algorithms are described in Section 1.6.

A robot calibration can be divided into three levels. The calibration method will be defined depending on which real error factor it represents and how many error parameters it uses. As explained in (Mooring *et al.*, 1991), there are three levels of robot calibration.

1.3.1 Level-1 models

A level-1 calibration is also known as a "joint level" calibration. The purpose is to correctly define the relationship between the desired joint position (θ_d) and the real joint position (θ_r). In a nominal model we consider $\theta_r=\theta_d$, but in real life we have a complex relationship $\theta_r = \mathrm{f}(\theta_d)$. This relationship may be difficult to find but we can reach good approximations with linear functions. The most basic one would be:

$$\theta_r = k_1\theta_d + k_0 \tag{1.1}$$

where k_0 is the offset constant and is close to zero whereas k_1 is the proportionality constant.

1.3.2 Level-2 models

A level-2 calibration is defined as the entire robot *kinematic calibration*. That means that some (or all) of the geometric design parameters are changed. Distance and angle offsets are added as error parameters to the robot's nominal design. At the same time, a level-2 model can include a level-1 model to calibrate the joints.

When an entire kinematic calibration is needed we can identify the robot's joint axes and extract the kinematic parameters placing frames that relate each joint axis with the next one. The calibration needs the "virtual" joint axes all in the same absolute reference frame and the geometry of the tool (end-effector) referred to the robot's tool frame. To extract the virtual axes we must set all robot joints at 0º, and each joint has to be moved one by one taking measures by intervals (Mooring *et al*, 1991, p. 177). A circle that minimizes the sum of error squares can fit these points. From this circle we can extract the axis.

This idea was developed independently by several researchers. Once we have the virtual robot axes there are two basic methods to extract the kinematic parameters: Stone's method and Sklar's method (Mooring *et al*, 1991, p. 177). Stone's method (Stone., 1986) finds the kinematic model known as "S-model" (6 parameters per joint) and Sklar's method finds the DH representation of the robot placing the frames at the appropriate place. Both methods are explained and compared in (Mooring *et al.*, 1991).

1.3.3 Level-3 models

The level-3 calibration takes into account the non-geometrical error sources. Non-geometrical sources of errors can be stiffness, friction, backlash, dynamical parameters, etc. Level-3 calibration is usually combined with a level-2 and level-1 calibration. Most common robot calibrations include a full kinematic calibration (level-2) and sometimes a few parameters describing the stiffness of the robot's arm (level-3).

1.4 Kinematic modeling

For a nominal kinematic modeling, the best-known four-parameter representation is the one given by Denavit-Hartenberg (Denavit and Hartenberg, 1955). This so-called DH notation is widely used in robotics.

There is also a very similar and well-known representation commonly referred to as Denavit-Hartenberg Modified (DHM) notation which is the notation defined by Craig (1986). The main difference between the last two representation methods remains on the order of the geometrical transformations. Both make a translation and rotation over the X and Z axis (one translation and one rotation each). The DH notation starts with the X axis while the DHM notation starts with the Z axis (translation and rotation around the same axis can be alternated with no final effect). For a detailed review of the direct kinematic modeling, see (Craig, 1986; Paul, 1981; Slotine and Asada, 1992).

The DH notation has been used by several researchers for robot calibration, such as Wu (1984) or Ma *et al*. (1994). However, this representation introduces singularity problems when two consecutive axes are parallel or almost parallel (Hayati, 1983). The CPC model eliminates this problem (Zhuang and Roth, 1992) by representing the relationship between each link with three translations and one rotation instead of two translations and two rotations.

Other types of representations have also been used. There is five-parameter representation for prismatic joints (Hayati and Mirmirani, 1985) or even six parameter representation (Stone., 1986), but if we insert more than four parameters the calibration problem becomes redundant.

1.5 Inverse kinematics computation

We should normally describe how we are going to solve the inverse kinematics of the calibrated robot. There are many inverse kinematics solutions but not all of them are suitable to all robot calibration methods. Depending on which type of calibration we use we will need one or another inverse kinematics solution. Inverse kinematics calculation can be divided in two main types: algebraic and iterative.

1.5.1 Algebraic

We should usually try to find an algebraic solution from our direct kinematics model. However, as we add error parameters to our basic kinematic model, the simplifications that we can usually do on a nominal model can no longer be done.

1.5.2 Iterative

When an algebraic solution cannot be found, an iterative method must be applied. This numerical method approaches to the solution at each iteration. Industrial robots often have path motion planners that divide a trajectory into a large quantity of points, and inverse kinematics must be applied to each point of the path.

Any of the optimization methods previously described for robot calibration could be used, however, better optimization methods exist as the problem is more specific. In the worst case we have to find as many parameters as the number of joints that the robot has.

The iterative (or numerical) methods can be divided into two types (Chen and Parker, 1994; Wang and Chen, 1991): (1) Newton-Raphson and predictor-corrector-type algorithms, and (2) optimization techniques by formulating a scalar cost function.

Examples of the first type can be found in (Angeles, 1985) who uses Newton-Raphson method and in (Goldenberg *et al.*, 1987) or (Tsai and Orin, 1987) who use a predictor-corrector-type algorithm. The problems of these methods appear when the Jacobian matrix is singular.

Examples of the second type can be found in (Chen and Parker, 1994; Goldenberg *et al.*, 1987; Goldenberg and Lawrence, 1985; Wang and Chen, 1991). These predictor-corrector-type algorithms are numerically more stable since the Jacobian matrix is not used.

1.6 Optimization algorithms

Once we have defined a model (kinematic or non-kinematic) we must find the error parameters by measures taken from the real robot. The optimization algorithm which is suitable for most types of robot calibration is nonlinear and unconstrained. Different methods and algorithms have been developed, such as the CPC error model (Complete and Parametrically Continuous). It establishes an error model with a minimum number of parameters (Motta and McMaster, 1999; Zhuang and Roth, 1992).

Plenty of algorithms, more precisely the genetic algorithm (Wang, 2009), represent small variations of direct kinematic parameters and the end-effector error is represented by a fitness function. At every generation, a population of parameters is created and brings a better solution to replace the existing solution. This technique does not need complex calculations like the inverse of the Jacobian matrix.

Other alternatives for robot calibration have been tested, like Taguchi method (Karan and Vukobratovic, 1994) or (Judd and Knasinski, 2002). The work (Zhuang and Roth, 2002), for example, uses different methods (similar to the CPC model) to identify the unknown parameters.

All optimization methods can be mainly classified into two types: line-search methods and trust-region methods. We can also mention an optimization method that differs from the first two types: the Nelder-Mead method.

1.6.1 Line-search methods

There are various line-search methods. They differ by the way they compute the line search direction. We can find the following line-search methods: Newton's method, gradient descent method and Quasi-Newton method (Bonnans and Lemaréchal, 2006).

Newton's method is also known as Newton-Raphson method. Newton algorithms are implemented in Matlab's optimization toolbox in the functions fsolve, fminunc and lsqcurvefit.

1.6.2 Trust-region method

The trust-region method is also known as restricted step method. It handles the case when the Jacobian matrix is singular and is useful when the initial guess is far from a local minimum. This method approximates the objective function with a simpler function in the neighborhood of the solution at each iteration.

Trust region methods are dual to line search methods. The first one chooses a step size before a search direction while the second one chooses a search direction and then a step size.

1.6.3 Nelder-Mead

This optimization algorithm was proposed by John Nelder and Roger Mead (Nelder and Mead, 1965). It is also called simplex method (a non linear method that is different from the known linear simplex method). It evaluates the objective function over a polytope in the parameter space. If we have two parameters, the polytope is a triangle as we are in a 2D plane. If there are n-dimensions, we have an (n+1)-sided polytope.

The algorithm compares these n+1 points and deletes the worst one. The worst point is replaced by its reflection through the remaining points in the polytope. This algorithm is simple and does not need gradient information but it takes time to achieve a solution when we have more than six variables. This method is also implemented in Matlab's optimization toolbox, in the function fminsearch.

1.7 Commercial solutions for robot calibration

Most robot manufacturers offer calibration as an option. For example, in the case of ABB Robotics, most (but not all) of its robots can be calibrated at the factory with the CalibWare software for about C$2,000, using a Leica laser tracker, a single SMR (Spherically-mounted reflector) and around 40 error parameters. However, ABB does not offer an on-site calibration service, unlike KUKA. ABB also has a tool to improve resolver offsets due to motor exchange and maintenance: the calibration pendulum.

As an example, L-3 MAS Canada at Mirabel use Motoman industrial robots and have them calibrated on-site by Motoman, who use a third-party calibration software (from Dynalog). Similarly, Messier-Dowty at Mirabel use three KUKA industrial robots and have them calibrated on-site by KUKA.

1.7.1 Dynalog

Dynalog is a Detroit-based privately held company founded in 1990 by Dr. Pierre De Smet, then professor at Wayne State University. Dynalog is by far the most renowned expert in robot calibration. While the company offers several products improving the accuracy of industrial robots, the two of greatest interest are the CompuGauge hardware and the DynaCal software. The first is a 3D (*x, y ,z*) measurement device based on four string encoders that intersect at one point. Dynalog claims that the volumetric accuracy of the CompuGauge measurement device is 0.150 mm and its repeatability is 0.020 mm inside a cubic working volume of side 1.5 m. The price of this device is at least US$9,000, but while not expensive, the device is quite bulky and difficult to install.

DynaCal is software for robot calibration that accepts measurement data from the CompuGauge device or from any other precision 3D or 6D measurement device. The software and here adapters for fixing SMRs are sold to industry for more than US$40,000. While all demonstrations of DynaCal show the use of a laser tracker and a single SMR, it seems that DynaCal can also work with three SMRs, thus calibrating the complete pose of the end-effector.

Dynalog also has a specific patented product to calibrate robots that is going to be used for piece inspections (De Smet, 2001). Dynalog offers a complete robot library which makes it possible to calibrate any robot from any brand.

1.7.2 Nikon Metrology

Metris International Holding was purchased by Nikon in 2009 to create Nikon Metrology. Metris, a market leader for CMM based laser scanning, was founded in 1995 and is headquartered in Belgium. In 2005, Metris acquired Belgium-based Krypton, which was specializing in robot calibration since 1989.

Nikon Metrology offers a large number of metrology systems, but the two that are of particular interest to us are the K-Series Optical CMM and the ROCAL software. The first one is basically a three-camera system that measures the spatial coordinates of up to 256 infrared LEDs (thus, it can provide 6D measurements). The volumetric accuracy of the K-Series Optical CMM is better than 0.090 mm which is close to the laser tracker accuracy and certainly sufficient for robot calibration. Its price is about C$80,000.

ROCAL is software for robot calibration, very similar to Dynalog’s DynaCal. It seems that some of the differences are a better integration with some robot brands (KUKA, Mitsubishi and COMAU) and the software’s incompatibility with measurement devices other than the K-Series Optical CMM. The software also relies on complete pose measurement data.

1.7.3 Teconsult

Teconsult is a Germany based university spin-off offering a unique 3D optional measurement device called ROSY and the robot calibration software that goes with it. Teconsult was founded by Prof. Lukas Beyer in 1999. ROSY is a measuring tool based on a videometric principle with two digital CCD cameras. Two cameras are used in order to get a more uniform volumetric accuracy. The tool is attached to the robot flange and is used to measure, with respect to the robot flange frame, the spatial position of the center of a small white ceramic ball that is fixed with respect to the robot's base. The ROSY device itself is calibrated on a CMM before shipment.

The calibration procedure consists of reorienting the tool and measuring the position of the ball for 40 different poses (Beyer and Wulfsberg, 2004), for a single location of the ceramic ball. According to reference (Beyer and Wulfsberg, 2004) the volumetric accuracy of ROSY is ±0.020 mm inside a spherical measurement range of ±2 mm. However, ROSY is offered in several different sizes, and there is no information whether that volumetric accuracy is for a small or for a large ROSY device.

ROSY is rather bulky and requires removal of the end-effector from the robot. Furthermore, it requires several relatively thick cables to be run along the robot arm. A complete ROSY system for tool, base and robot calibration is about €17,500 (US$21,000). However, it seems that Teconsult does not offer any means to calculate the inverse kinematics.

1.7.4 Wiest AG

Wiest is another Germany based university spin-off offering another unique 3D optical measurement device called LaserLAB and the robot calibration software that goes with it. Wiest AG was founded by Dr. Ulrich Wiest who has been working in the field of robot calibration since 1996 (he obtained his doctoral degree in 2001).

LaserLAB is patent-pending (Wiest, 2003) and consists of five small-range one-dimensional laser distance sensors mounted to a common frame and with their lasers intersecting at a common point. A ball is attached to the end-effector of the robot while the LaserLAB device is stationary. By measuring the five distances to the ball (when the center of the ball is approximately at the lasers intersecting point), the spatial coordinate of the center of the ball with respect to the LaserLAB are determined. The repeatability of the LaserLAB is ±0.020 mm, while its volumetric accuracy is better than ±0.100 mm (typically ±0.035 mm), inside a measurement range of 39.5 mm × 38.5 mm × 36.5 mm.

One disadvantage of the LaserLAB is the high likelihood of the sphere colliding with the measurement device while the robot is re-oriented. Furthermore, the only way to measure with a wide range of robot configurations is to use extension rods of different lengths at the end of which a sphere is mounted, rendering that solution practically inconvenient and therefore realistically inaccurate.

1.7.5 American Robot Corporation

American Robot Corporation (ARC) is a US company based in Pittsburg, Pennsylvania. ARC was established in 1982 and is a manufacturer of industrial robot controllers, industrial robots, and automation systems. It has three major product lines, the Universal Robot Controller, the Merlin articulated six axis robot, and the Gantry 3000 modular gantry robot. ARC also offers a robot calibration software called MasterCal, which makes use of a standard touch probe attached to the flange of a robot and two fixed precision balls separated by a precisely known distance.

The MasterCal calibration procedure was invented and patented by Mr. Wally Hoppe (Hoppe, 2008), a Group Leader and Senior Research Engineer at the University of Dayton Research Institute in Ohio, USA. The basic concept for Mr. Hoppe’s calibration method is an extension of (Meggiolaro *et al.*, 2000), where a single ball-in-socket mechanism was used. Mr. Hoppe’s institution had a huge military contract for robot inspection of aircraft engines and this is how he ended up devising a robot calibration method (he no longer works in robotics). In the course of the patent application, he eventually came across some inventions that are pretty close to this one, although he worked with his lawyer to demonstrate that they do not infringe. The closest method to his invention is by ABB (Snell, 1997). That method uses a single large-diameter precision ball of known diameter and a touch probe. Another very close invention is (Knoll and Kovacs, 2001), which is very general and does not give a lot of detail.

1.8 Recent calibration results reported in the literature

Recent research has been mainly focused on level-2 and level-3 calibrations. An example of a level-3 calibration using a stiffness model is (Lightcap *et al.*, 2008), that applies a torsional spring model to represent the flexibility of the harmonic drives by physically meaningful parameters, this model takes into account the flexibility caused by the end-effector. The model improves the mean / maximum values from 1.77 mm / 4.0 mm to 0.55 mm / 0.92 mm for a Mitsubishi PA10-6CE when loaded at 44 N (validated with only ten measurements on a CMM). This method is more simple than the one proposed in (Khalil and Besnard, 2002) as it does not need the computation of the generalized Jacobian. Also (Caenen and Angue, 1990) represented the angular deformation caused by gravity force. A similar method exists dealing with joint angle dependent errors (Jang *et al.*, 2001).

An example of kinematic calibration is given in (Ye *et al.*, 2006), where an absolute calibration was performed to an IRB 2400/L with a Faro *Xi* laser tracker. The mean position error is reduced from 0.963 mm to 0.470 mm for twenty measurements (maximum values are not given, the area of calibration is not given either).

Another example of absolute calibration with a laser tracker is (Newman *et al.*, 2000). Using a Motoman P8 robot, the 27 error parameters from their kinematic model are identified by measuring 367 targets moving each axis separately. The kinematic model that gave best results (for a validation of 21 measurements) corresponds to a “circle-point” algorithm that improves the RMS error from 3.595 mm to 2.524 mm.

We can finally mention the work performed by (Bai *et al.*, 2003) that uses a modified CPC model (MCPC) (Zhuang *et al.*, 1993) to improve the kinematics of a PUMA 560 with 30 error parameters and a laser tracker measure system. Using 25 measures for parameter identification and 15 measures for verification they reach a mean position error of 0.1 mm, however, when they use a CMM they find that the same position error is 0.4-0.5 mm. The CPC model avoids the singularities associated with parallel axes.

Other examples of stiffness kinematic models that do not use meaningful parameters are (Jang *et al.*, 2001; Meggiolaro *et al.*, 2005). As explained by Lightcap *et al.* (2008), it is better to use meaningful parameters to be able to extrapolate to unknown charges.

In the IRB 1600 product documentation, it is stated that the typical mean/maximum positioning accuracy is 0.300 /0.650 mm. We know from Dr. Torgny Brogardh, that this is validated for one tool target (apparently the same target used for calibration). However, we do not have more information regarding the validation procedure, such as the number of measurement poses.

CHAPTER 2

OBJECTIVES AND METHODOLOGY

This section describes the objectives of this project and the methodology followed.

2.1 Thesis objectives

The main objective of this project is to establish a robust and efficient calibration method for our ABB IRB 1600 industrial robot. Another objective is to investigate and compare the performance of several level-1, level-2 and level-3 calibration methods. No dynamic calibration will be taken into account.

The calibration methods proposed will be compared to the nominal model, which will include the calibration of the robot base and the end-effector. The calibration of the base is taken into account in the nominal model since we are not able to measure the robot base, so the results are more favorable for the robot manufacturer. The calibration of the tool is also taken into account since we do not have the exact measures of the tool with respect to the tool flange of the robot.

Once the robot parameters are identified, we need to modify the original desired targets (poses) into so-called “fake” targets. To do so, we need to use an iterative inverse kinematics algorithm based on the new direct robot model to calculate the corrected joint values for the desired end-effector pose.

Once we have an improved model of the robot and the procedure to generate the fake targets is established, we will be able to validate our model. To validate our models we are going to use a large number of measures in contrast to other researchers. Many researchers and companies usually use from 50 to 100 measures to find a calibrated model and no more than 50 measures to validate the model found. In this case it is not possible to have a good estimation of the maximum error value found with the new model. However, we will use about 1000 measures to validate our models.

We will also validate our models with at least three points from the tool (most validation tests will be performed with eight tool-point measurements). So if we improve the position error of at least three tool-points (which are not in the same line), this will mean that we also improve the orientation errors. In addition, tests for evaluating the improvement of the orientation errors will also be performed.

2.2 Methodology

All calibration models will be tested with an ABB IRB 1600-6/1.45 robot and using a Faro ION laser tracker as a measurement system. They are both controlled by a PC running Matlab and an Ethernet local area network (LAN).

A laser tracker is relatively simple to use and can quasi-continuously (typically every millisecond) measure the position of a single SMR or even measure the complete pose in static mode (by measuring the positions of three SMRs, however, using ADM only), in the entire workspace of an industrial robot. Unfortunately, laser trackers are excessively expensive ($100K or more) and very sensitive to air turbulences. Furthermore, their volumetric accuracy and repeatability is much worse than that of two high-precision single-axis measurement instruments commonly used for machine tool calibration: the laser interferometer system and the telescoping bar. These are, however, not described in the ISO guide and are rarely used in conjunction with industrial robots.

The mathematic representation chosen to model the kinematics of the robot is the one defined by Craig (Craig, 1986) and also known as the DHM notation (Denavit Hartenberg Modified). Using the methods explained by Mooring *et al* (1991, p. 177) and adapting the kinematic parameter extraction to the DHM notation we can find the kinematic model of a robot. In addition we consider some additional error parameters representing the behavior of the 6th axis and the stiffness of the robot.

An end-effector was manufactured to hold nine spherically-mounted reflectors (SMRs). This end-effector allows a wide range of poses in which at least one (or three) SMR are visible by the laser tracker. The weight of this end-effector is 3 kg (holding all SMRs). Extra weight can be added by up to eight steel discs (each disc weights 375 g), so the maximum end-effector weight is 6 kg.

CHAPTER 3

KINEMATIC CALIBRATION

This chapter describes the first step of our robot calibration study, namely, what we can loosely refer to as preliminary robot calibration. Basically, we directly identify each of the six robot axes by rotating the joints one by one.

The base frame should normally be measured directly, by measuring the mounting pattern on the bottom of the robot's base. However, once the robot is installed, this can only be done if the robot was mounted using the guide holes on a precise base plate with, for example, three tooling balls (the location of the centers of which are precisely known with respect to the robot's base mounting pattern). In our case, and in most industrial installations, such a precise mounting base plate is not used. Similarly, yet this is much simpler, the geometry of the tool can be measured exactly on a CMM (i.e., the location of the TCP with respect to the flange reference frame, i.e., *tool0* for an ABB robot). However, this is not always done in practice. Hence, we find the optimal base and tool reference frames through two separate calibration processes.

Finally, when the kinematic parameters (concerning the base, the robot geometry and the tool) have been found, we can add other error parameters that do not correspond to geometric errors, as explained in Chapter 5.

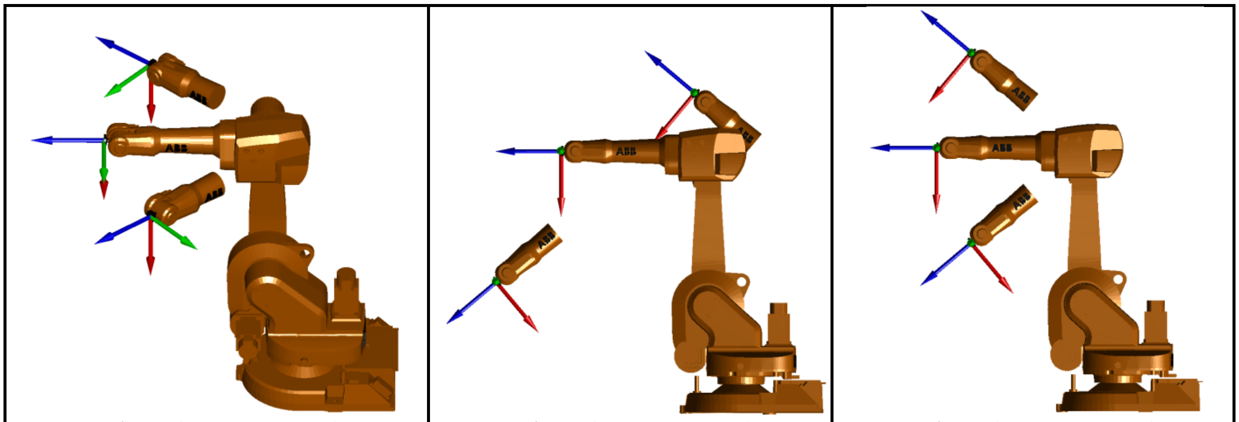
3.1 Kinematic parameter extraction from identified joint axes

A complete and full kinematic calibration is explained, which neither needs special computer power nor iterations to minimize an error function. The procedure is similar to other procedures explained by Mooring *et al.* (1991, p.177), who made an extraction of kinematic parameter from joint axes for a DH model (Sklar method) and for an S-model (Stone method). The calibration needs the "virtual" joint axes all in the same absolute reference and the dimensions of the used tool referred to the robot's tool frame.

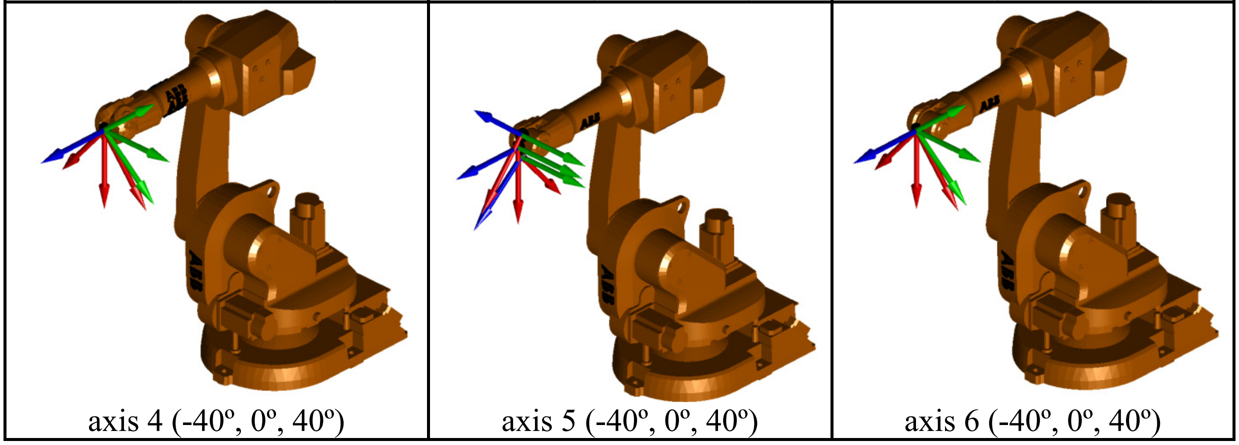
If the errors in the real robot were due only to kinematic errors, with this entire kinematic calibration we can extract the real geometric links of the robot obtaining an error equivalent to the noise measurement. That's what we found by simulating the method explained in this chapter. In simulation, if the measurement noise is forced to zero, the position error is zero.

3.1.1 Method to find the axis of each joint

Starting from the home configuration in which all robot joints are at 0º, each joint has to be moved one by one in equal increments, each time measuring the position of the TCP. A circle that minimizes the sum of all error squares can be fitted to these points for each axis. From this circle we can extract the robot axis.



axis 1 (-40º, 0º, 40º) axis 2 (-40º, 0º, 40º) axis 3 (-40º, 0º, 40º)



axis 4 (-40º, 0º, 40º) axis 5 (-40º, 0º, 40º) axis 6 (-40º, 0º, 40º)

Figure 3.1 Axis-by-axis rotation for identification

Although we need to fit a number of points in 3D space to a circle, for which many algorithms exist, they all require an initial estimate for the center and the plane of that circle. This initial estimate can be obtained analytically from any three points, ideally equally distanced along the circle.

Let the coordinates of three points P_1, P_2, and P_3, be denoted by the vectors $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$, respectively. Also let the unit vector that is normal to the plane (that contains these three points) be denoted by $\mathbf{v}_{\text{axis}}$:

$$\mathbf{v}_{\text{axis}} = \frac{(\mathbf{p}_1 - \mathbf{p}_2)\times(\mathbf{p}_3 - \mathbf{p}_2)}{\left\|(\mathbf{p}_1 - \mathbf{p}_2)\times(\mathbf{p}_3 - \mathbf{p}_2)\right\|} \tag{3.1}$$

It can be easily shown that the coordinates x, y, and z, of the center of the circle are the solution of the following system of linear equations (representing three planes passing through the center of the circle, the first also passing through all three points, the second also normal to line $\mathbf{p}_1\mathbf{p}_2$ and the third also normal to $\mathbf{p}_3\mathbf{p}_2$):

$$\det\left(\begin{bmatrix} \mathbf{p}_c & \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 \\ 1 & 1 & 1 & 1 \end{bmatrix}\right) = 0 \tag{3.2}$$

$$2\mathbf{p}_c^T(\mathbf{p}_1 - \mathbf{p}_2) + \mathbf{p}_2^T\mathbf{p}_2 - \mathbf{p}_1^T\mathbf{p}_1 = 0 \tag{3.3}$$

$$2\mathbf{p}_c^T(\mathbf{p}_3 - \mathbf{p}_2) + \mathbf{p}_2^T\mathbf{p}_2 - \mathbf{p}_3^T\mathbf{p}_3 = 0 \tag{3.4}$$

where $\mathbf{p}_c=[x,y,z]^T$. A different solution is explained in (Schneider and Eberly, 2003, Section 13-10).

Once the initial estimate for $\mathbf{v}_{\text{axis}}$ and $\mathbf{p}_c$ is found, we can use the Gauss-Newton least-squares fitting method (Gander *et al.*, 1994).

3.1.2 Placing the link frames

Once we find the axes of all joints, the next step is to place the frame of each joint i with the information of the axis i and the axis i+1. All frames must be referenced to the same absolute coordinate system, which can be the base frame of the measure system (a particular case for the robot's base and for parallel axis is explained below).

To make all transformations suitable to a DHM representation we will use the information between two consecutive axes (like the common normal line, its distance or the angle between two axes). Figure 3.2 shows two consecutive axes and the common normal line that connects them (dotted line). To place the frame *i* we must follow next steps if a DHM notation is used.

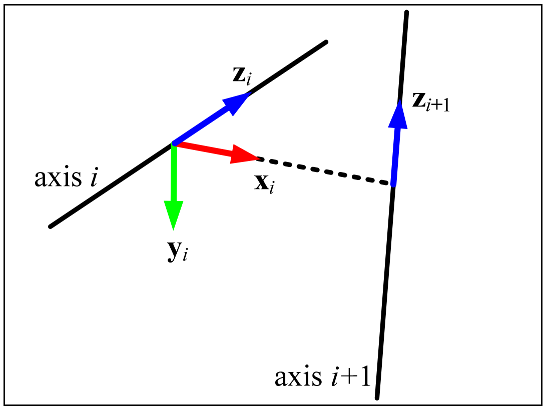


Figure 3.2 Placing frame *i*

We will always place first the vector $\mathbf{z}_i$ in the direction of the i^{th} axis. The origin is at the intersection between the i^{th} axis and the common normal line (or the intersection of both axes if they intersect). The component $\mathbf{x}_i$ is found by $\mathbf{x}_i = \mathbf{z}_i \times \mathbf{z}_{i+1}$, which corresponds to the common normal vector if the two axis are not coincident. Finally, we have that $\mathbf{y}_i = \mathbf{z}_i \times \mathbf{x}_i$.

In this way it is possible to make a DHM representation suitable between two consecutive frames. This happens because only with a translation and rotation around the common normal line (vector $\mathbf{x}_i$) we can make the transformation from $\mathbf{z}_i$ to $\mathbf{z}_{i+1}$. After that, it is possible to make a rotation and translation around the $\mathbf{z}_{i+1}$ axis to place the origin, $\mathbf{x}_{i+1}$ and $\mathbf{y}_{i+1}$ axis of the frame *i*+1 where necessary according to axis *i*+2.

However, a DHM representation will not be possible from an arbitrary base frame F_0 because its **z** vector will never be perfectly perpendicular to the base plane **xy** (F_0 has its **z** vector in the same direction as the first axis, see Section 3.2).

3.1.3 Dealing with parallel axes

When we have two parallel axes we can use a threshold tolerance to consider those axes parallel or not. In other words, two axes will never be parallel because there is an alignment angle error. If this error is considerable we can use the method described before to place the frames.

If the alignment error is very small it is possible to force both axes to be parallel. That means that the center point of each axis is kept but we apply the same direction for both of axes. In this case, the center is the origin of the new frame.

In the case where the alignment error is very small but we do not force the axes to be parallel, if we use the DHM model, a small variation of parameters can result into big kinematic changes. In this case it is important to take into account that the CPC model can be more suitable than the DH or DHM models since this problem is attenuated.

3.1.4 Error parameters needed for a full calibration

As described before, all link transformations can suit a DHM representation except for the first one, which needs two extra error parameters if we consider an arbitrary base. So we have six parameters for the first joint and four parameters for the other five joints which make 26 error parameters to determine. If we want a relative calibration we can skip the first six parameters of the base frame.

Finding those parameters is direct once we have identified the axes of the robot, finding the axes may take a few seconds if there are a lot of points that define each axis. No computational power is needed and the solution can be obtained directly.

We mentioned that 26 parameters are required to complete the robot calibration. However, in this robot calibration the end-effector is represented by a DHM transformation, so only two parameters are taken into account for the position of the TCP relative to the end-effector due to the kinematic representation chosen (which only takes into account 4 parameters per link). In this case, if we want to consider the TCP as part of the calibration process we must add one error parameter (because to define a position with respect to a frame we need 3 parameters). Summarizing, a total of 27 error parameters are needed for a complete kinematic calibration for the position the TCP with respect to an arbitrary base. This corresponds to the minimality justification from Bernhardt and Albright (1993, p.163).

3.1.5 Homogeneous transformation

Sometimes we will need to represent a homogeneous transformation in its six descriptive parameters. A 4×4 homogeneous transformation matrix can hold up to six parameters. Here we show one method to extract these six parameters. This transformation is represented by a translation along the $\mathbf{x}_i$, $\mathbf{y}_i$ and $\mathbf{z}_i$ axes followed by rotations about the $\mathbf{x}_i$, $\mathbf{y}_i$ and $\mathbf{z}_i$ axes consecutively:

$$\mathbf{A}_i^j = \mathrm{Trans}(x_i, y_i, z_i)\,\mathrm{Rot}(\mathbf{x}, \alpha_i)\,\mathrm{Rot}(\mathbf{y}, \beta_i)\,\mathrm{Rot}(\mathbf{z}, \gamma_i), \tag{3.5}$$

$$\mathbf{A}_i^j = \begin{bmatrix} \mathrm{c}(\beta_i)\mathrm{c}(\gamma_i) & -\mathrm{c}(\beta_i)\mathrm{s}(\gamma_i) & \mathrm{s}(\beta_i) & x_i \\ \mathrm{s}(\alpha_i)\mathrm{s}(\beta_i)\mathrm{c}(\gamma_i)+\mathrm{c}(\alpha_i)\mathrm{s}(\gamma_i) & -\mathrm{s}(\alpha_i)\mathrm{s}(\beta_i)\mathrm{s}(\gamma_i)+\mathrm{c}(\alpha_i)\mathrm{c}(\gamma_i) & -\mathrm{s}(\alpha_i)\mathrm{c}(\beta_i) & y_i \\ -\mathrm{c}(\alpha_i)\mathrm{s}(\beta_i)\mathrm{c}(\gamma_i)+\mathrm{s}(\alpha_i)\mathrm{s}(\gamma_i) & \mathrm{c}(\alpha_i)\mathrm{s}(\beta_i)\mathrm{s}(\gamma_i)+\mathrm{s}(\alpha_i)\mathrm{c}(\gamma_i) & \mathrm{c}(\alpha_i)\mathrm{c}(\beta_i) & z_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.6}$$

It is possible to extract the original parameters $\{x_i, y_i, z_i, \alpha_i, \beta_i, \gamma_i\}$ if we have the generic $\mathbf{A}_i^j$ transformation matrix. The translation $\{x_i, y_i, z_i\}$ can be directly obtained. Angles can be obtained proceeding with next steps, imposing the cosine positive for β_i to obtain values closer to zero:

$$\beta_i = \operatorname{atan2}\left(A^j_{i\,1,3}, +\sqrt{1-\sin^2(A^j_i)} \right) \tag{3.7}$$

Once we obtain β_i, α_i and γ_i are completely defined:

$$\alpha_i = \operatorname{atan2}\left(\frac{-A^j_{i\,2,3}}{\cos(\beta_i)}, \frac{A^j_{i\,3,3}}{\cos(\beta_i)} \right) \tag{3.8}$$

$$\gamma_i = \operatorname{atan2}\left(\frac{-A^j_{i\,1,2}}{\cos(\beta_i)}, \frac{A^j_{i\,1,1}}{\cos(\beta_i)} \right) \tag{3.9}$$

Obviously, this generic representation can suit a DHM representation but not vice versa.

3.2 Base frame definition

In robotics, the **xy** base plane is placed perpendicularly to the first axis ($\mathbf{z}_0$ is placed along the first robot axis). This plane will never be exactly the same as the "tangible" reference plane (the plane of the base mount) since we are looking for a geometrical error. Consider Figure 3.3, where the first frame $\mathbf{F}_1$ is placed as explained before and errors are exaggerated.

The definition of the base is important if we want to calibrate the robot for a specific, physical and measurable frame (see the comparison between an absolute and a relative calibration in the introduction). We could call $\mathbf{F}_B$ the real base frame (composed by $\{\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\}$) and $\mathbf{F}_0$ the nominal base frame (composed by $\{\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0\}$). The first one is tangible but the second one is not. It is possible to find the second one by the information of the first and second axes and the nominal parameters.

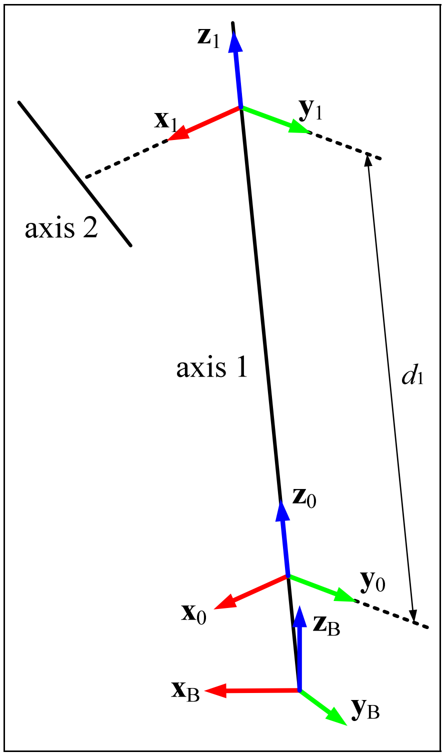


Figure 3.3 Position of the nominal and real robot base frame

In the case where we consider the base frame as the “official” tangible base frame, provided by the manufacturer, $\mathbf{F}_0$ and $\mathbf{F}_B$ should be very close. If the first and second axis were exactly in its nominal position (so there are no errors for the first and second axis), frames $\mathbf{F}_0$ and $\mathbf{F}_B$ (nominal and real robot base frame respectively) would be exactly the same. The real base frame $\mathbf{F}_B$ is found by measurements on the robot base plane (which is the same as $\mathbf{x}_B\mathbf{y}_B$ plane) and has the origin on the intersection of this plane with the first axis line. $\mathbf{z}_B$ is perpendicular to the robot’s real floor plane and $\mathbf{x}_B$ and $\mathbf{y}_B$ can be placed arbitrarily on the floor plane.

The nominal base frame $\mathbf{F}_0$ is found by a translation along $-\mathbf{z}_1$ of the nominal distance d_1 from the $\mathbf{F}_1$ frame (no error parameters are included in this step):

$$\mathbf{F}_1 = \mathbf{F}_0 \ \mathrm{Trans}(0,0,d_1) \tag{3.10}$$

In this way we can compare the relationship between $\mathbf{F}_B$ and $\mathbf{F}_0$.

$$\mathbf{F}_0 = \mathbf{H}_B^0 \; \mathbf{F}_B \tag{3.11}$$

$$\mathbf{H}_\mathrm{B}^0 \cong I_4 \tag{3.12}$$

Ideally, $\mathbf{H}_\mathrm{B}^0$ should be equal to the identity matrix. However, with errors it can hold up to six parameters for a geometrical transformation. If our base frame definition is important (in other words, if the link between $\mathbf{F}_\mathrm{B}$ and $\mathbf{F}_0$ is important) we must use those six parameters. Otherwise, we have six parameters to place this $\mathbf{F}_\mathrm{B}$ frame wherever we want or we can force those six error parameters to be zero (then, $\mathbf{F}_0$ and $\mathbf{F}_\mathrm{B}$ are coincident).

To have an idea, if we consider that the robot base is the nominal base $\mathbf{F}_0$ for the ABB IRB 1600-6/1.45 robot, we find a maximum position error of 2.1 mm for 337 measurements. However, if we consider that the real robot base $\mathbf{F}_\mathrm{B}$ has its origin placed at the intersection of the tangible robot base plane **xy** with the 1st axis, $\mathbf{z}_\mathrm{B}$ perpendicular to this plane and $\mathbf{x}_\mathrm{B}$ through the projection of $\mathbf{x}_0$ to the floor plane, we get errors up to 15 mm for the same 337 measures.

3.3 Finding the base from three points

Usually we will not be able to measure the exact position of the robot's base. We must find some characteristic points on the floor plane where the robot is placed. We describe here one method to find the origin $\mathbf{p}_\mathrm{B}$ and $\mathbf{x}_\mathrm{B}$, $\mathbf{y}_\mathrm{B}$, $\mathbf{z}_\mathrm{B}$ axis of the robot base from the 3 points shown in Figure 3.4.

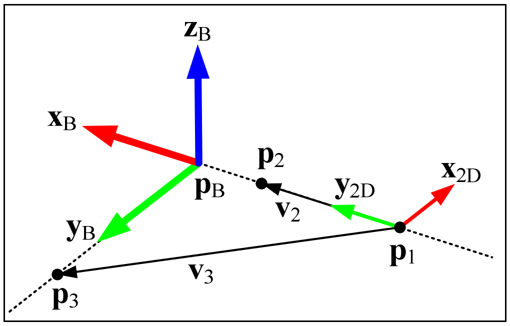


Figure 3.4 Three points that determine the base frame

These three points are:

1. The first one on the axis $\pm\mathbf{x}_B$.
2. The second one on the axis $\pm\mathbf{x}_B$, this one must be on the direction of $+\mathbf{x}_B$ in comparison to the first point.
3. The third one on the axis $+\mathbf{y}_B$.

From the figure we can obtain $\mathbf{x}_B$, $\mathbf{y}_B$, $\mathbf{z}_B$:

$$\mathbf{v}_2 = \mathbf{p}_2 - \mathbf{p}_1 \tag{3.13}$$

$$\mathbf{v}_3 = \mathbf{p}_3 - \mathbf{p}_1 \tag{3.14}$$

$$\mathbf{x}_B = \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|} \tag{3.15}$$

$$\mathbf{z}_B = \frac{\mathbf{v}_2 \times \mathbf{v}_3}{\|\mathbf{v}_2 \times \mathbf{v}_3\|} \tag{3.16}$$

$$\mathbf{y}_B = \mathbf{z}_B \times \mathbf{x}_B \tag{3.17}$$

Finally, to find the origin $\mathbf{p}_B$ we can project the point $\mathbf{p}_3$ on the line $\mathbf{p}_1 + t\mathbf{v}_2$ (Schneider and Eberly, 2003):

$$t_0 = \frac{\mathbf{v}_2^{\mathrm{T}}(\mathbf{p}_3 - \mathbf{p}_1)}{\mathbf{v}_2^{\mathrm{T}}\mathbf{v}_2} \tag{3.18}$$

$$\mathbf{p}_B = \mathbf{p}_1 + t_0\mathbf{v}_2 \tag{3.19}$$

3.4 End-effector calibration

An end-effector was manufactured to hold all of our nine spherically-mounted reflectors (SMRs), namely five 1.5-inch and four 0.5-inch SMRs. This end-effector increases the range of poses in which at least three SMRs are visible by the laser tracker, thus allowing measurements of the complete pose of the robot end-effector.

In this section we explain two methods to calibrate the tool: one that can always be used and does not need the exact dimensions of the tool and another one that we can use if we have the exact dimensions of the tool. For our tests we used the one that does not need the exact dimensions of the tool.

3.4.1 End-effector design

The dimensions of the designed tool are 200×200×100 mm. It is made of aluminum and weighs 1.8 kg, without any of the possible attachments. Five 1.5-inch SMR targets can be added in the front side and four 0.5-inch SMR targets in the back side. These nine targets with their magnetic mounts (also called nests) weigh a total of 1.2 kg.

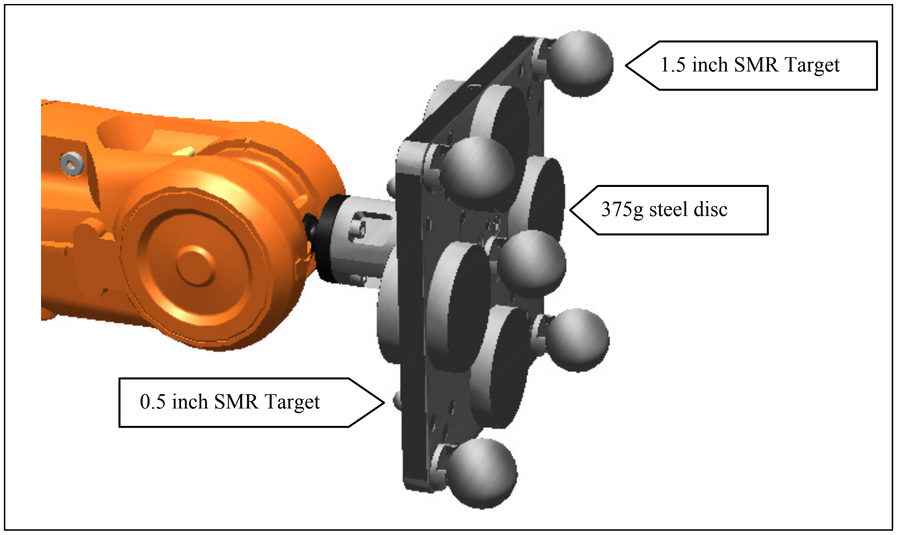


Figure 3.5 End-effector used for holding the nine SMRs

Extra weight can be added by attaching steel discs. Each disc weighs 375 g and we can add up to eight discs (up to 3 kg can be added, which makes 6 kg the maximum payload). With all nine SMRs and eight steel discs attached, the center of gravity of the tool is at 75 mm along the Z direction with respect to the robot tool frame *tool0*.

3.4.2 End-effector calibration for unknown dimensions

To perform the end-effector calibration we must find the 6th frame according to the calibration method used. For an arbitrary position of the end-effector (all five 1.5” SMRs must be visible by the laser tracker at this position) we must identify the 5th and 6th axis by moving them independently and taking at least three measures per axis, as shown in Figure 3.6.

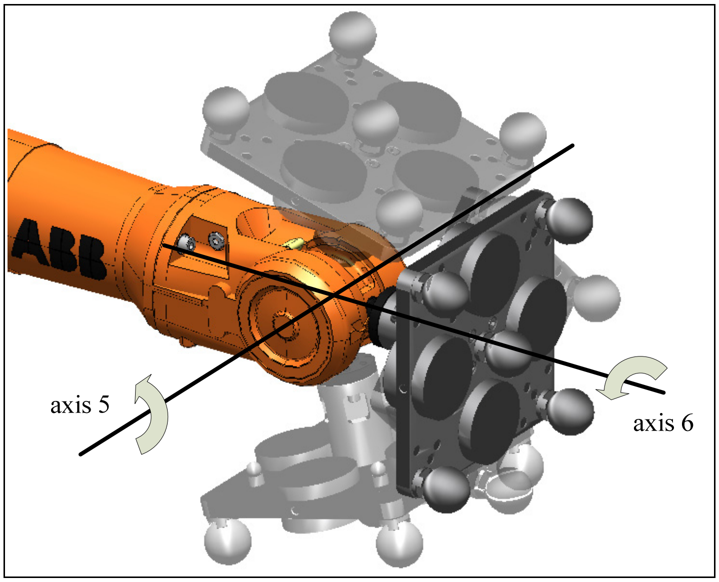


Figure 3.6 Identifying 5th and 6th axes

Once we have identified the 5th and 6th axes we must find their common normal line (dotted line in Figure 3.7), and we will impose that $\mathbf{x}_6$ has the same direction as this common normal line. Ideally, those two axes should intersect but in real life there will always be an error so we will be able to find this common normal line. We have then:

$$\mathbf{x}_6 = \mathbf{z}_6 \times \mathbf{z}_5 \tag{3.20}$$

$$\mathbf{y}_6 = \mathbf{z}_6 \times \mathbf{x}_6 \tag{3.21}$$

With this information we can virtually find the frame represented as 6’ (see Figure 3.7) placing the origin at the intersection between the common normal line and the 6th axis (errors are exaggerated in the figure). At this point we do not have enough information to place the real 6th frame which corresponds to the *tool0* of the robot. But this information is not necessary as we found the end-effector referenced to a frame which is similar but not the 6th frame. In the case where we know the exact dimensions of the tool, the method is explained in the next section.

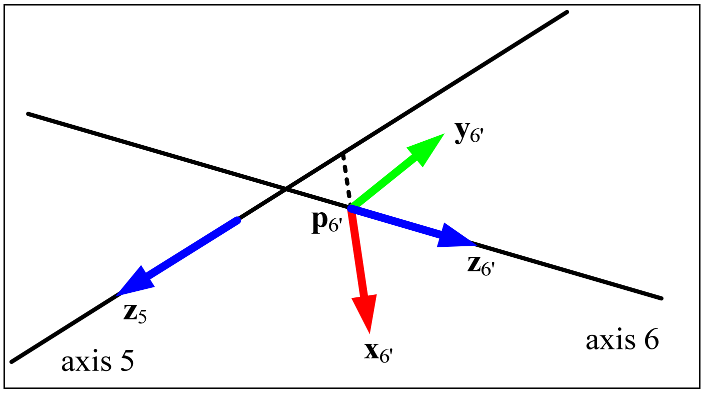


Figure 3.7 Placing the 6th frame

Once the 6th frame is virtually placed we can translate it by the nominal distance through the $\mathbf{z}_6$ direction to obtain a good approximation of the real 6th frame. Then we can take the measures of all targets. All measures must work on the same arbitrary and absolute base. Taking $\mathbf{t}_i^{\text{abs}}$ as one target measure and $\mathbf{F}_6^{\text{abs}}$ as the tool reference frame we can find the coordinates of this target referenced to the 6th frame. So once $\mathbf{t}_i^{\text{abs}}$ and $\mathbf{F}_6^{\text{abs}}$ are known by measures with the laser tracker we can find $\mathbf{t}_i^{F_6}$:

$$\mathbf{t}_i^{\text{abs}} = [x_i, y_i, z_i, 1]^T \tag{3.22}$$

$$\mathbf{F}_6^{\text{abs}} = \begin{bmatrix} x_{6,x} & y_{6,x} & z_{6,x} & p_{6,x} \\ x_{6,y} & y_{6,y} & z_{6,y} & p_{6,y} \\ x_{6,z} & y_{6,z} & z_{6,z} & p_{6,z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.23}$$

$$\mathbf{t}_i^{F_6} = (\mathbf{F}_6^{\text{abs}})^{-1}\ \mathbf{t}_i^{\text{abs}} \tag{3.24}$$

3.4.3 End-effector calibration for known dimensions

This method is useful if we need to use different tools in the same robot. In this case we must know the exact dimensions of all the tools that we are going to use in the robot. Using one of these tools with an SMR target we can calibrate the position of the robot tool frame.

To find the exact position of the last frame (or robot tool frame, $\mathbf{F}_6$ for a 6-DOF robot) we must know the exact dimensions of the end-effector that we are using to take measures. If this is not the case, the method explained in the previous section should be used. If the dimensions of the tool are known we just need one measurement $\mathbf{p}_\text{T}$ to place the last frame.

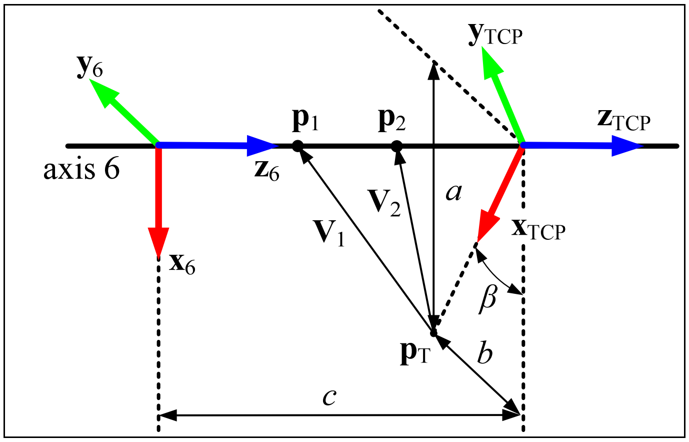


Figure 3.8 Geometry to find the 6th frame

From the figure we can say that $\mathbf{A}_{\mathrm{TCP}}^{6} = \mathbf{F}_6 \,\mathrm{Trans}(0,0,c)\,\mathrm{Rot}(\mathbf{z},\beta)$ if $\mathbf{z}_{\mathrm{TCP}} = \mathbf{z}_6$. So by finding the robot TCP frame we can find the 6th frame. The robot end-effector frame is found as follows:

$$\mathbf{z}_{\mathrm{TCP}} = \mathbf{z}_6 \tag{3.25}$$

$$\mathbf{v}_1 = \mathbf{p}_1 - \mathbf{p}_{\mathrm{T}} \tag{3.26}$$

$$\mathbf{v}_2 = \mathbf{p}_2 - \mathbf{p}_T \tag{3.27}$$

$$\mathbf{y}_{\mathrm{TCP}} = \mathbf{v}_1 \times \mathbf{v}_2 \tag{3.28}$$

$$\mathbf{x}_{\mathrm{TCP}} = \mathbf{y}_{\mathrm{TCP}} \times \mathbf{z}_{\mathrm{TCP}} \tag{3.29}$$

where $\mathbf{p}_1$ and $\mathbf{p}_2$ are two arbitrary points on the 6th axis (6th axis is known from circle movements). We know β as we know the position of the point $\mathbf{p}_{\mathrm{T}}$ in $\mathbf{F}_6$ coordinates:

$$\beta = \mathrm{atan2}(b, a) \tag{3.30}$$

3.4.4 Experimental values

Using the calibration methodology explained in Section 3.4.2 we can compare the coordinates of the nine targets of the tool that was used. To identify the axis we moved the 5th and 6th axis independently from −90º to 90º by steps of 30º. Table 3.1 shows the *xyz* coordinates of the CAD design and they are compared with the real values obtained by measures with the laser tracker.

Table 3.1 *xyz* coordinates of the 9 targets (CAD + real points)

Target	x_{CAD} (mm)	y_{CAD} (mm)	z_{CAD} (mm)	x_{Real} (mm)	y_{Real} (mm)	z_{Real} (mm)	Error (mm)
1	−87.300	87.300	109.480	−87.370	87.202	110.020	0.553
2	84.300	84.300	109.480	84.419	84.260	109.928	0.465
3	85.800	−85.800	109.480	85.990	−85.788	109.959	0.515
4	−86.300	−86.300	109.480	−86.207	-86.339	110.039	0.568
5	−59.500	59.500	56.800	−59.465	59.477	57.489	0.690
6	63.500	63.500	56.800	63.460	63.527	57.407	0.609
7	62.000	−62.000	56.800	62.071	−61.982	57.408	0.612
8	−63.000	−63.000	56.800	−62.918	−63.055	57.454	0.661
9	0.000	0.000	109.480	0.049	−0.032	109.974	0.497

3.5 Kinematic parameter optimization

Once we obtain a first geometrical approach of the real parameters we try to improve those parameters by computing a minimization of an objective function. This objective function is the result of the maximum error of all desired points. In this step we may incorporate 6 additional error variables (k_1,... ,k_6) so that each motor has a linear model error, with offset and proportional constants:

$$\theta_i' = \theta_{0,i} + k_i\theta_i \tag{3.31}$$

It is not needed to incorporate new error variables for $\theta_{0,i}$ because we already have θ_i from DHM table that describes this offset angle error. So now the error model can have up to 26+6=32 variables:

$$\mathbf{H}(\mathbf{q}) = \mathbf{A}_{\text{BASE}}\mathbf{A}_1^0\mathbf{A}_2^1\mathbf{A}_3^2\mathbf{A}_4^3\mathbf{A}_5^4\mathbf{A}_6^5\mathbf{A}_{\text{TCP}} \tag{3.32}$$

$$\mathbf{q} = [\theta_1\,\theta_2\,\theta_3\,\theta_4\,\theta_5\,\theta_6\,]^{\mathrm{T}} \tag{3.33}$$

where

$$\mathbf{A}_i^{i-1} = \mathbf{F}_i = \text{Rot}(\mathbf{x},\alpha_i)\,\text{Trans}(a_i,0,0)\,\text{Rot}(\mathbf{z},\theta_{0,i}+k_i\theta_i)\,\text{Trans}(0,0,d_i) \tag{3.34}$$

Actually we could see that the error parameters needed are 5 for each link plus 2 extra for the base definition (6×5+2=32). The objective error function for *n* measures is:

$$\mathbf{T} = \arg\min_{\mathbf{T}} \left(\max \left(\mathbf{E} \right) \right) \qquad 1 \le j \le n \tag{3.35}$$

where **E** is the vector of errors E_j between the measured position $\mathbf{p}_j = \left[p_{j,x}\ , p_{j,y}, p_{j,z} \right]^{\mathrm{T}}$ and the position given by the direct kinematic model for the same joints $\mathbf{q}_j$, and **T** is the vector containing the error parameters. Taking into account that $\mathbf{H}_j\mathbf{=H(q}_j\mathbf{)}$, we can represent the error for one point:

$$E_j = \sqrt{(p_{j,x} - H_{j,1,4})^2 + (p_{j,y} - H_{j,2,4})^2 + (p_{j,z} - H_{j,3,4})^2} \tag{3.36}$$

In this function we have *n* points, so the objective function is the maximum error of all tested measures.

CHAPTER 4

ANALYSES OF THE BEHAVIOR OF EACH JOINT

In this section we study the behavior of each individual joint by rotating it and measuring the TCP with a laser tracker (Figure 4.1). We will see that this behavior is not linear, especially for joints 2, 3 and 6.

Figure 4.1 shows the installation that was used for measuring the positioning performance of an ABB IRB1600-6/1.45 robot with a Faro laser tracker ION (ADM option only). According to the specifications, the typical accuracy of this laser tracker when measuring the length of a 2.3 m horizontal scale bar distanced at 2 m is 22 μm. According to our own tests, the largest error when measuring the length of a 1 m horizontal scale bar distanced at 2.7 m is 22 μm, but when the bar is vertically oriented, the error is 50 μm. According to our tests, the tracker is more precise when the scale bar is distanced at 5 m. Unfortunately, however, our installation does not allow us to place the laser tracker farther from the robot.

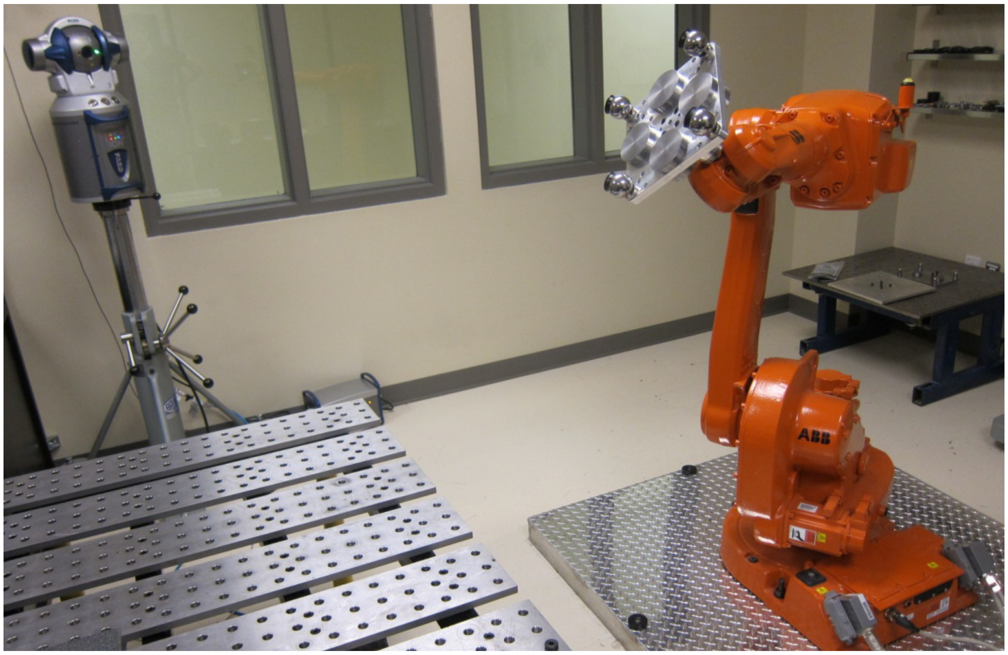


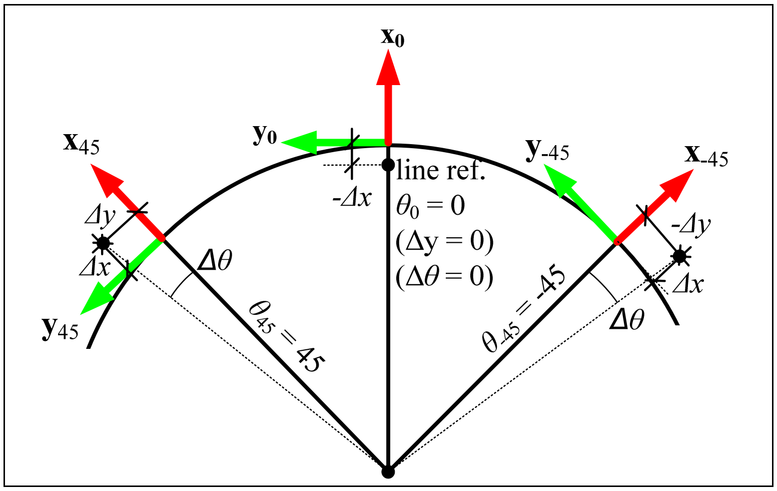
Figure 4.1 The IRB 1600 robot of which the positioning performance was analyzed using a Faro laser tracker

All measurements were performed at temperatures varying between 22°C and 23°C. During all measurements, the laser tracker was never displaced from its location shown in Figure 4.1, in order to avoid the accumulation of errors. All results shown in this chapter correspond to measures taken with the target number 1 (see Table 3.1). This target is going to be considered as the TCP in this chapter.

The laser tracker and the robot are controlled by Matlab via Ethernet LAN. The position commands sent to the robot are the joint values (i.e., we send the joints for each position and use the instruction MoveAbsJ, which corresponds to a joint movement in the RAPID programming language).

4.1 Axis error analysis

Moving the joints one by one and taking measures at the end-effector, a frame is placed on each point of the circle to analyze the error. This frame has the **x** axis in the radius direction, the **y** axis in the tangential direction of the circle and the **z** axis in the circle axis direction (shown in Figure 4.3). Errors are called as "deltas" (Δx, Δy, Δz and $\Delta\theta$ for the angle). We have imposed a "zero" reference for the angle 0º. For this point $\Delta\theta=\Delta y=0$ (there may be an error in **x** and **z** direction).



x_0
y_0
$-\Delta x$
line ref.
$\theta_0 = 0$
$(\Delta y = 0)$
$(\Delta\theta = 0)$
x_{45}
Δy
Δx
y_{45}
$\Delta\theta$
$\theta_{45} = 45$
y_{-45}
x_{-45}
$-\Delta y$
Δx
$\Delta\theta$
$\theta_{-45} = -45$

Figure 4.2 Axis error analysis

The ranges and steps used for the axis analysis are shown in Table 4.1:

Table 4.1 Range of motion for each joint

Joint	Range in degrees	Step in degrees
1	[−40, 40] & [40, −40]	10
2	[−50, 50] & [50, −50]	1
3	[−50, 50] & [50, −50]	1
4	[−150, 150] & [150, −150]	30
5	[−90, 90] & [90, −90]	30
6	[−180, 180] & [180, −180]	20

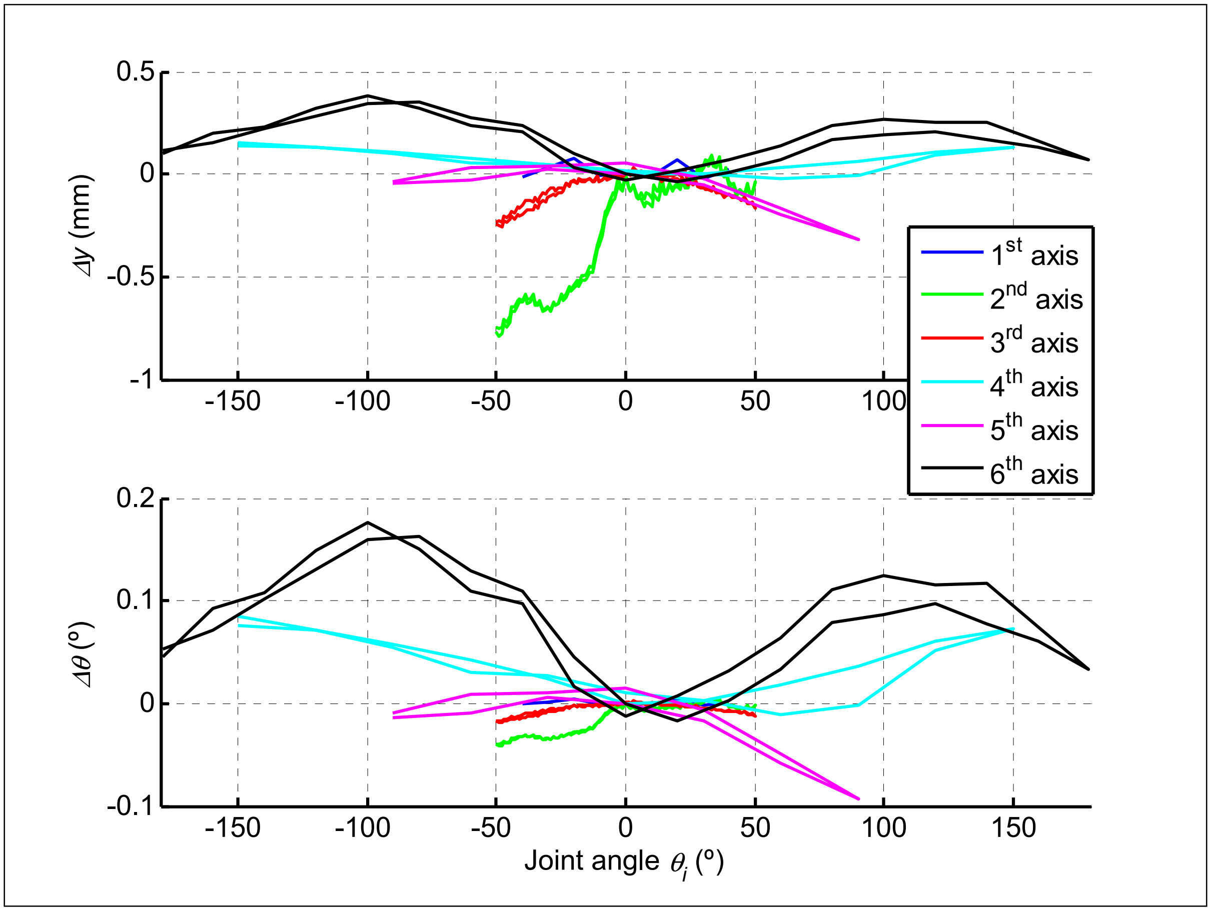


Figure 4.3 Position and angular errors at the TCP when moving each joint one by one

Firstly, it is important to note that the radial and axial errors observed at the TCP were very small (Δx and Δz maximum errors were about 0.050 mm). In other words, the total error presented in Figure 4.3(a) is mainly due to the tangential errors, i.e., is proportional to the angular errors $\Delta\theta_i$.

We can see that joints 2 and 6 lead to large TCP errors. The error for joint 2 may be due to the stiffness effect (as well as error for joint 3, which however seems to be less important), and more specifically to the preloading effect. It is quite possible that the gravity effect on joint 2, when all joints are at zero degrees, is virtually zero (i.e., close to the static equilibrium). Displacing joint 2 from zero degrees, suddenly changes the torsional effect due to the weight of the robot arm. We cannot, however, explain why the error is not similar in both sides of motion. The curve corresponding to axis 2 is not as smooth as the other ones due to noise measurement, since we took a measurement every degree.

The error for joint 6 is probably due to the gear train that runs inside the robot arm.

4.2 Detailed analysis of joint 6

Taking a closer look at the 6th axis, five tests have been performed to study its behavior. The robot has been moved to five different configurations for joints 1 to 5 as shown in Table 4.2.

Table 4.2 Tests for 6th axis analysis

Test	θ_1 (°)	θ_2 (°)	θ_3 (°)	θ_4 (°)	θ_5 (°)	θ_6 range (°)	θ_6 step (°)
1	0	0	0	0	0	[−400; 400] & [400; −400]	20
2	45	−45	45	200	45	[−400; 400] & [400; −400]	20
3	0	45	−45	−100	−45	[−400; 400] & [400; −400]	20
4	20	45	−45	−199	90	[−400; 400] & [400; −400]	20
5	45	−30	20	100	10	[−400; 400] & [400; −400]	5

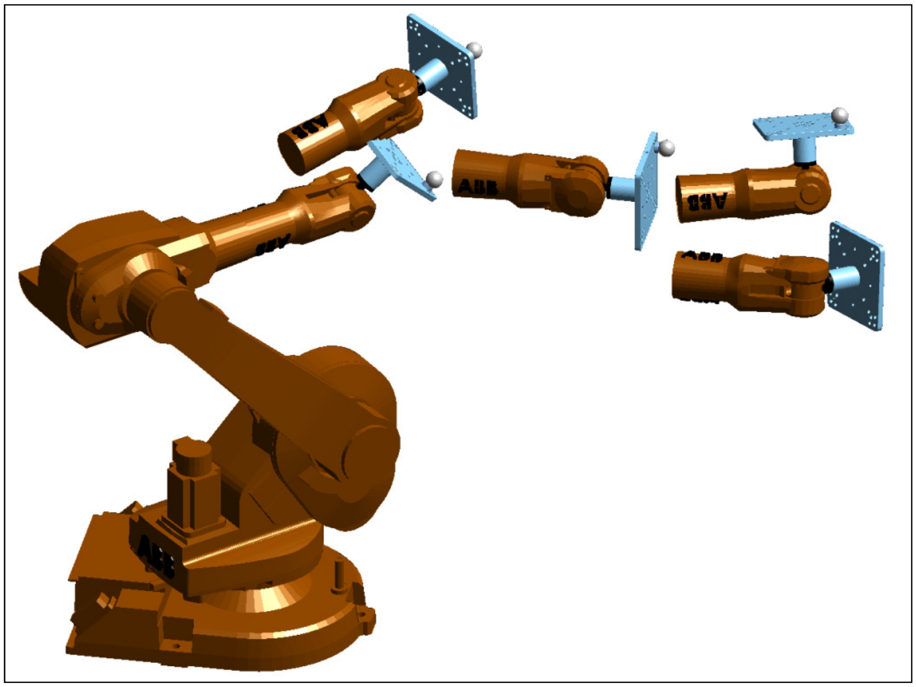


Figure 4.4 Position representation for each of the five tests.

Results for tests 1 to 5 are shown in Figure 4.5.

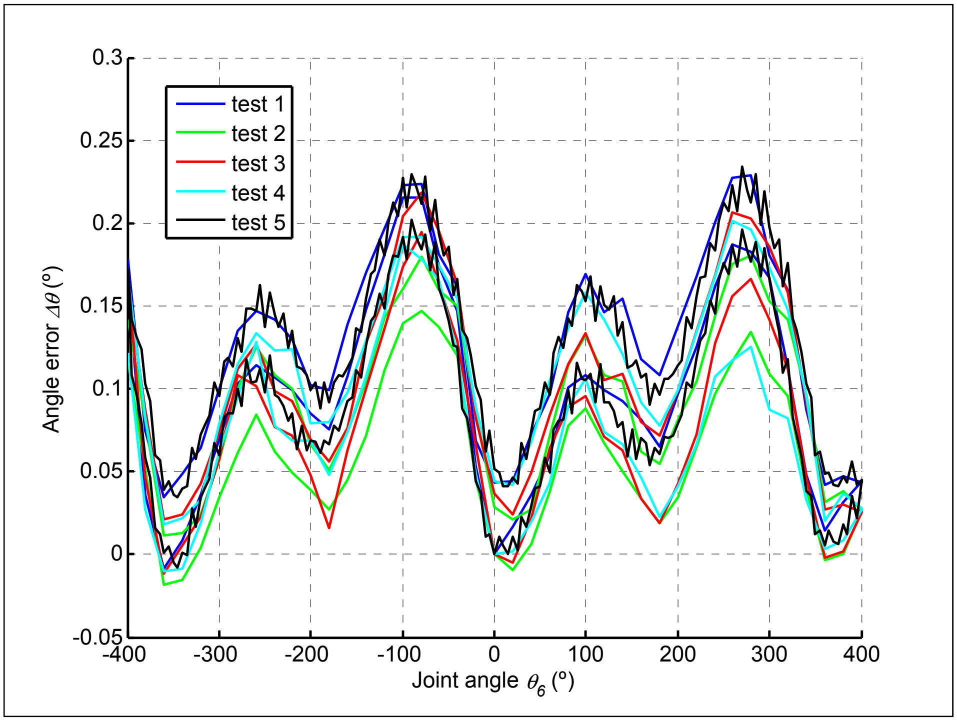


Figure 4.5 Joint 6 angle errors

Firstly, it can be clearly seen that there is a backlash of about 0.05°. Secondly, it can be seen that the shape of the curves in all five tests are nearly identical. Finally, it can be seen that the non-linear shape is cyclic and repeats every full rotation.

4.3 Detailed analysis of the 2nd axis

The second axis behavior is similar to the third axis and it is due to the flexibility of the robot. Because of this flexibility we will end up adding "stiffness parameters", which try to explain this behavior.

4.3.1 Effect of the tool weight

A first series of tests have been performed to evaluate the effect of the tool weight at the end-effector. The robot arm was fully extended and the coordinates of six points along this arm were measured with our laser tracker, as shown in Figure 4.6 and Figure 4.7. Joint 3 is kept constant at −90º and joint 2 varies from −90º to 90º and back from 90º to −90º by steps of 10º.

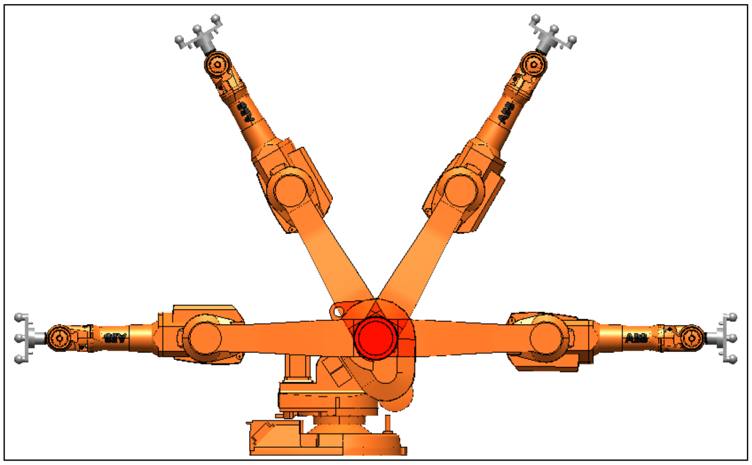


Figure 4.6 Moving axis 2 to evaluate stiffness

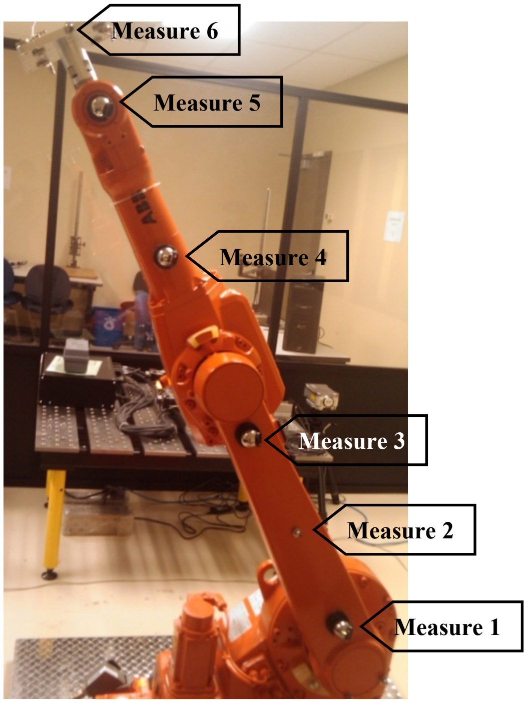

Measure 6
Measure 5
Measure 4
Measure 3
Measure 2
Measure 1

Figure 4.7 Picture of the real measurement points

If the robot links were perfectly stiff, each measurement point would describe a perfect circle as joint 2 rotates. When we fit a circle to each series of points and find the corresponding axis as described in Section 4.1, we obtain the following results:

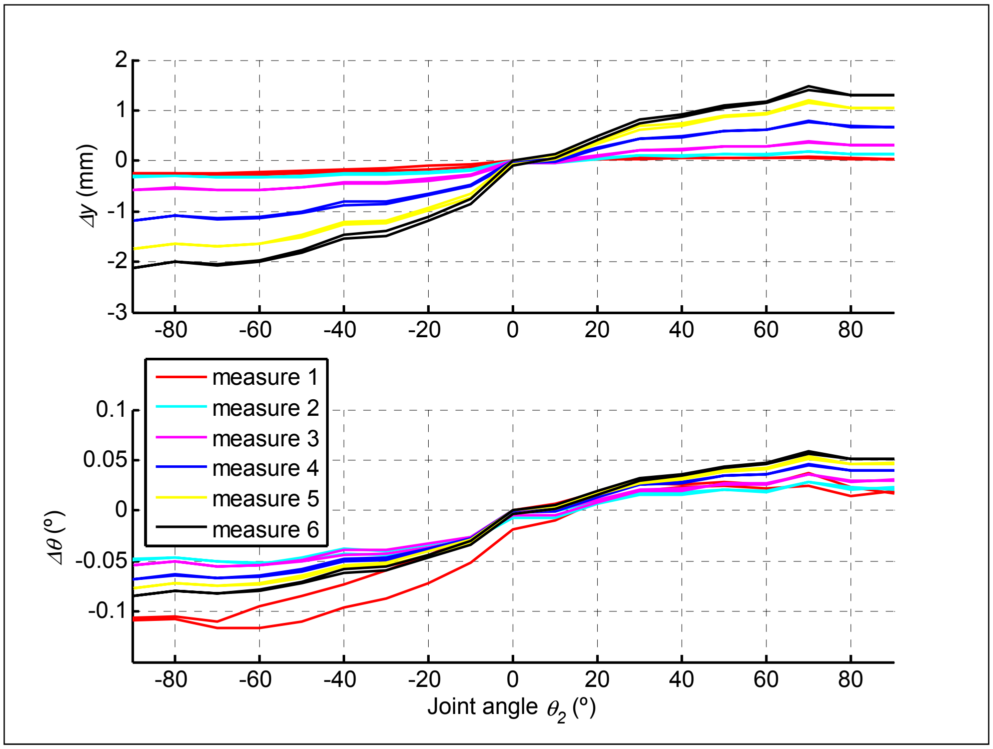


Figure 4.8 Stiffness effect with fully extended arm and a payload of 1.8 kg

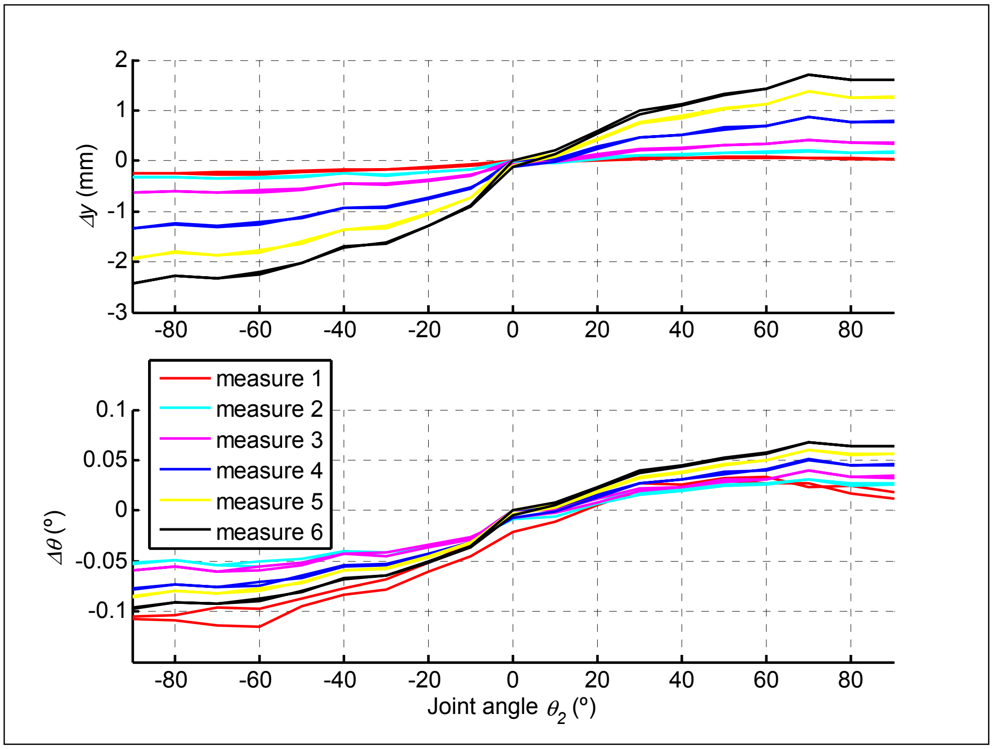


Figure 4.9 Stiffness effect with fully extended arm and a payload of 4.8 kg

Table 4.3 describes the peak effect of adding 3 kg at the end-effector (errors in all directions are taken into account although Δx and Δz are very small compared to Δy). As we can see, the peak displacement is around 0.3 mm. This value is very small compared to the 1.297 mm (and more) that the robot displaces due to its own weight.

Table 4.3 Maximum TCP errors

Angle θ_3 (°)	Max. with 1.8 kg (mm)	Max. with 4.8 kg (mm)	Difference (mm)
−90	2.134	2.426	0.292
90	1.297	1.607	0.310

We can say that the weight added at the end-effector will not be significant compared to the weight of the robot arm (the bodies moved when joint 2 rotates), which weighs approximately 50 kg.

4.3.2 Robot arm stiffness effect

Table 4.4 shows the joints used for other stiffness tests (the step increment is 1°). The tool weight is 1.8 kg.

Table 4.4 Tests for stiffness analysis

test	θ_1 (°)	θ_2 range (°)	θ_3 (°)	θ_4 (°)	θ_5 (°)	θ_6 (°)	θ_2 step (°)
1	0	[−50,50] & [50, −50]	30	0	10	0	1
2	0	[−50,50] & [50, −50]	0	0	10	0	1
3	0	[−50,50] & [50, −50]	−30	0	10	0	1
4	0	[−50,50] & [50, −50]	−60	0	10	0	1
5	0	[−50,50] & [50, −50]	−90	0	10	0	1
6	45	[−50,50] & [50, −50]	0	0	10	0	1

Results for these tests are shown in Figure 4.10. The reference of the zero error has been imposed at the angle of 0°. The error shown is the equivalent angle error for joint 2 (as it cumulates the error for joint 3 also).

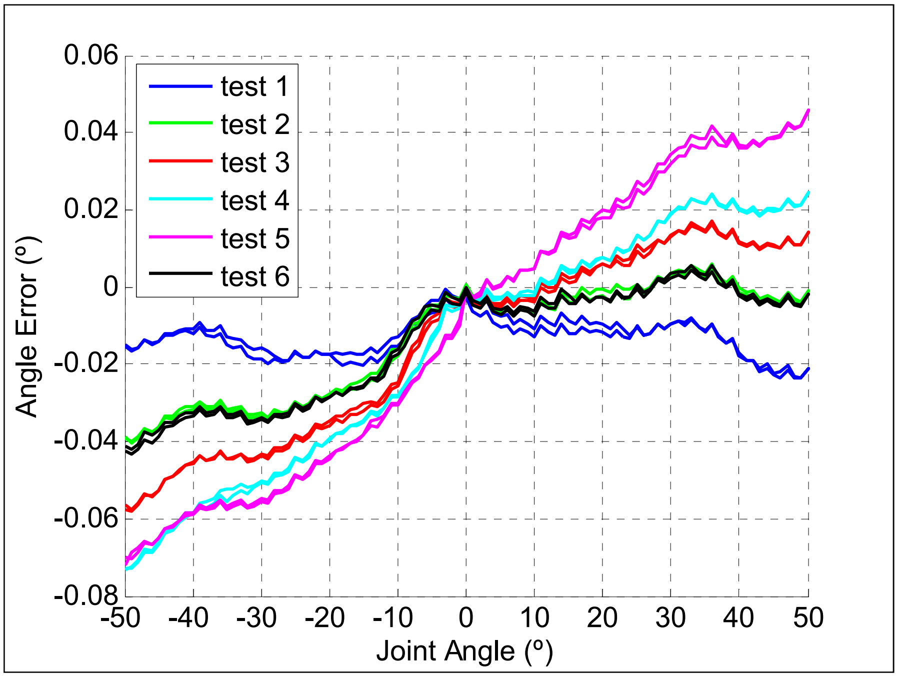


Figure 4.10 Equivalent θ_2 error

CHAPTER 5

ERROR MODELS

An important step of the calibration process consists of choosing the appropriate error model that describes the direct kinematics of our robot. In this section we first explain the DHM representation used and we show the nominal direct kinematics of our robot ABB IRB 1600. Based on the study performed in Chapter 4, we demonstrate that joints 2 and 3 should be modeled as compliant, while a second order Fourier series should be used to model the motion pattern of joint 6. These two particularities are taken into account in a 34-parameter model.

We consider that the nominal kinematic model contains a tool and base calibration. We also show all the five kinematic error models that have been taken into account for our calibration tests, which correspond to 6 error parameters (nominal), 11, 16, 26 and 34 error parameters. Only the last one takes into account the non linear errors from axis 2, 3 and 6.

5.1 Nominal direct kinematics

The used kinematic representation is the one defined by Craig (also known as Denavit Hartenberg Modified (Craig, 1986)), which corresponds to a rotation plus translation through the **x** axis followed by a rotation plus translation through the **z** axis (rotation and translation along the same axis can be interchanged).

$$\mathbf{A}_i^{i-1} = \mathrm{Rot}(\mathbf{x},\alpha_i)\cdot\mathrm{Trans}(a_i,0,0)\cdot\mathrm{Rot}(\mathbf{z},\theta_i)\cdot\mathrm{Trans}(0,0,d_i) \tag{5.1}$$

$$\mathbf{A}_i^{i-1} = \begin{bmatrix} \mathrm{c}(\theta_i) & -\mathrm{s}(\theta_i) & 0 & a_i \\ \mathrm{s}(\theta_i)\mathrm{c}(\alpha_i) & \mathrm{c}(\theta_i)c(\alpha_i) & -\mathrm{s}(\alpha_i) & -d_i\mathrm{s}(\alpha_i) \\ \mathrm{s}(\theta_i)\mathrm{s}(\alpha_i) & \mathrm{c}(\theta_i)s(\alpha_i) & \mathrm{c}(\alpha_i) & d_i\mathrm{c}(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

It will be useful to know the original DHM parameters of a generic $\mathbf{A}_i^{i-1}$ transformation. The geometrical transformation must correspond to a DHM transformation:

$$a_i = \mathbf{A}_i^{i-1}{}_{1,4} \tag{5.3}$$

$$\alpha_i = \operatorname{atan2}\left(-\mathbf{A}_i^{i-1}{}_{2,3}, \mathbf{A}_i^{i-1}{}_{3,3}\right) \tag{5.4}$$

$$\theta_i = \operatorname{atan2}\left(-\mathbf{A}_i^{i-1}{}_{1,2}, \mathbf{A}_i^{i-1}{}_{1,1}\right) \tag{5.5}$$

$$\operatorname{abs}\left(d_i\right) = \sqrt{(\mathbf{A}_i^{i-1}{}_{2,4})^2 + (\mathbf{A}_i^{i-1}{}_{3,4})^2} \tag{5.6}$$

$$\operatorname{sgn}\left(d_i\right) = \operatorname{sgn}(\mathbf{A}_i^{i-1}{}_{3,4} c\left(\alpha_i\right)) \tag{5.7}$$

Table 5.1 corresponds to the DHM parameters according to the frames placed in Figure 5.1.

Table 5.1 Nominal DHM parameters for the ABB IRB 1600 robot

i	α_i (º)	a_i (mm)	θ_i (º)	d_i (mm)
1	0	0	θ_1	486.5
2	−90	150	θ_2−90	0
3	0	700	θ_3	0
4	−90	0	θ_4	600
5	90	0	θ_5	0
6	−90	0	θ_6+180	65

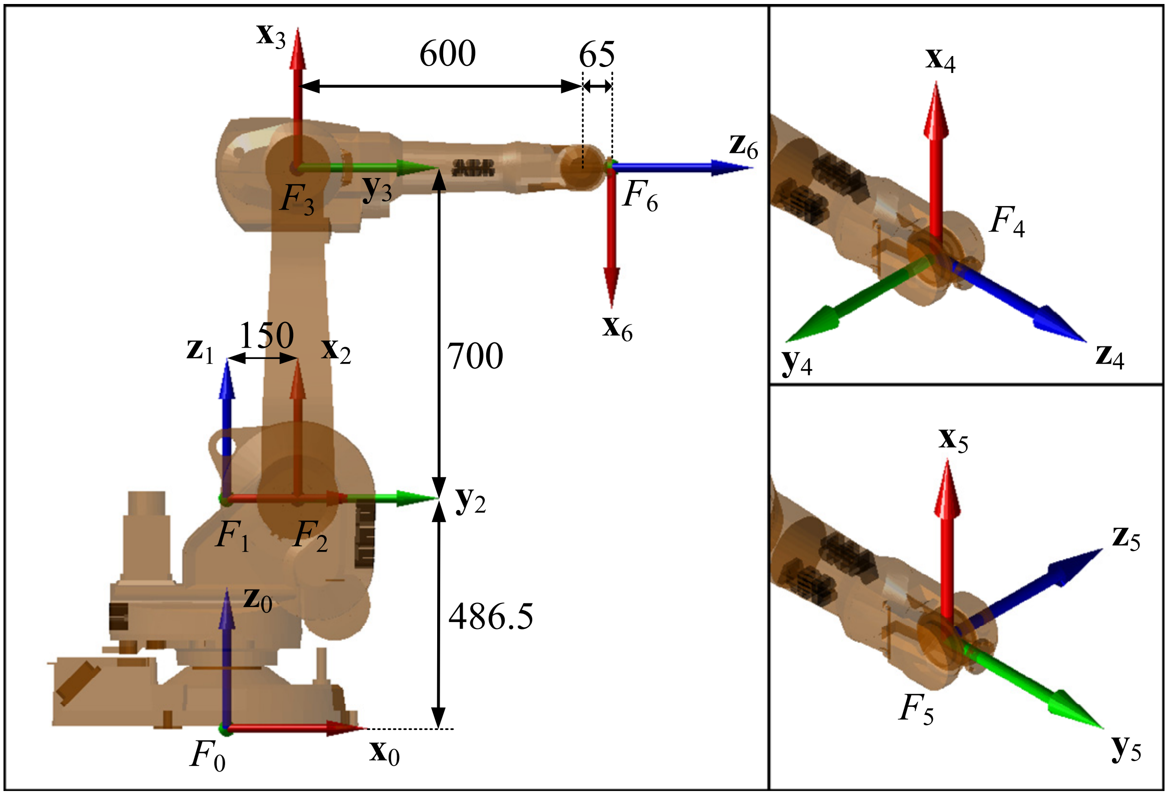

x3
600
65
z6
F3
y3
F6
x6
z1
150
x2
700
y2
F1
F2
z0
486.5
F0
x0
x4
F4
y4
z4
x5
z5
F5
y5

Figure 5.1 Frames corresponding to the DHM notation for the ABB IRB 1600 robot

5.2 Axis 6 model

Using the information found from the tests performed in Section 4.2 we tried to fit these data into a second order Fourier series. We used the results obtained from the fifth test represented in Figure 4.5 (we used the two way curves) to find the best fit of a Fourier series function. Using the "Curve Fitting Tool" from Matlab we obtain the fit shown in Figure 5.2. We used a Fourier function because the error is cyclic and the pattern is repeated every 360 degres.

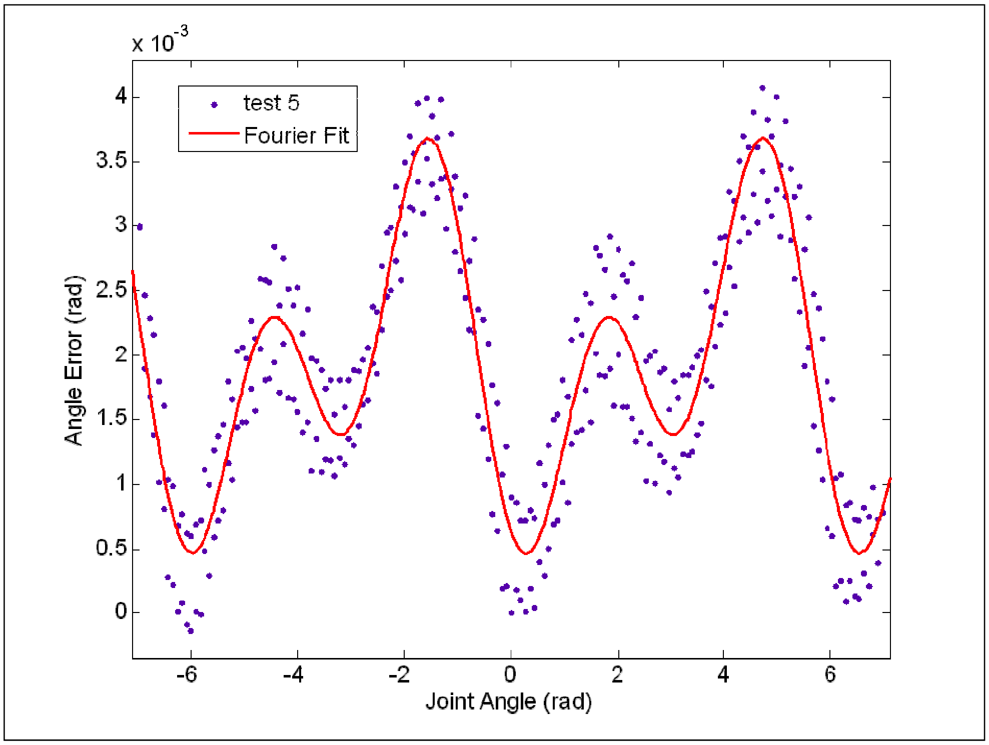


Figure 5.2 Fourier series fit for the nonlinear motion pattern of joint 6

The equation for this second order Fourier fit is:

$$F(x) = a_0 + a_1 \cos(\omega x) + b_1 \sin(\omega x) + a_2 \cos(2\omega x) + b_2 \sin(2\omega x) \qquad (5.8)$$

The obtained coefficients are shown in Table 3.3.

Table 5.2 Coefficients of the Fourier fit

Parameter	Value
a_0	0.00198
a_1	−0.000373
b_1	−0.000750
a_2	−0.000956
b_2	−0.000244
ω	1.001

It may be interesting to see that ω is very close to 1. We will arbitrarily impose that ω is constant and equal to 1. The parameter a_0 is already considered in our kinematic calibration.

5.3 Stiffness model

A model that has 4 error parameters is used to fit the stiffness behavior of the serial robot. This method only takes into account the deformation of the arm (joints 2 and 3). These 4 parameter correspond to the maximum deflection when the arm is extended, the center of gravity of joint 2 and 3 and the relation between masses of joints 2 and 3.

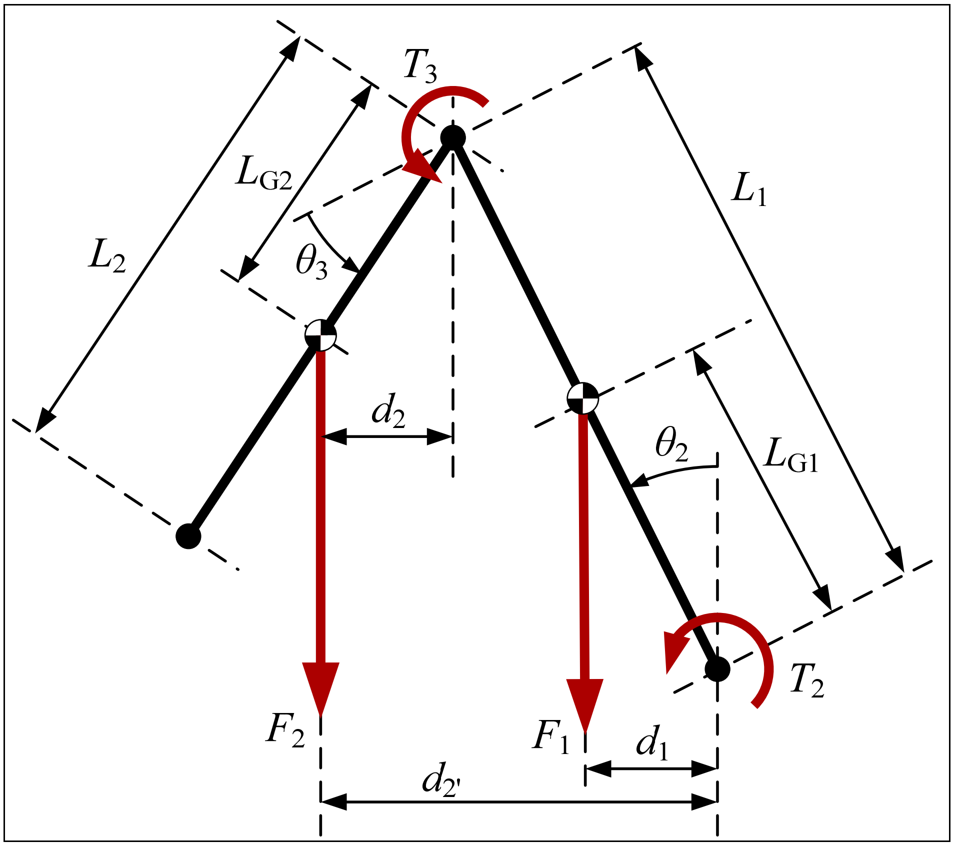


Figure 5.3 Robot arm representation

We can directly say from Figure 5.3:

$$d_1 = L_{G1}\sin(\theta_2) \tag{5.9}$$

$$d_2 = L_{G2}\sin\left(\theta_2 + \theta_3 + \frac{\pi}{2}\right) \tag{5.10}$$

$$d_2' = L_1 \sin(\theta_2) + d_2 \tag{5.11}$$

According to Figure 5.3, the torque of joints 2 and 3 that are applied to the robot are:

$$T_2 = F_1 d_1 + F_2 d_2' \tag{5.12}$$

$$T_3 = F_2 d_2 \tag{5.13}$$

Using a new parameter m as the relationship between the two masses ($m=F_1/F_2$) and considering that the variation of angle is proportional (with constant k) to the torque supported by the robot at each joint we can represent next equations:

$$\delta\theta_2 = kT_2 = kF_1\left(d_1 + md_2'\right) \tag{5.14}$$

$$\delta\theta_3 = kT_3 = mkF_1 d_2 \tag{5.15}$$

If we consider the worst case of stiffness (where torque is maximum for joints 2 and 3) we can guess the product kF_1 if we know the displacement of the end effector for this position.

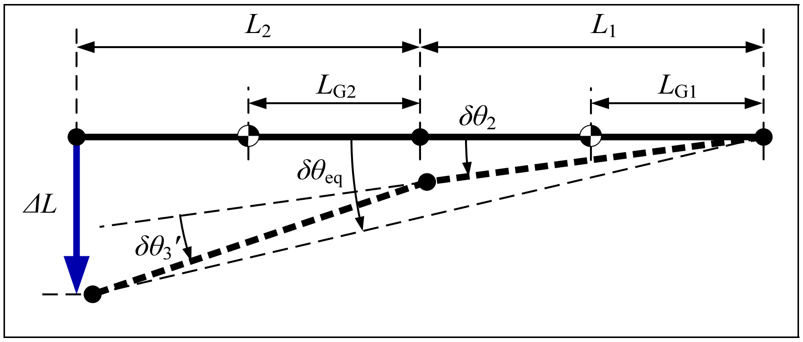


Figure 5.4 Estimating ΔL

ΔL is the displacement of the end-effector on the case limit (joint 2 is 90° and joint 3 is −90°). If errors are small we can write the next equivalences (since $\sin(x) \cong x \cong \sin^{-1}(x)$ when x is small):

$$\Delta L \cong \left(L_1 + L_2\right)\sin(\delta\theta_{\text{eq}}) \tag{5.16}$$

Using $\delta\theta_3$’ as a function of $\delta\theta_3$ and using equations (5.14) and (5.15) (we took into account the case where t_2=90 and e_3=−90), we obtain:

$$\delta\theta_{\text{eq}} \cong \delta\theta_2 + \sin^{-1}\left[\frac{L_2\sin\left(\delta\theta_3\right)}{L_1 + L_2}\right] \cong kF_1\left[L_{\text{G1}} + m\left(L_1 + L_{\text{G2}}\right) + m\frac{L_2}{L_1 + L_2}L_{\text{G2}}\right] \tag{5.17}$$

Finally, combining equations (5.16) and (5.17) we can find kF_1 as a function of ΔL:

$$kF_1 \cong \frac{\sin^{-1}\left(\dfrac{\Delta L}{L_1 + L_2}\right)}{L_{\text{G1}} + m\left(L_1 + L_{\text{G2}}\right) + m\dfrac{L_2}{L_1 + L_2}L_{\text{G2}}} \tag{5.18}$$

In this way, the variables L_{G1}, L_{G2}, m and ΔL, define the 4 error parameters of the stiffness model. L_1 and L_2 are known as they are the arm lengths. A good guess of these parameters could be:

$$m \approx 1 \tag{5.19}$$

$$L_{\text{G1}} \approx \frac{L_1}{2} \tag{5.20}$$

$$L_{\text{G2}} \approx \frac{L_2}{2} \tag{5.21}$$

$$\Delta L \approx 1.5 \tag{5.22}$$

ΔL has been established according to the results on Table 4.3. With a simulation on Matlab we can predict the behavior of the equivalent θ_2 error (θ_{eq}) used in (5.19) as shown in Figure 5.5:

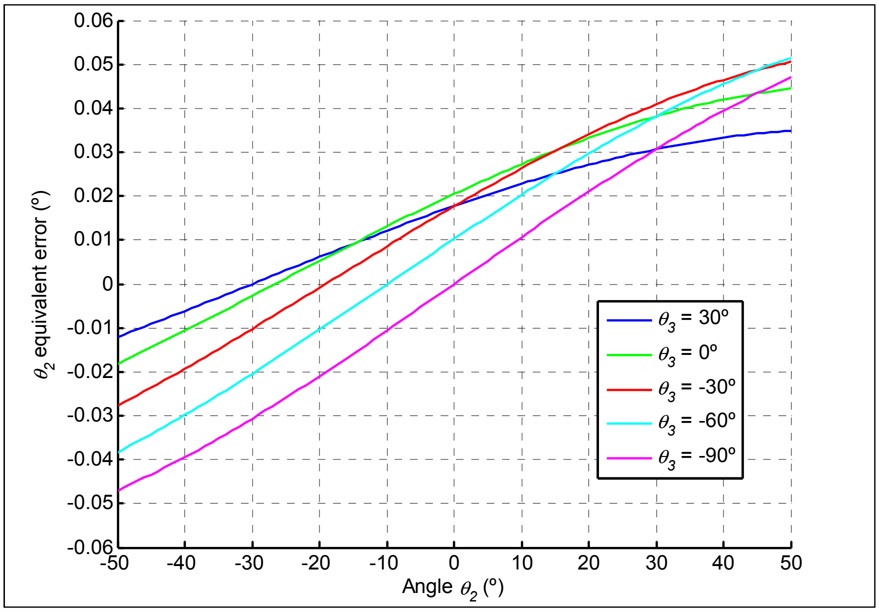


Figure 5.5 Equivalent θ_2 error

We can make other guesses so that this figure looks similar to Figure 4.10. For example, randomly trying some combinations, we obtained a good approximation of the real test using the parameters ΔL=1.8, L_{G1}=300, L_{G2}=900 and m=2. We also forced the zero reference error at 0º. We obtain the results shown in Figure 5.6:

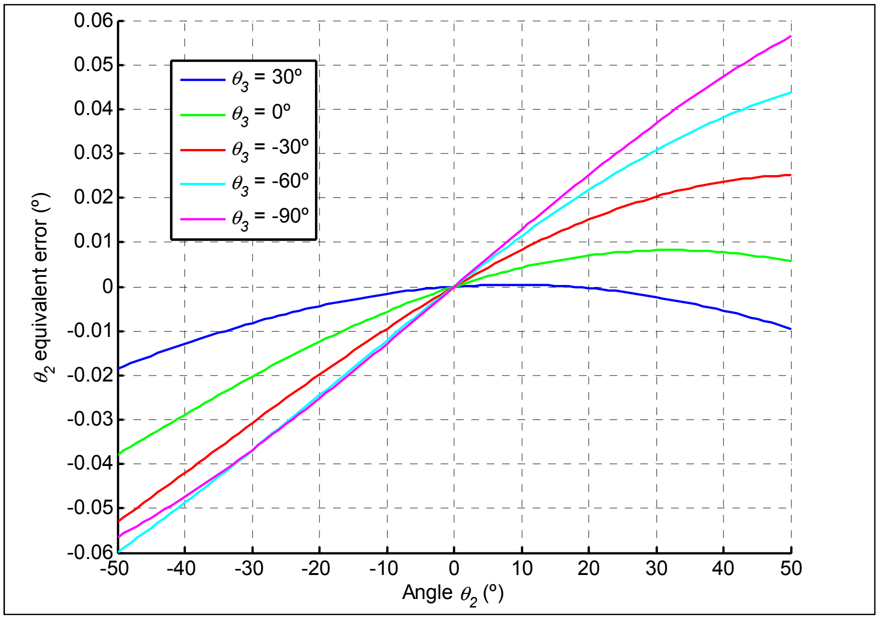


Figure 5.6 Equivalent θ_2 error for a different combination of parameters

Although it may not have as much physical sense as the first parameters, this last combination may give better results.

5.4 Direct kinematic models

Five different direct kinematic models have been tested to our ABB IRB 1600. The first one is just a calibration of the robot base and tool (considered as the nominal kinematics). The second one takes into account the joint calibration (level-1 calibration). The third one considers joint calibration plus an adjustment of some distances. The fourth one is an entire kinematic calibration (level-2). Finally, the fifth calibration method takes into account a complete kinematic calibration plus an adjustment of stiffness and 6th axis errors.

5.4.1 Nominal model

Only the robot base and the tool are calibrated. This model adds no error parameters to the robot. Actually there are 6 error parameters that determine the base plus 3 parameters positioning each tool target (considering that we do not measure orientation). Inverse kinematics can be directly obtained with the algebraic algorithm.

Table 5.3 Nominal robot model

i	α_i (º)	a_i (mm)	θ_i (º)	d_i (mm)
1	0	0	θ_1	486.5
2	−90	150	$\theta_2 - 90$	0
3	0	700	θ_3	0
4	−90	0	θ_4	600
5	90	0	θ_5	0
6	−90	0	$\theta_6 + 180$	65

5.4.2 11-parameter model

Robot base, tool and offset angles of the motors are calibrated. This model is the same as the first one but it adds 5 error parameter to the robot calibration (the six offsets of the 6 motors).

The offset of the first motor is expressed as a robot base rotation, so actually it is already considered in the first calibration (nominal). Inverse kinematics can be directly obtained with the algebraic algorithm. This model represents a level-1 calibration.

Table 5.4 Robot model for 11 parameters

i	α_i (°)	a_i (mm)	θ_i (°)	d_i (mm)
1	0	0	θ_1	486.5
2	−90	150	$\theta_2-90+\delta\theta_2$	0
3	0	700	$\theta_3+\delta\theta_3$	0
4	−90	0	$\theta_4+\delta\theta_4$	600
5	90	0	$\theta_5+\delta\theta_5$	0
6	−90	0	$\theta_6+180+\delta\theta_6$	65

5.4.3 16-parameter model

Robot base, tool, offset angles and some distances are calibrated. This method is the same as the second one but five geometrical distances are added as error parameters. Those distances are chosen so that inverse kinematics can be directly obtained with the algebraic algorithm. Those distances are a_2, a_3, a_4, d_4 and d_6.

Table 5.5 Robot model for 16 parameters

i	α_i (°)	a_i (mm)	θ_i (°)	d_i (mm)
1	0	0	θ_1	486.5
2	−90	$150+\delta a_2$	$\theta_2-90+\delta\theta_2$	0
3	0	$700+\delta a_3$	$\theta_3+\delta\theta_3$	0
4	−90	δa_4	$\theta_4+\delta\theta_4$	$600+\delta d_4$
5	90	0	$\theta_5+\delta\theta_5$	0
6	−90	0	$\theta_6+180+\delta\theta_6$	$65+\delta d_6$

5.4.4 Kinematic calibration

This model corresponds to a level-2 calibration of the robot (26 parameters). Robot base and tool calibration are also taken into account. We often used the full kinematic model to compare results. The error parameters for the geometrical models are directly obtained from the axis calibration explained in Section 3.1. Inverse kinematics must be calculated by a numerical and iterative method (three iterations are enough in most cases). This model will be referred to as a full kinematic model (or entire kinematic model), which is the result of an entire kinematic calibration.

Table 5.6 Full robot kinematic model

i	α_i (°)	a_i (mm)	θ_i (°)	d_i (mm)
1	0	0	θ_1	486.5
2	$-90+\delta\alpha_2$	$150+\delta a_2$	$\theta_2-90+\delta\theta_2$	δd_2
3	$\delta\alpha_3$	$700+\delta a_3$	$\theta_3+\delta\theta_3$	δd_3
4	$-90+\delta\alpha_4$	δa_4	$\theta_4+\delta\theta_4$	$600+\delta d_4$
5	$90+\delta\alpha_5$	δa_5	$\theta_5+\delta\theta_5$	δd_5
6	$-90+\delta\alpha_6$	δa_6	$\theta_6+180+\delta\theta_6$	$65+\delta d_6$

5.4.5 34-parameter model

This model corresponds to a complete kinematic calibration of the robot (26 error parameters). It also takes into account the stiffness effect of the robot arm with 4 more parameters for joints 2 and 3 (as explained in Section 5.3). In addition it reduces the errors of 6th axis with 4 extra error parameters. Inverse kinematics must be calculated by a numerical and iterative method (three iterations are enough in most cases). This model corresponds to a level-3 calibration.

Table 5.7 Robot model for 34 parameters

i	α_i (°)	a_i (mm)	θ_i (°)	d_i (mm)
1	0	0	θ_1	486.5
2	$-90+\delta\alpha_2$	$150+\delta a_2$	$\theta_2-90+\delta f\theta_2(\theta_2,\theta_3,\delta\mathbf{p}_{23})$	δd_2
3	$\delta\alpha_3$	$700+\delta a_3$	$\theta_3+\delta f\theta_3(\theta_2,\theta_3,\delta\mathbf{p}_{23})$	δd_3
4	$-90+\delta\alpha_4$	δa_4	$\theta_4+\delta\theta_4$	$600+\delta d_4$
5	$90+\delta\alpha_5$	δa_5	$\theta_5+\delta\theta_5$	δd_5
6	$-90+ \delta\alpha_6$	δa_6	$\theta_6+180+\delta f\theta_6(\theta_6,\delta\mathbf{p}_6)$	$65+\delta d_6$

The 4 parameters concerning the 6th axis errors correspond to the parameters represented in Table 5.2. We considered that the parameters that correspond to the joint 6 model are $\delta\mathbf{p}_6=\{a_0, a_1, b_1, a_2, b_2\}$ The parameter ω is arbitrarily forced to 1 and the parameter a_0 was already taken into account in the previous models as $\delta\theta_6$.

The 4 parameters concerning the stiffness are inserted as $\delta f\theta_2(\theta_2,\theta_3,\delta\mathbf{p}_{23})$ and $\delta f\theta_3(\theta_2,\theta_3,\delta\mathbf{p}_{23})$ because they are a function that depends on the arm position (θ_2 and θ_3) and the four stiffness error parameters: $\delta\mathbf{p}_{23}=\{\Delta L, L_{G1}, L_{G2}, m\}$ as explained in Section 5.3.

CHAPTER 6

INVERSE KINEMATICS

The computation of the inverse kinematics is useful once we have obtained a calibrated model of our robot. To validate the model that we found we need to change our real targets into fake targets using the inverse kinematics, so that the position of the robot is improved. In other words, we need to change the joints of a desired position into a different (but close) set of joints so that error for that position is improved.

The nominal inverse kinematics may be analytical, but to compute the inverse kinematics of a calibrated robot we may need a numerical method. The method used to compute the improved inverse kinematics is iterative and it approaches to the solution at each iteration. Joints are called as $\mathbf{q} = [q_1, q_2, q_3, q_4, q_5, q_6]^T$.

6.1 Nominal inverse kinematics

The nominal inverse kinematics for the ABB IRB 1600-6/1.45, as well as many other industrial robots, is analytical.

6.1.1 Finding q_1, q_2 and q_3

Taking into account that the direct kinematics is represented as follows:

$$\mathbf{X} = \mathbf{T}_6^0 = \mathbf{A}_1^0\mathbf{A}_2^1\mathbf{A}_3^2\mathbf{A}_4^3\mathbf{A}_5^4\mathbf{A}_6^5 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.1}$$

We can find the position of the origin of the 4th frame using one of the two next equations:

$$\mathbf{T}_4^0 = \mathbf{A}_1^0\mathbf{A}_2^1\mathbf{A}_3^2\mathbf{A}_4^3 = \mathbf{X}(\mathbf{A}_5^4\mathbf{A}_6^5)^{-1} = \begin{bmatrix} & \mathbf{R}_4^0 & & \mathbf{p}_{4,\text{origin}}^0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.2}$$

$$\mathbf{p}_{4,\text{origin}}^0 = \mathbf{A}_1^0\mathbf{A}_2^1\mathbf{A}_3^2\mathbf{A}_4^3\begin{bmatrix}0\\0\\0\\1\end{bmatrix} = \mathbf{X}\begin{bmatrix}0\\0\\-d_6\\1\end{bmatrix} \tag{6.3}$$

We can find q_1, q_2 and q_3 from:

$$\mathbf{p}_{4,\text{origin}}^0 = \begin{bmatrix}p_{x_4}\\p_{y_4}\\p_{z_4}\end{bmatrix} = \begin{bmatrix}\mathrm{c}_1(a_2+a_3\mathrm{s}_2+d_4\mathrm{c}_{23}+a_4\mathrm{s}_{23})\\ \mathrm{s}_1(a_2+a_3\mathrm{s}_2+d_4\mathrm{c}_{23}+a_4\mathrm{s}_{23})\\ d_1+a_3\mathrm{c}_2-d_4\mathrm{s}_{23}+a_4\mathrm{c}_{23}\end{bmatrix} = \begin{bmatrix}p_x-d_6a_x\\p_y-d_6a_y\\p_z-d_6a_z\end{bmatrix} \tag{6.4}$$

Combining the two first equations from (6.4), solution for q_1 is direct considering that the term $a_2+a_3\mathrm{s}_2+d_4\mathrm{c}_{23}+a_4\mathrm{s}_{23}$ is different from zero:

$$\frac{\mathrm{c}_1}{p_x-d_6a_x} = \frac{\mathrm{s}_1}{p_y-d_6a_y} \tag{6.5}$$

There are two solutions for q_1:

$$q_{1,1} = \operatorname{atan2}(p_y-d_6a_y,\, p_x-d_6a_x) \tag{6.6}$$

$$q_{1,2} = \operatorname{atan2}(-p_y+d_6a_y,\, -p_x+d_6a_x) \tag{6.7}$$

We find here the so-called alignment singularity when $p_y=d_6a_y$ and $p_x=d_6a_x$, which means that the term $a_2+a_3\mathrm{s}_2+d_4\mathrm{c}_{23}+a_4\mathrm{s}_{23}$ is zero, so this singularity is the only case where we cannot apply the equation (6.5).

If q_1 is known, solution for q_3 can be obtained from next two equations:

$$\mathrm{c}_1p_{x_4}+\mathrm{s}_1p_{y_4}-a_2 = a_3\mathrm{s}_2+d_4\mathrm{c}_{23}+a_4\mathrm{s}_{23} \tag{6.8}$$

$$a_3\mathrm{c}_2-d_4\mathrm{s}_{23}+a_4\mathrm{c}_{23} = p_{z_4}-d_1 \tag{6.9}$$

Squaring the above two equations, adding them, and simplifying we have that

$$a_3^2 + d_4^2 + a_4^2 - 2a_3d_4s_3 + 2a_4a_3\mathrm{c}_3 = \left(p_{z_4} - d_1\right)^2 + \left(\mathrm{c}_1 p_{x_4} + \mathrm{s}_1 p_{y_4} - a_2\right)^2 \tag{6.10}$$

So c_3 and s_3 are linearly dependent

$$\mathrm{c}_3 = \frac{A + B\mathrm{s}_3}{C} \tag{6.11}$$

Where C is a constant which can be zero or different to zero:

$$A = \left(p_{z_4} - d_1\right)^2 + \left(c_1 p_{x_4} + s_1 p_{y_4} - a_2\right)^2 - a_3^2 - d_4^2 - a_4^2 \tag{6.12}$$

$$B = 2a_3d_4 \tag{6.13}$$

$$C = 2a_4a_3 \tag{6.14}$$

6.1.2 Case with *C*=0

If C=0 it means that a_4 is zero (a_3 could also be zero but in this case it would not be useful to have the 3rd axis). That is the case of robot ABB IRB 1600-6/1.45, s_3 can be directly obtained (B is a constant different to 0):

$$\mathrm{s}_3 = -\frac{A}{B} \tag{6.15}$$

There are two solutions for q_3:

$$q_{3,1} = \mathrm{atan2}(\mathrm{s}_3, \sqrt{1 - \mathrm{s}_3^2}) \tag{6.16}$$

$$q_{3,2} = \mathrm{atan2}(\mathrm{s}_3, -\sqrt{1 - \mathrm{s}_3^2}) \tag{6.17}$$

6.1.3 Case with *C*≠0

In the case where $C \neq 0$ there is the line (6.11) intersecting the circle $\mathrm{s}_3^2 + \mathrm{c}_3^2 = 1$ (working in the plane **xy**, where **x**=s_3 and **y**=c_3).

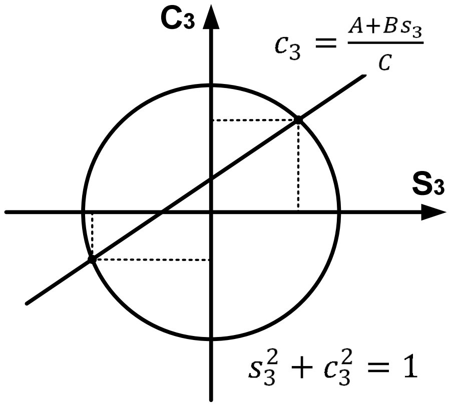


Figure 6.1 Solutions for q_3

Two solutions can be found:

$$s_3^2 + \left(\frac{A+Bs_3}{C}\right)^2 = 1 \tag{6.18}$$

$$\left(1+\frac{B^2}{C^2}\right)s_3^2 + \frac{2AB}{C^2}s_3 + \frac{A^2}{C^2} - 1 = 0 \tag{6.19}$$

$$s_{3_{1,2}} = \frac{-\frac{AB}{C^2} \pm \sqrt{\frac{A^2B^2}{C^4} + \left(1+\frac{B^2}{C^2}\right)\left(1-\frac{A^2}{C^2}\right)}}{1+\frac{B^2}{C^2}} \tag{6.20}$$

In the case where the absolute value of s_3 is bigger than one, the pose is not reachable because it is too far from the robot. In the case where the value of s_3 is exactly one, we find the so-called elbow singularity.

Once we obtained s_3:

$$q_{3,1} = \operatorname{atan2}\left(s_{3,1}, \frac{A + Bs_{3,1}}{C} \right) \tag{6.21}$$

$$q_{3,2} = \operatorname{atan2}\left(s_{3,2}, \frac{A + Bs_{3,2}}{C} \right) \tag{6.22}$$

Finally, q_2 can be obtained from (6.8) and (6.9). Rewriting these equations:

$s_2(6.8) + c_2(6.9)$: $$a_3 - d_4 s_3 + a_4 c_3 = k_1 s_2 + k_2 c_2 \tag{6.23}$$

$c_2(6.8) - s_2(6.9)$: $$d_4 c_3 + a_4 s_3 = c_2 k_1 - s_2 k_2 \tag{6.24}$$

Where:

$$k_1 = c_1 p_{x_4} + s_1 p_{y_4} - a_2 \tag{6.25}$$

$$k_2 = p_{z_4} - d_1 \tag{6.26}$$

Joint q_2 is completely defined as we have 2 lineal equations (q_3 is known):

$$\begin{bmatrix} k_1 & k_2 \\ -k_2 & k_1 \end{bmatrix} \begin{bmatrix} s_2 \\ c_2 \end{bmatrix} = \begin{bmatrix} a_3 - d_4 s_3 + a_4 c_3 \\ d_4 c_3 + a_4 s_3 \end{bmatrix} \tag{6.27}$$

$$q_2 = \operatorname{atan2}(s_2, c_2) \tag{6.28}$$

6.1.4 Finding q_4, q_5 and q_6

To find last three joints we must use $\mathbf{R}_6^3$:

$$\mathbf{T}_6^3 = \left(\mathbf{A}_1^0 \mathbf{A}_2^1 \mathbf{A}_3^2 \right)^{-1} \mathbf{X} = \mathbf{A}_4^3 \mathbf{A}_5^4 \mathbf{A}_6^5 = \begin{bmatrix} & \mathbf{R}_6^3 & & \mathbf{p}_{6,\text{origin}}^3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.29}$$

where

$$\mathbf{R}_6^3 = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{bmatrix} = \begin{bmatrix} * & * & -\mathrm{c}_4\mathrm{s}_5 \\ -\mathrm{s}_5\mathrm{c}_6 & \mathrm{s}_5\mathrm{s}_6 & \mathrm{c}_5 \\ * & * & \mathrm{s}_4\mathrm{s}_5 \end{bmatrix} \tag{6.30}$$

We omitted the expressions marked with * as they are not of interest. We obtain two solutions for q_5:

$$q_{5,1} = \mathrm{atan2}(\sqrt{1-r_{2,3}^2}, r_{2,3}) \tag{6.31}$$

$$q_{5,2} = \mathrm{atan2}(-\sqrt{1-r_{2,3}^2}, r_{2,3}) \tag{6.32}$$

Once q_5 is found, q_4 and q_6 are completely defined if $q_5 \neq 0$:

$$q_4 = \mathrm{atan2}(\frac{r_{3,3}}{\mathrm{s}_5}, \frac{-r_{1,3}}{\mathrm{s}_5}) \tag{6.33}$$

$$q_6 = \mathrm{atan2}(\frac{r_{2,2}}{\mathrm{s}_5}, \frac{-r_{2,1}}{\mathrm{s}_5}) \tag{6.34}$$

If we are always working in the range of [−180, 180] there are 2×2×1×2×1×1=8 solutions. If this range is extended there can be more solutions.

6.1.5 Wrist singularity (q_5=0)

We will often find the wrist singularity (where q_5=0, here there is a possible solution simplifying (6.30)):

$$\mathbf{R}_6^3 = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{bmatrix} = \begin{bmatrix} -\mathrm{c}_4\mathrm{c}_6 + \mathrm{s}_4\mathrm{s}_6 & * & 0 \\ 0 & 0 & 1 \\ \mathrm{c}_4\mathrm{s}_6 + \mathrm{s}_4\mathrm{c}_6 & * & 0 \end{bmatrix} \tag{6.35}$$

We can completely find q_6 if q_4 is known (we have 1 DOF so we can impose q_4=0)

$$\mathrm{s}_4 r_{1,1} + \mathrm{c}_4 r_{3,1} = \mathrm{s}_4^2 \mathrm{s}_6 + \mathrm{c}_4^2 \mathrm{s}_6 = \mathrm{s}_6 \quad (6.36)$$

$$\mathrm{s}_4 r_{3,1} - \mathrm{c}_4 r_{1,1} = \mathrm{c}_6 \quad (6.37)$$

$$q_6 = \operatorname{atan2}(\mathrm{s}_6, \mathrm{c}_6) \quad (6.38)$$

6.2 Iterative inverse calculation

The iterative inverse calculation is useful when the nominal inverse kinematics cannot be applied. Here we explain a numeric and iterative method that it approaches to the final solution at each iteration. This method does not need the computation of the jacobian nor derivatives of any kind. It only needs the complete direct model and the nominal inverse kinematics solution. We used this inverse instead of other existing ones (such as (Chen and Parker, 1994)) because we experimentally found that we obtained good results.

The expression $\mathbf{T}_6$ is defined as the nominal direct kinematics:

$$\mathbf{T}_6 = \mathbf{A}_1^0 \mathbf{A}_2^1 \mathbf{A}_3^2 \mathbf{A}_4^3 \mathbf{A}_5^4 \mathbf{A}_6^5 \quad (6.39)$$

Taking $\mathbf{T}_{\mathrm{R6}}$ as the “real” direct model (the model that takes into account all errors and is closer to the real robot):

$$\mathbf{T}_{\mathrm{R6}} = \mathbf{A}_{R,1}^{\,0} \mathbf{A}_{R,2}^{\,1} \mathbf{A}_{R,3}^{\,2} \mathbf{A}_{R,4}^{\,3} \mathbf{A}_{R,5}^{\,4} \mathbf{A}_{R,6}^{\,5} \quad (6.40)$$

If we want to find the inverse solution for the pose **H** which represents the 6th frame with respect to the robot base frame “0”, we must proceed with next steps for iteration *i* from 1 to *n* (we take $\mathbf{H}_{\mathrm{fake},0}$=**H** and $\mathbf{H}_{\mathrm{aprox},0}$=**H** so that at first iteration $\mathbf{H}_{\mathrm{fake},1}$=**H**):

1. $\mathbf{H}_{\mathrm{fake},i} = \mathbf{H}_{\mathrm{fake},i-1} (\mathbf{H}_{\mathrm{aprox},i-1})^{-1} \mathbf{H}$.
2. $\boldsymbol{q}_{\mathrm{aprox},i}$ is the nominal inverse kinematics solution for $\mathbf{H}_{\mathrm{fake},i}$.
3. We take $\mathbf{H}_{\mathrm{aprox},i} = \mathbf{T}_{\boldsymbol{R}6}(\boldsymbol{q}_{\mathrm{aprox},i})$.
4. If the error between $\mathbf{H}_{\mathrm{aprox},i}$ and **H** is smaller than the desired error we take $\boldsymbol{q}_{\mathrm{aprox},i}$ as the real inverse kinematics solution for **H**, otherwise we keep iterating.

The block diagram of this algorithm is shown in Figure 6.2.

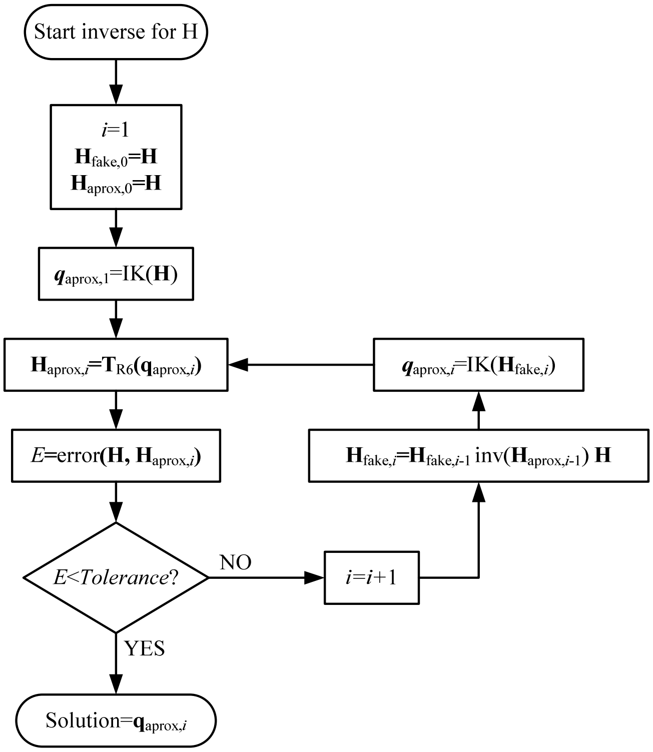

Start inverse for H
i=1
$\mathbf{H}_{\text{fake},0}=\mathbf{H}$
$\mathbf{H}_{\text{aprox},0}=\mathbf{H}$
$\boldsymbol{q}_{\text{aprox},1}=\text{IK}(\mathbf{H})$
$\mathbf{H}_{\text{aprox},i}=\mathbf{T}_{\text{R6}}(\mathbf{q}_{\text{aprox},i})$
$\boldsymbol{q}_{\text{aprox},i}=\text{IK}(\mathbf{H}_{\text{fake},i})$
$E=\text{error}(\mathbf{H}, \mathbf{H}_{\text{aprox},i})$
$\mathbf{H}_{\text{fake},i}=\mathbf{H}_{\text{fake},i\text{-}1}\,\text{inv}(\mathbf{H}_{\text{aprox},i\text{-}1})\;\mathbf{H}$
$E<Tolerance$?
NO
$i=i+1$
YES
Solution=$\mathbf{q}_{\text{aprox},i}$

Figure 6.2 Block diagram of the algorithm

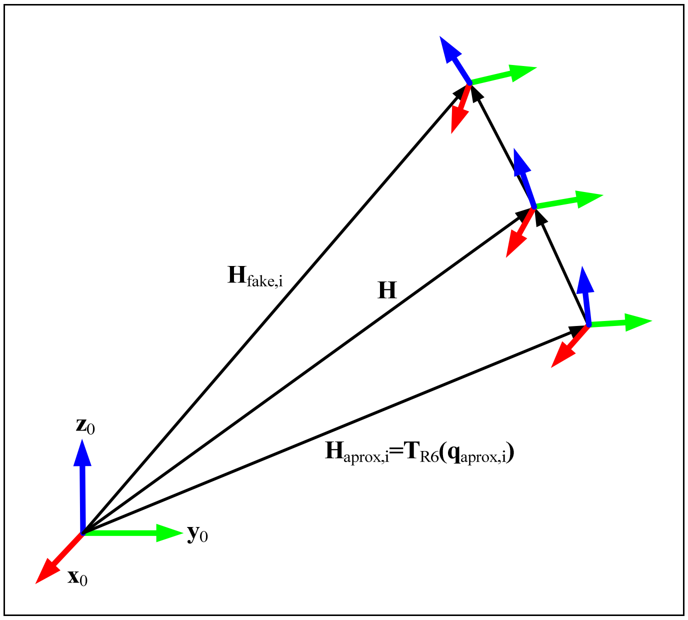

$\mathbf{H}_{\text{fake,i}}$
$\mathbf{H}$
$\mathbf{z}_0$
$\mathbf{y}_0$
$\mathbf{x}_0$
$\mathbf{H}_{\text{aprox,i}}=\mathbf{T}_{\text{R6}}(\mathbf{q}_{\text{aprox,i}})$

Figure 6.3 Iterative inverse kinematics representation for iteration *i*

The used error function takes into account the distance error and the orientation error (Schneider and Eberly, 2003):

$$E=\sqrt{x_e^2+y_e^2+z_e^2}+k\ \cos^{-1}\left(\frac{r_{e1,1}+r_{e2,2}+r_{e3,3}-1}{2}\right) \tag{6.41}$$

Considering that for one pose the joints are $\boldsymbol{q}_i$, the complete transformation error is:

$$\mathbf{H}_\mathrm{e}=\mathbf{T}_6\left(\mathbf{q}\right)\left(\mathbf{T}_{\mathrm{R}6}(\mathbf{q})\right)^{-1}=\begin{bmatrix} & & & x_e \\ & \mathbf{R}_{\mathrm{e},3\times 3} & & y_e \\ & & & z_e \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.42}$$

For most tests we took k=180/π so that we give the same importance to 1 mm and 1º ($\cos^{-1}$ gives back the angle in radians).

There is no theoretical proof that this method is always convergent. To experimentally test the convergence we used 2000 random poses to numerically compute the inverse kinematics with three different models. These three models are chosen so that the maximum errors compared to the nominal model are 1.176, 2.425 and 3.881 mm respectively, according to Table 6.1. The second model corresponds to one of the best kinematic models found in Chapter 7.

Table 6.1 Stabilization of the iterative inverse kinematics

	Iteration	1			2			3		
	Error	μ	σ	max	μ	σ	max	μ	σ	max
Model 1	Position error (mm)	0.588	0.088	1.176	0.001	0.001	0.017	0.000	0.000	0.000
	Angle error (°)	0.049	0.012	0.118	0.000	0.000	0.003	0.000	0.000	0.000
Model 2	Position error (mm)	0.925	0.245	2.425	0.002	0.001	0.026	0.000	0.000	0.001
	Angle error (°)	0.120	0.028	0.279	0.000	0.000	0.008	0.000	0.000	0.000
Model 3	Position error (mm)	1.558	0.417	3.881	0.005	0.004	0.097	0.000	0.000	0.004
	Angle error (°)	0.120	0.028	0.279	0.000	0.001	0.023	0.000	0.000	0.000

As seen in Table 6.1, for the three models used the error was really small after 3 iterations. For 4 iterations and more all errors were smaller than 5×10^{-4} mm (or $5\times10^{-4\circ}$).

CHAPTER 7

CALIBRATION METHODS

In this chapter we describe the calibration method by which we obtained best results for our working conditions. The method was applied to the ABB IRB 1600-6/1.45 robot with a Faro laser tracker measure system. Although many calibration tests have been performed we show here the most relevant ones.

We can divide a full calibration in three main steps: we must first take measurements of the robot's position at a number of different poses, then we must run an optimization procedure that finds the error parameters for which the position errors at the measurement poses would have been minimum, finally, we must verify the accuracy of the calibrated robot at various other poses.

The variables that must be taken into account to reach the best calibration may vary depending on every robot. We can consider variables the stabilization time needed by the robot to take a measurement, the design of the tool, the number of targets used, the number of measures needed for calibration or the first guess of the error parameters to start iterations.

7.1 Procedure for taking measurements

To take the position measurements we must take into account three factors:

- We must establish an arbitrary base in the working space fixed with respect to the robot. It is important to do that in case the laser tracker has to be moved between measurements or in case we want to perform an absolute calibration. Otherwise, this base is not necessary. The base is a frame that must be measured by three points as described in Section 3.3.
- It is important to let the robot stabilize for a couple of seconds before taking a measurement. Even if the robot seems to be stopped there are vibrations that add noise to the measurement system. Appendix V shows the stabilization time needed for the robot

used. It is found that the robot is fully stabilized after two seconds. For our tests we used four seconds of wait time to take the measurements.

- For a laser tracker, the angular accuracy tests (suggested accuracy tests by the Faro software) must be between the specified limits. It is strongly recommended to verify this before taking any measurements and after warming up the laser tracker.

Once these three aspects are clear we can proceed with the measurement procedure. This procedure consists of seven steps:

1. Initialization of the laser tracker measurement system: this step allows the laser tracker to warm-up. It takes around 15 minutes.
2. Warm-up of the robot during one hour. While the robot is being warmed up we can run the angular accuracy tests to check the precision of the laser tracker and proceed with compensation if the laser tracker needs it.
3. Measure an arbitrary base by three points as described before in this section.
4. Tool calibration: the procedure explained in Section 3.4.2 must be followed in this step if the dimensions of the tool are not known. Even if the dimensions of the tool are known at a high precision it is recommended to perform this tool calibration to compare both results. This step takes from 5 to 15 minutes depending on the number of targets that we want to use (in our case we used the eight corner targets described in Section 3.4.4).
5. Measurement of the two first axes to find the robot base: six measurements are needed (three measurements per axis). To predict the position of the target it is useful to know the position of the robot base referenced to an arbitrary base. In this way, using the nominal direct kinematics the positions of the targets are known with the nominal accuracy. If the nominal error is around 2 mm the laser tracker finds immediately the target without having to launch the search algorithm or having to change the position of the SMR.
6. Measurement of 18 points describing each of the six axes: this step needs an operator close to the robot to reorient the targets so that at each position one SMR target can be seen by the laser tracker. This step takes from five to ten minutes.

7. Measurement of 102 random points: fifty percent of these points must be in the desired working zone of the robot and the other fifty percent must cover the complete workspace of the robot. The orientation of the tool for these points is chosen so that one target can always be seen by the laser tracker without having to reorient the SMR. We can measure at a rate of 360 points per hour, waiting 4 seconds by measurement and moving the robot at 500 mm/s with slow accelerations. This procedure can be done in 17 minutes. If it is the first time that we calibrate the robot it is better to take more measures to deeply analyze the behavior of the robot and eliminate useless error parameters if necessary. Such as the model for the 6th axis.

It is very useful to develop an application that controls the robot and the laser tracker at the same time as the procedure of taking measures is much faster. Otherwise, the 7th step may take longer and can be dangerous for an operator. In our case we controlled the robot and the laser tracker with Matlab.

7.2 Points needed for calibration

We mentioned a minimum of 120 measurements, plus 6 measurements to find the robot base (recommendable as the procedure is much faster) plus 3 points to localisate the base (optional if the laser tracker does not have to be moved) plus the points needed to calibrate the tool, if needed (which corresponds to a minimum of 6+*n*, where *n* is the number of targets that the tool has).

If it is the first time that we calibrate a new kind of robot it is much better to take more points so that we have more information of its behavior. For example, if we take more points that describe each of the axis we can make an axis analysis as described in Section 4.1 and shown in Figure 4.3. If we do so, we can decide if a stiffness model and/or a 6th axis model is necessary to calibrate the robot (as described in Sections 4.3 and 4.2 respectively). It would also be useful to obtain the first guess to start iterations. We can do all these if we take around 158 measurements of the axis instead of the 18 measurements that we mentioned. In this case we would need a total of 240 measurements.

In addition, if it is the first time we calibrate a robot, to be able to make a pre-validation of the kinematic model obtained from optimisation before testing the model on the robot (as explained in Section 7.6) it is recommended to take more measurements all over the robot working space.

Although we tested different combinations (concerning the number of measurements) to obtain a good direct kinematic model we explain a combination based on 120 measures. Best results were obtained when we used 120 measurements or more.

The combination of the 120 measurements is divided in three groups of points:

- The 18 measurements from axes identification. Keeping all axis at zero degrees, every axis is moved one by one at the positions: −30, 0 and 30º. These 18 points are used for axes identification as well as for an entire kinematic calibration and objective function minimisation. These measurements correspond to the positions shown in Figure 3.1.
- The "ISO cube" points: 52 measurements which correspond to the largest cube allowable in the frontal robot working space (as specified in ISO 9283:1998). The orientation of the tool is chosen so that the laser tracker can always see at least one SMR target. Robot configuration is not restrained. We found that for the ABB IRB 1600-6/1.45 this cube has edge length of 650 mm.
- The "all robot range" measurements: 50 measurements inside the whole workspace of the robot (limited by the obstacles close to the robot as well as the visibility of the target by the laser tracker). Same as before, the orientation is chosen so that the laser tracker can always see at least one SMR target. Robot configuration is not restricted, so each joint can achieve any position in its full joint range. In our case, because of the obstacles, we restricted the Z value (referenced in the robot base) to be higher than 700 mm in the front side ($X \geq 150$ mm) and 1100 mm in the back side ($X < 150$ mm).

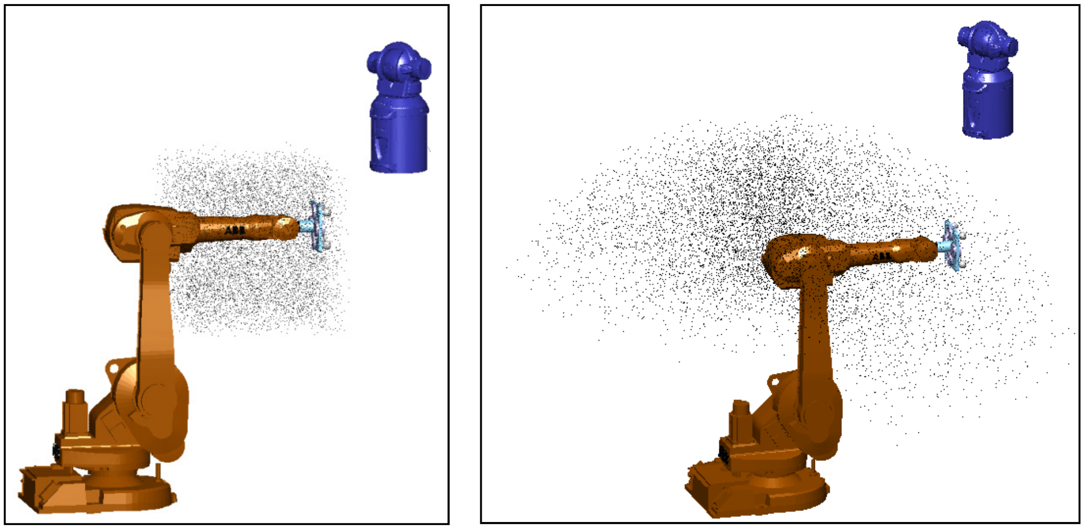


Figure 7.1 Random ISO cube points (left) and all range points (right) with respect to the robot and laser tracker.

Even if we only want to be precise on a specific area we realized that it is better to take measurements all over the robot space and include it in the optimization procedure.

7.3 Calibration procedure

Having decided which direct kinematic model we are going to use, we must make the model suitable to the measured points to find the best fit for all these measurements. The error parameters that include the kinematic model are also commonly called robot’s signature or birth certificate.

The direct kinematic model used must be applied according to the accuracy specifications needed. If the model is more complex (it has more error parameters) the inverse kinematics computation becomes more complex. Some robot companies may let the user modify some of the nominal distances and it will still compute the inverse kinematics even for the path of a linear or circular movement. In other cases we may need an external filter (software) that changes our targets into fake targets. For example, in some cases we may need maximum path accuracy so we can use a simpler model before using a more complex model that only allows improvement of the accuracy at the start and end poses of the path.

The optimization problem to find the error parameters is solved as explained in Section 3.5. As described in that section, the algorithm used to iterate to solve the optimization problem is the Nelder-Mead method. The objective function is the maximum position error of all used measures for tools number one, two and three (according to Section 3.4.4). The robot models used are described in Section 5.4. The most complete model is the 34-parameter model. To obtain the error parameters for this model, before starting the optimization algorithm we used as a first guess the error parameters obtained from the kinematic calibration, which corresponds to a model found by the 18 measurements taken moving one axis at a time (the extraction of the these kinematic parameters is explained in Section 3.1).

We tried other objective functions instead of the minimization of the maximum error, like least-squares optimization or mean optimization. Although we may improve the mean and standard deviation errors a little, the maximum error values are much worse. Although we also tried proportional error parameters for the encoders (as described in equation (3.31)) we realized that they did not improve the precision, so they have not been taken into account.

7.4 Verification procedure

Once we have found one or more kinematic models that try to improve the nominal direct kinematics we are ready to test the real accuracy. It is at this point that the iterative inverse kinematics is important to create fake targets.

We must test the accuracy of a direct kinematic model in the area where we want to be precise and with the conditions that we are going to use (different payloads, different tools, etc.).

If we use more points in Section 7.2, the statistics found should look more similar to the expected values (using the method explained in Section 7.6).

In our case we proceeded with many different verification procedures using different models found in Section 7.3, we also often used the expected error values to predict how good a kinematic model would be. Results are shown in Section 7.7.

7.5 Error vs. iterations

When the optimization algorithm iterates to find the minimum value for the objective function it is supposed to find an equivalent or better combination of error parameters at the end of each iteration.

During a complete minimization procedure we looked at the evolution of the expected error for three different groups of points:

1. The first group of 140 points was used by the objective function: 54 points correspond to the robot axis, 46 random points correspond to the ISO cube area and 40 random points correspond to all the range of the robot (except for the obstacles).
2. The second group corresponds to 20 random points in the ISO cube area. These points are different from the 46 taken on the first group.
3. The third group corresponds to 20 random points in all the range of the robot (except for the obstacles). These points are different from the 40 taken on the first group.

During 2000 iterations minimizing the objective function with a direct kinematic model composed by 34 parameters we stored the expected error for the two extra groups of points every ten iterations.

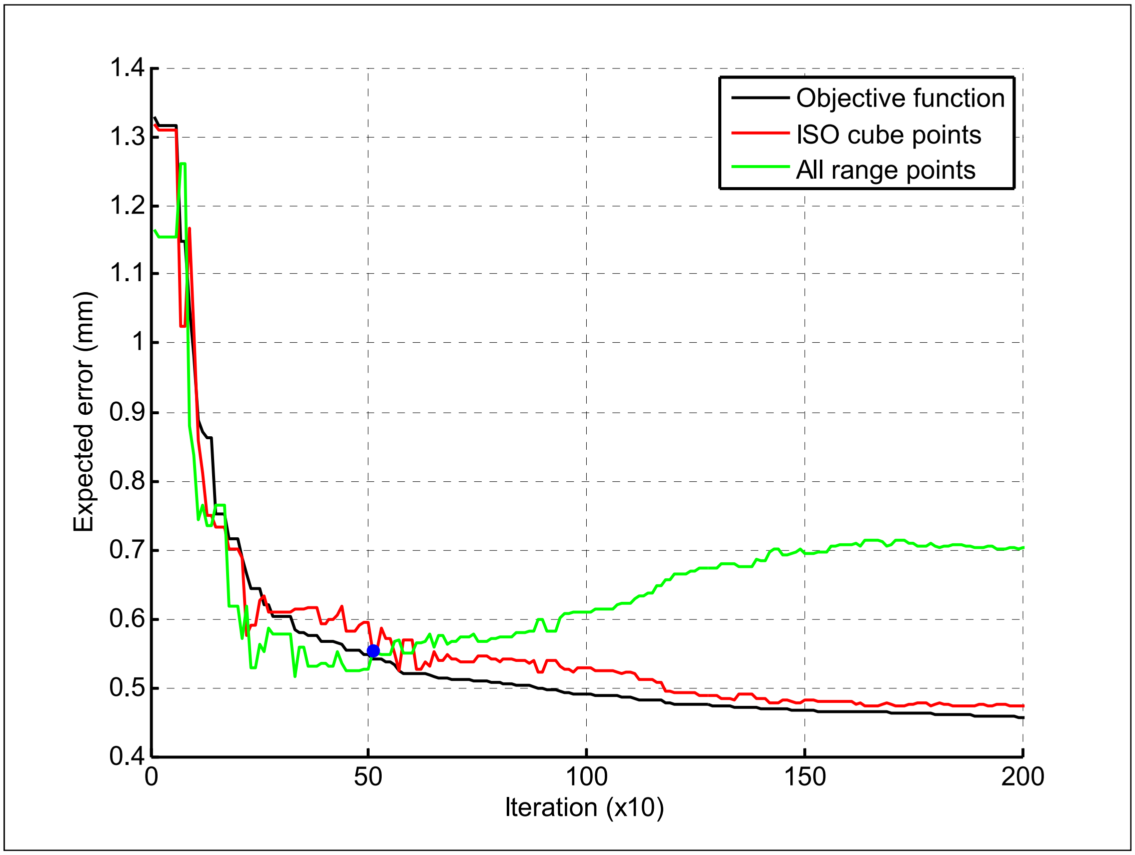


Figure 7.2 Evolution of calibration step

As seen in Figure 7.2, while the objective function tends to improve every time, the other two groups of points have a different behavior through the whole iteration. For example, the group of the all robot range points (green line) reaches its minimum between 300 and 500 iterations after the algorithm started. Between 500 and 1500 iterations the robot model gets worse for these points as the objective function improves.

The blue point corresponds to the minimum taking into account all observations. Although we probably could have obtained better robot kinematic models treating this information we considered the result after 2000 iterations.

This information tells us that there is a local minimum for the first group of points that does not correspond to the local minimum for the other groups of points. To avoid similar results we should use the observability matrix and improve the optimization method.

7.6 Expected error vs. real position error

Considering $\mathbf{T}_6$ as the nominal direct kinematics analytical expression:

$$\mathbf{T}_6(\boldsymbol{q}) = \mathbf{A}_1^0\mathbf{A}_2^1\mathbf{A}_3^2\mathbf{A}_4^3\mathbf{A}_5^4\mathbf{A}_6^5 \tag{7.1}$$

If we take $\mathbf{T}_{R6}$ as the found (and improved) direct model, which takes into account the robot errors:

$$\mathbf{T}_{R6}(\mathbf{q}) = \mathbf{A}_{R1}^{\;0}\mathbf{A}_{R2}^{\;1}\mathbf{A}_{R3}^{\;2}\mathbf{A}_{R4}^{\;3}\mathbf{A}_{R5}^{\;4}\mathbf{A}_{R6}^{\;5} \tag{7.2}$$

For a given position of the robot $\mathbf{q}_A = [q_1, q_2, q_3, q_4, q_5, q_6]^T$ we can have the measured point (real position for $\mathbf{q}_A$), the nominal kinematics position result ($\mathbf{T}_6(\mathbf{q}_A)$) and the real direct model position result ($\mathbf{T}_{R6}(\mathbf{q}_A)$). As shown in Figure 7.3.

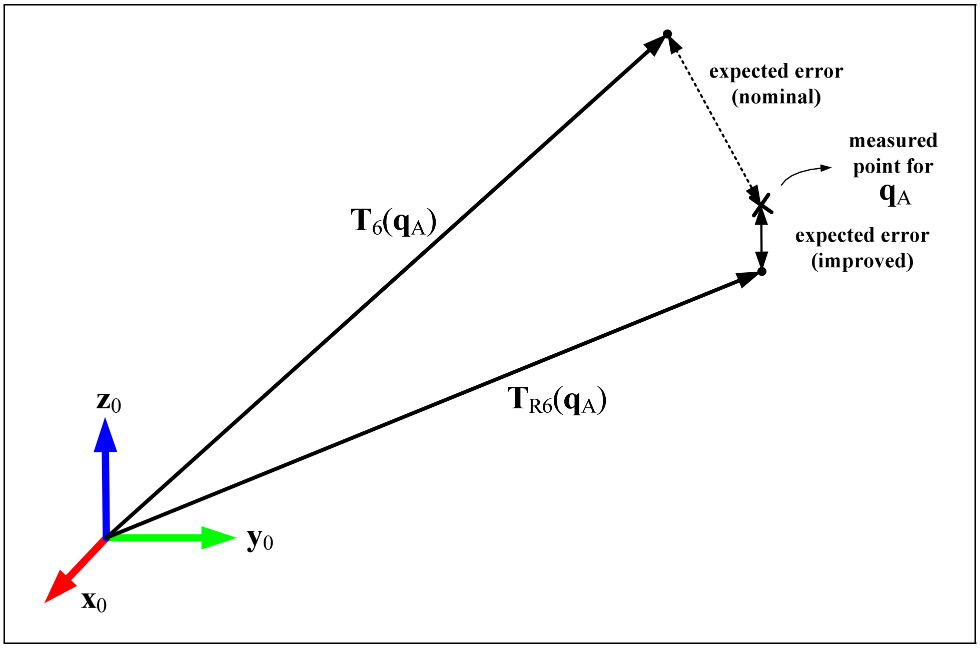

expected error
(nominal)
measured
point for
$\mathbf{q}_A$
expected error
(improved)
$\mathbf{T}_6(\mathbf{q}_A)$
$\mathbf{T}_{R6}(\mathbf{q}_A)$
$\mathbf{z}_0$
$\mathbf{y}_0$
$\mathbf{x}_0$

Figure 7.3 Analysis of the expected error

The expected error (improved) is the distance between the measured point and the real direct kinematics for $\mathbf{q}_A$.

Ideally, for this given $\mathbf{q}_\mathrm{A}$ position we should calculate the iterative inverse of $\mathbf{T}_6(\mathbf{q}_\mathrm{A})$ obtaining another point $\mathbf{q}_\mathrm{B} \cong \mathbf{q}_\mathrm{A}$ that should bring the robot to a position close to the measured point. Errors have been exaggerated in the Figure. In next equation ${\mathbf{T}_{\mathrm{R}6}}^{-1}$ represents the inverse of the direct representation $\mathbf{T}_6\ (\mathbf{q}_\mathrm{A})$:

$$\mathbf{q}_\mathrm{B} = {\mathbf{T}_{\mathrm{R}6}}^{-1}\left[\mathbf{T}_6(\mathbf{q}_\mathrm{A})\right] \cong \mathbf{q}_\mathrm{A} \tag{7.3}$$

This is the way we must do if we want to quickly evaluate the precision of a found direct kinematics model. We do not need to take other measurements (like in this case it would be $\boldsymbol{q}_\mathrm{B}$) if we have a lot of measurements to test with (since the measurements used to validate the model must be different from the ones used for the minimization of the objective function).

7.7 Verification results

We performed a complete study of our robot considering a lot of tests with the expected errors. We did that to decide how many measurements we needed to stablish a calibration method and to see if the effect of taking random points had any influence on the final results. We also wanted to proove that the expected error was similar to the error found when the verification was performed.

Taking into account the three groups of points mentioned in Section 7.2 to calibrate the robot, we took 158 measurements from moving each axis individually: 17 measurements from each of the first five axis (each axis is moved from −40 to +40º and back to −40º by steps of 10º keeping all other axis at 0º) and 73 measurements from the 6th axis (which is moved from −180 to +180º and back to −180º by steps of 10º). From these 158 measurements we performed an analisys as we did in Section 4.2. We found the following error parameters from the 6th axis analisys, that we used as first guess to start the optimization algorithm (when using the 34-parameter model): a_1=-2.802×10^{-4} , b_1=-7.081×10^{-4} , a_2=-9.494×10^{-4} , b_2=-3.080×10^{-4} which are similar to the parameters already found in Table 5.2. Arbitrarily we considered ω=1, and the parameter a_0 is already considered by the parameter $\delta\theta_6$ so we did not take it into account.

For the second and third groups of points mentioned in Section 7.2 we took 1000 measurements per tool point. As we have eight tool points (or SMR targets) we took 8000 measurements for the area of the ISO cube and other 8000 measures for all robot range. We took the measurement twice: without extra charge and with extra 3 kg (the final payload was 3 kg and 6 kg respectively). After the test we realised that 48 measurements were not visible for the cube ISO group of points and 286 measures were not visible for the all robot range group of points.

With all these 32000 measurements (plus 158 measurements from axis identification), everytime we wanted to perform an optimisation test we picked random points that corresponded to a tool weigh of 3 kg (without extra charge) and for tool targets 1, 2 and 3, what makes a bit less than 6000 measurements as some targets cannot be seen. We evaluated the expected error for all those 32000 points (with and without extra charge, for all 8 tool targets). Several optimization tests were performed, here we show the two most relevant ones.

We will often show the error value obtained from adding the mean plus three times sigma, which statistically corresponds to the maximum error for the 99.73% of the observations. All errors correspond to an absolute calibration unless otherwise specified. Sections 7.7.3.6 and 7.7.3.7 offer better results if we want to perform a relative calibration. All measurements were taken after the robot was stabilized for four seconds and at a robot speed of 500 mm/s.

7.7.1 Results from nominal kinematics model

Before giving any improved kinematic model we show here the expected nominal error for the 8000 measurements taken, wich correspond to the first calibration method (only the base and the tool are calibrated). As shown in Table 7.1 the worst position error is 2.326 mm.

Table 7.1 Expected position error for the nominal kinematic model for 8 targets.

	Tool load	μ (mm)	σ (mm)	max (mm)	μ+3σ (mm) (99.73% CI)
Errors for ISO cube area	3 kg	1.004	0.167	1.890	1.505
Errors for all range area		0.961	0.214	2.268	1.604
Errors for ISO cube area	6 kg	0.970	0.155	1.963	1.434
Errors for all range area		0.979	0.222	2.326	1.643

These results are the reference to compare all other models, so other models found should be better than the nominal position errors shown here.

7.7.2 Results from entire kinematic calibration model

We tried four different kinematic calibrations (using the 26-parameter model that takes into account the geometric error parameters), using from 18 measurements (3 per axis) to 158 measurements (17 for axes one to five and 73 for axis six). Since the points are measured with the first target (which corresponds to the target 1 from table Table 3.1) we also show the position errors for the case of just one target (target one).

Table 7.2 Expected position error for the kinematic calibration model (3 kg).

		Target 1			All 8 targets		
Number of points used	area	μ (mm)	σ (mm)	max (mm)	μ (mm)	σ (mm)	max (mm)
18 points	ISO cube	0.452	0.269	2.312	0.416	0.262	2.459
54 points		0.528	0.307	2.434	0.467	0.303	2.637
102 points		0.544	0.309	2.509	0.479	0.305	2.713
158 points		0.549	0.308	2.507	0.482	0.304	2.726
18 points	All range	0.741	0.315	2.575	0.736	0.304	2.738
54 points		0.836	0.336	2.740	0.815	0.325	2.963
102 points		0.844	0.342	2.794	0.825	0.328	3.032
158 points		0.847	0.342	2.810	0.826	0.328	3.041

Taking a look at Table 7.2 we can say that the best kinematic calibration is obtained when we use 18 measurements. As we increase the number of measurements the position error is worse for the kinematic calibration. Comparing these results with the nominal values we can see that the mean is always improved (mainly for the area of the ISO cube, as the axis points are mostly in this area) but maximum values are worse. Table 7.3 shows the expected errors for 6 kg of payload and for the full kinematic calibration from 18 measurements (or points).

Table 7.3 Expected position error for the kinematic model (6 kg).

		Target 1			All 8 targets		
Number of points used	Area	μ (mm)	σ (mm)	max (mm)	μ (mm)	σ (mm)	max (mm)
18 points	ISO cube	0.480	0.296	2.461	0.396	0.279	2.624
	All range	0.795	0.344	2.813	0.773	0.322	2.818

Even if the full kinematic calibration was performed with only one target and 3 kg of payload, we can see that the error for the other seven targets is similar, so we can say that the mean of the orientation errors is also improved.

7.7.3 Optimisation test results from 80 to 1000 identification measurements

We first tried to optimizate different combinations from the three groups of points (axis points, ISO cube points and all range points). We always used 18 points or more from axis analisys so that it was possible to find a first kinematic model as a first guess to start iterations (which gave better results). Considering the axis measurements and adding more measurements to the optimization algorithm (a total ranging from 80 to 1000 measurements was tested). We combined adding measurements only from the ISO cube, only from all robot range and from both groups of measurements. With this test we found that it is better to combine both groups of points even if we only want to be precise in a specific area (like the ISO cube).

7.7.3.1 Results for the 34-parameter model

When we used the 34-parameter model, the objective function (maximum error for the minimized measurements) ranged between 0.5 mm and 1.0 mm. The expected error for the 8000 ISO cube points was between 0.7 mm and 1.6 mm and the expected error for the 8000 all range points was between 0.85 mm and 1.60 mm.

One of the best kinematic models found in this test corresponds to 200 measurements (54 measurements for a first full kinematic calibration plus 76 measurements from the ISO cube area plus 70 measurements from all range area). The result of the objective function was 0.586 mm. Some information concerning the expected values from the four groups of 8000 measurements is given in Table 7.4 and Table 7.5 (first table with 3 kg of payload and second table with 6 kg of payload), we also performed a verification with 1000 measurements for each case (for targets from 1 to 8, what means 125 measurements per target), so a total of 4000 measurements were taken for a first verification (a few measurements could not be visible). Finally, we also performed a verification using the same 8000 measurements that we used to obtain the expected error for the case of 3 kg payload.

Table 7.4 Position errors for a 34-parameter calibration from 200 identification measurements (3 kg)

Area	ISO cube				All range			
Error (mm)	μ	σ	max	μ+3σ	μ	σ	max	μ+3σ
Expected from 8000 measures	0.272	0.065	0.714	0.466	0.321	0.072	0.858	0.538
Found from 1000 measures	0.277	0.063	0.647	0.466	0.329	0.070	0.703	0.539
Found from the 8000 measures	0.286	0.072	0.793	0.502	0.345	0.079	0.927	0.582

Table 7.5 Position errors for a 34-parameter calibration from 200 identification measurements (6 kg)

Area	ISO cube				All range			
Error (mm)	μ	σ	max	μ+3σ	μ	σ	max	μ+3σ
Expected from 8000 measures	0.288	0.070	0.851	0.497	0.353	0.096	1.056	0.641
Found from 1000 measures	0.270	0.067	0.658	0.471	0.346	0.085	0.798	0.601

First of all, we can see that the expected error values correspond to the validation error values. We can also see that if we take more points a much greater maximum value is found. However, if we use the statistic indicator of the 99.73% confidence interval (μ+3σ) we find a maximum value of 0.601 mm from 1000 measurements and an expected maximum value of 0.641 mm from 8000 measurements, for 6 kg of payload (maximum load) for the 99.73% of the observations.

On the other hand, if we are working in the ISO cube area with 3 kg of payload we will find a maximum value of 0.502 mm for the 99.73% of the observations.

7.7.3.2 Results for the 11-parameter model

We performed the same expected error tests for the 11-parameter and 16-parameter models. We realised that the 11-parameter model is considerably better than the nominal but the 16-parameter model does not improve the 11-parameter model. So we only validated the 11-

parameter model obtaining the results shown in Table 7.6 and Table 7.7 (the best expected error was found from a calibration made with 400 measurements).

Table 7.6 Position errors for a 11-parameter calibration from 400 identification measurements (3 kg)

Area	ISO cube				All range			
Error (mm)	μ	σ	max	μ+3σ	μ	σ	max	μ+3σ
Expected from 8000 measures	0.580	0.149	1.550	1.026	0.813	0.173	1.762	1.330
Found from 1000 measures	0.589	0.148	1.483	1.033	0.794	0.173	1.727	1.313

Table 7.7 Position errors for a 11-parameter calibration from 400 identification measurements (6 kg)

Area	ISO cube				All range			
Error (mm)	μ	σ	max	μ+3σ	μ	σ	max	μ+3σ
Expected from 8000 measures	0.619	0.147	1.522	1.060	0.858	0.188	2.035	1.423
Found from 1000 measures	0.636	0.155	1.467	1.101	0.849	0.188	1.928	1.413

We can see that in all cases the position error is better than the nominal error.

7.7.3.3 Graphics of the errors found

The following graphics show each measurement error found for the tables in Sections 7.7.3.1 and 7.7.3.2. For each of the four cases we combined the 3 kg and 6 kg tool charges with the ISO cube area and all robot range area. We plot the 1000 found position errors for the nominal model, the 11-parameter model and the 34-parameter model.

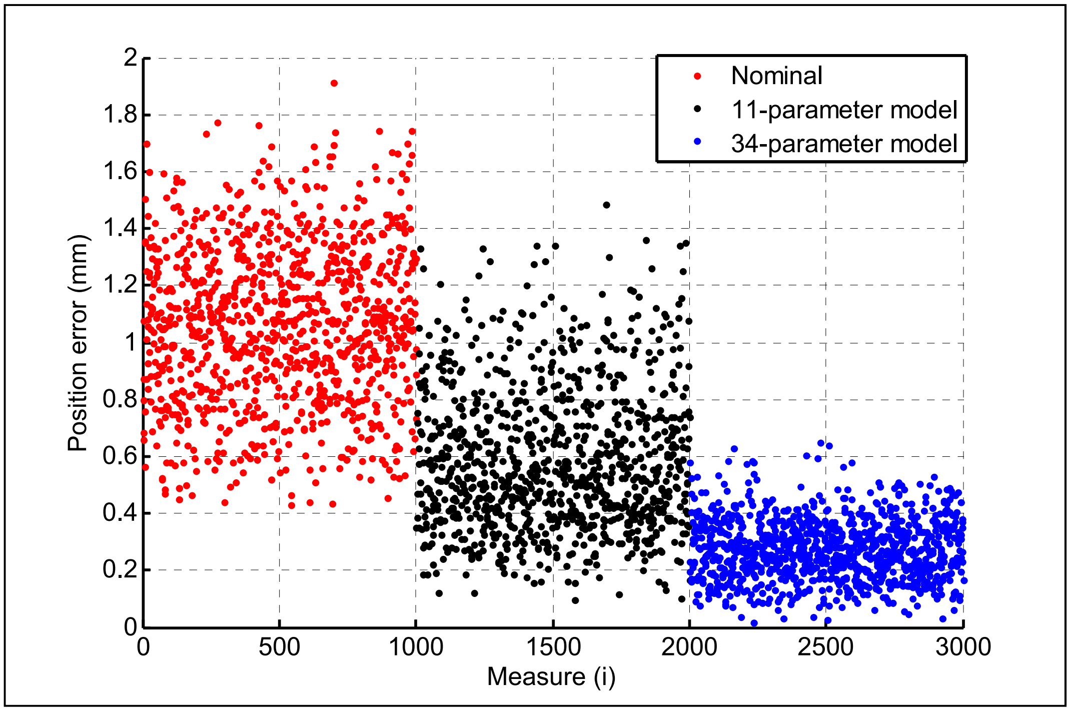


Figure 7.4 Position error in ISO cube area (3 kg)

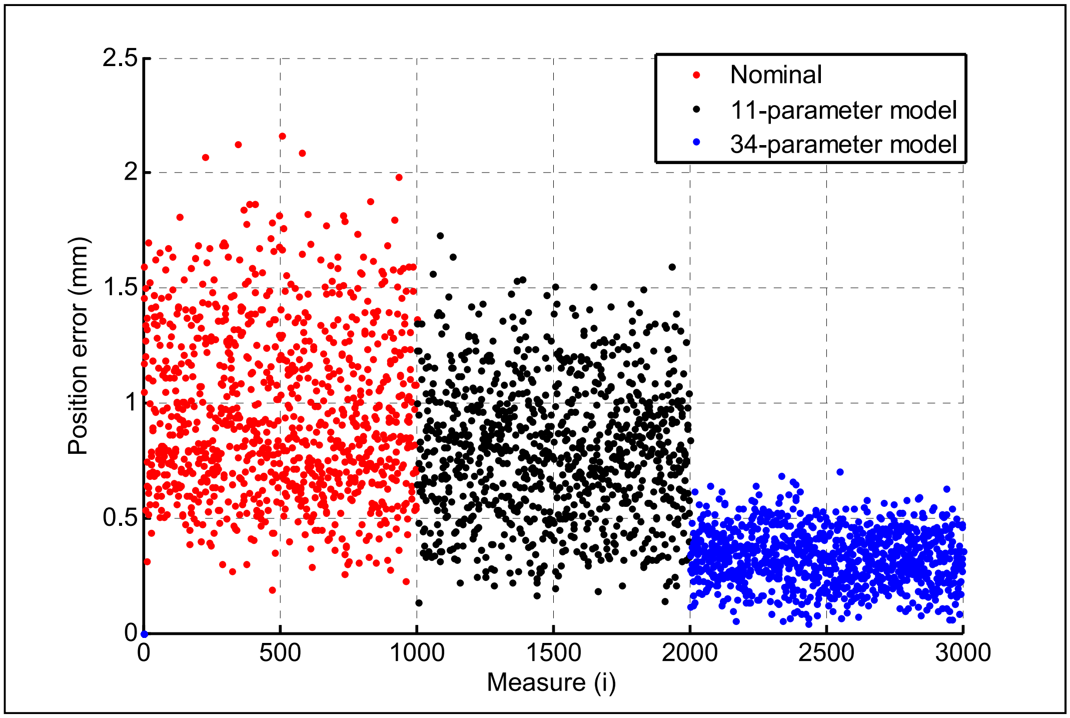


Figure 7.5 Position error in all robot range area (3 kg)

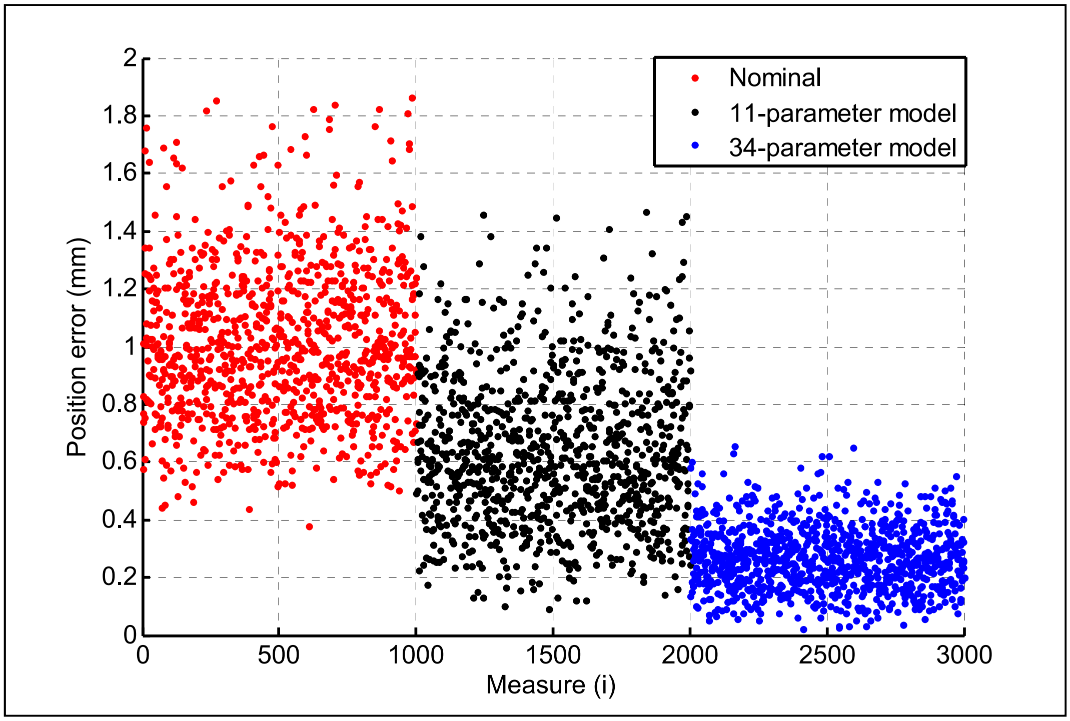


Figure 7.6 Position error in ISO cube area (6 kg)

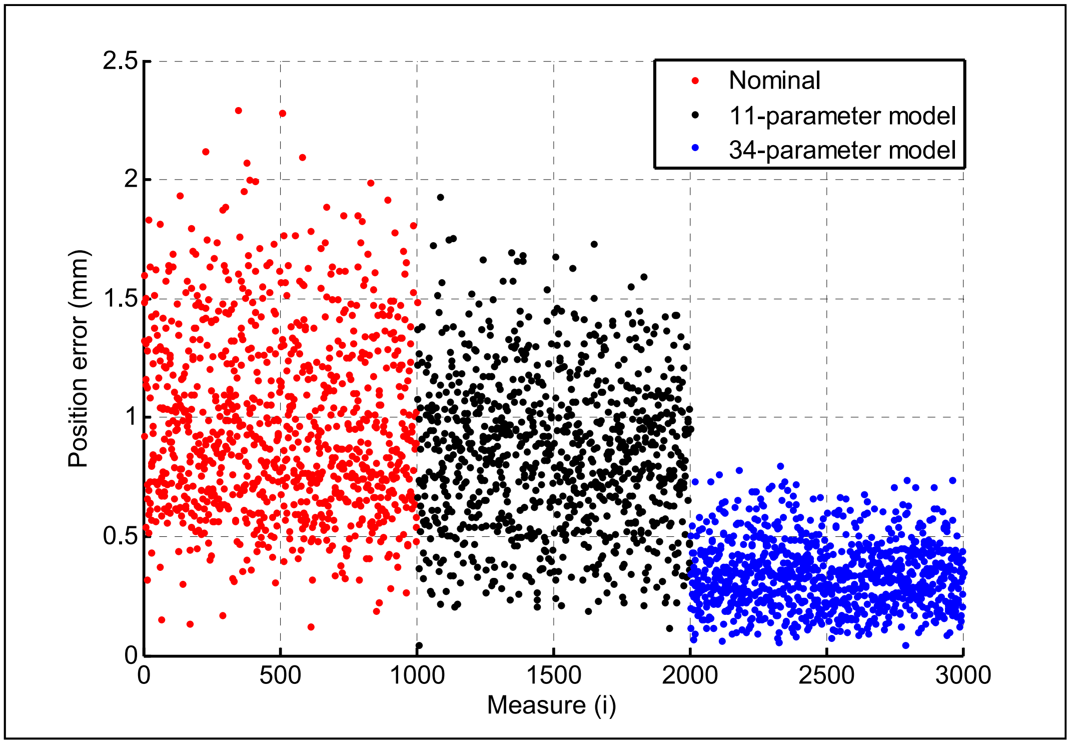


Figure 7.7 Position error in all robot range area (6 kg)

All of these values correspond to an absolute and arbitrary base (absolute calibration), but if the position of the base is not important we can perform a relative calibration.

7.7.3.4 Position and orientation errors from ISO 9283 tests

The norm ISO 9283 establishes some procedures to test the accuracy and repeatability of a robot. However, the tests proposed are not complete enough to have an idea of how accurate a robot is. In addition, the localisation of the robot base is not specified and the dimensions of the tool that we should use are not clear enough.

We show here the ISO 9283 accuracy results (described in Section "7.2.1: Pose accuracy" in the ISO norm) for four different kinematic models: the three models shown in Section 7.7.3.3 (nominal, 11-parameter and 28-parameter) and the full kinematic model (directly obtained from a kinematic calibration by axis). To measure the pose for each position we used the targets 1, 2 and 3 from the tool described in Section 3.4.1, payload was 3 kg. In the Annex I (where repeatability results are shown) it is specified how the five measurements from the ISO cube are taken in 30 cycles.

Table 7.8 ISO 9283 position and orientation errors for the four models (3 kg)

	Position errors (mm)		Orientation errors (°)	
Model	mean	max	mean	max
Nominal	0.964	1.122	0.103	0.18
11-parameter	0.559	0.62	0.107	0.165
Kinematic	0.288	0.458	0.103	0.183
34-parameter	0.316	0.371	0.077	0.154

Mean and maximum position errors are shown in the previous Table. We also show in Figure 7.8 all position and orientation errors by point ID and by cycle.

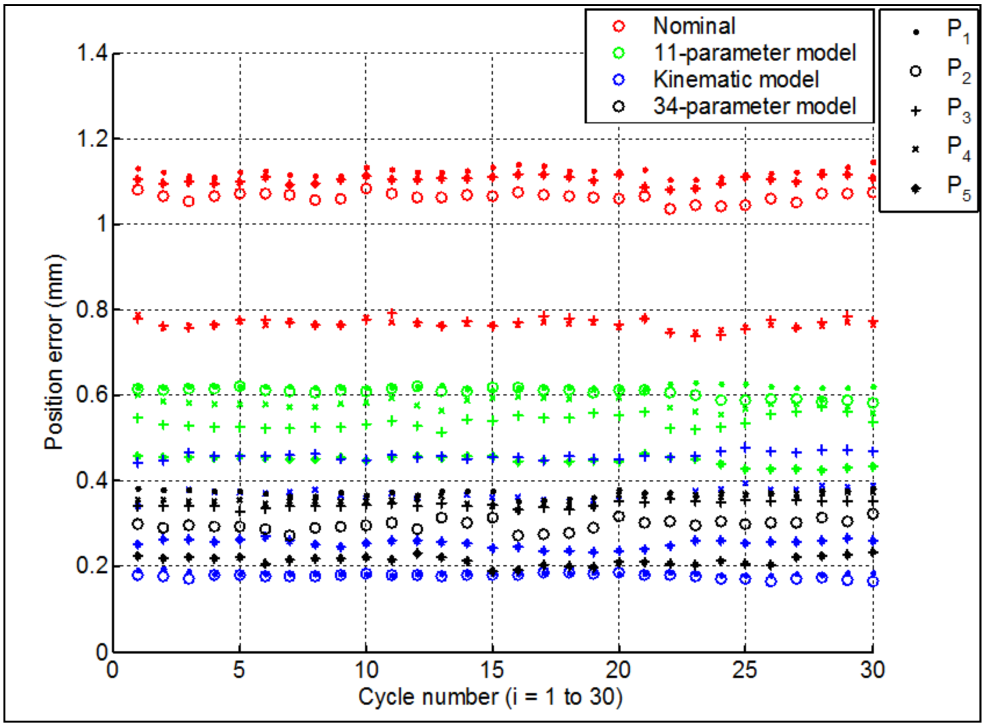


Figure 7.8 ISO 9283 position errors reduced to the 6th axis (3 kg)

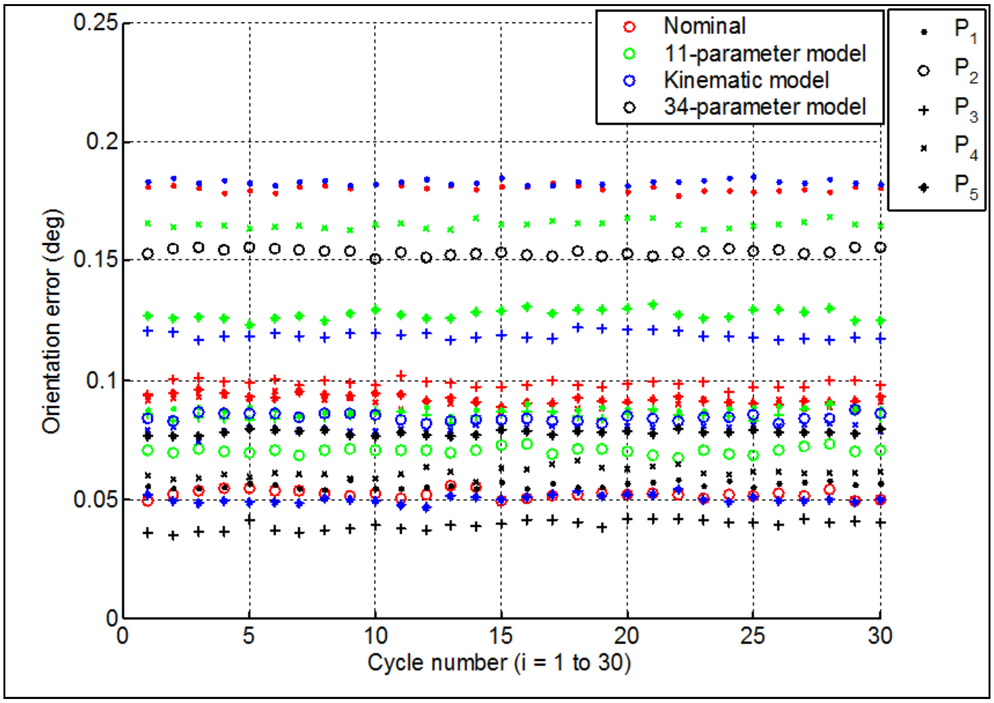


Figure 7.9 ISO 9283 orientation errors (3 kg)

As we can see, the position and orientation errors obtained here are much better than the ones shown previously, since there are just five positions being measured. We can even see that the mean position error of a full kinematic calibration is better than the 34-parameter model. So ISO 9283 accuracy tests are not enough to validate a good model.

7.7.3.5 Orientation error analisys

In the previous section (concerning the ISO 9283) we introduced the orientation errors. We performed another pose test for each of the four models shown in the previous section measuring 1000 poses. Payload is 3 kg. Here we show the position and orientation errors. The position errors are reduced to the 6th axis, so the errors of the first axis are only reflected as orientation errors.

Table 7.9 Position and orientation errors found for the four models (3 kg)

	Position errors (mm)				Orientation errors (º)			
Model	μ	σ	max	μ+3σ	μ	σ	max	μ+3σ
Nominal	0.969	0.078	1.241	1.204	0.108	0.022	0.229	0.174
11-parameter	0.428	0.086	0.96	0.685	0.109	0.015	0.178	0.154
Kinematic	0.237	0.075	0.98	0.463	0.103	0.024	0.232	0.176
34-parameter	0.223	0.043	0.451	0.351	0.082	0.020	0.166	0.141

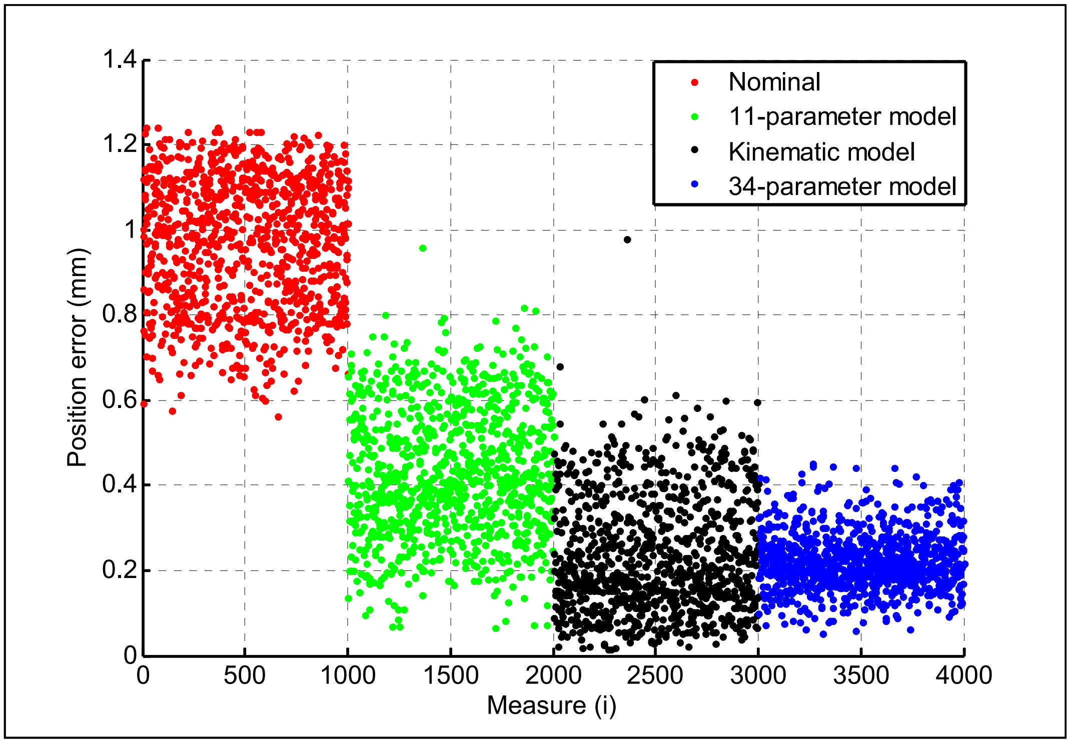


Figure 7.10 Position errors reduced to the 6th axis for the four models (3 kg)

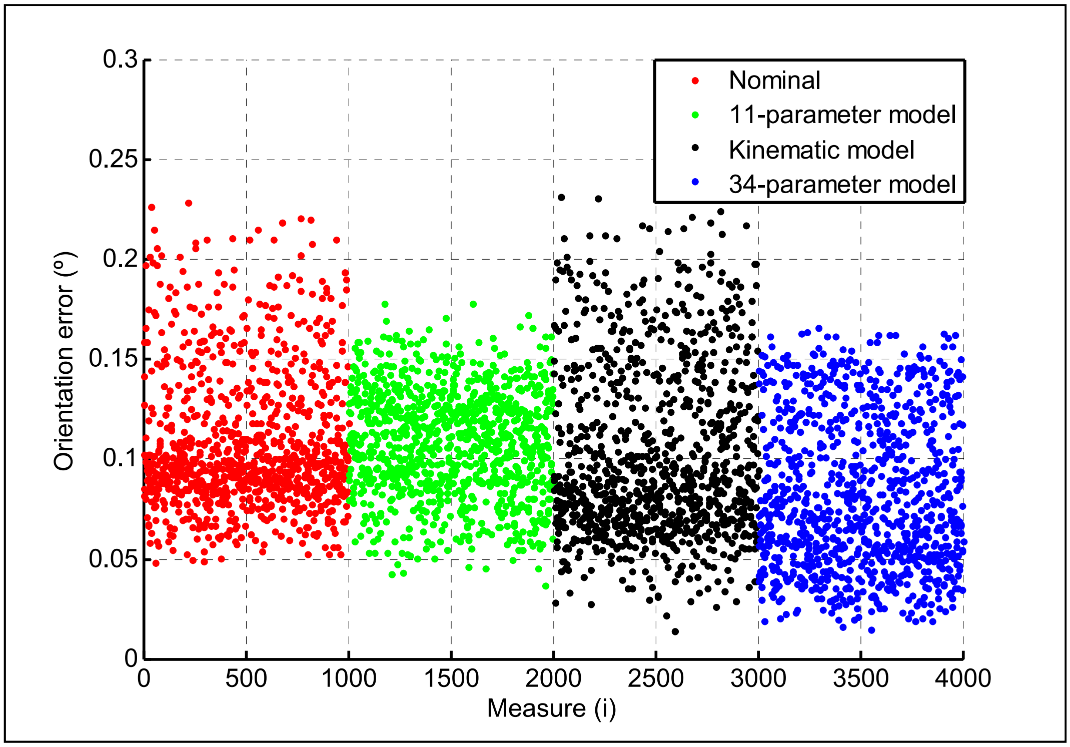


Figure 7.11 Orientation errors for the four models (3 kg)

We can clearly see that if we measure the position on the 6th axis, position error results are much better. We can also see that maximum orientation errors are worse for the kinematic model than for the nominal model. We can also see that the 11-parameter model and the kinematic model have one common pose which has a much worse position than all other measurements (points number 1342 and 2342).

7.7.3.6 Relative analyses from a virtual frame

Until now we only gave absolute calibration position errors. Here we show the results concerning the same 34-parameter model (found in Section 7.7.3) if we only want to be precise in a small area. We choosed a sphere of diameter 400 mm centered in the position {X=845.7 mm; Y=200 mm; Z=965.7 mm } with respect to the robot base frame, as shown in Figure 7.12. Robot configuration is chosen so that it is the closest one to the figure (robot at the "zero" position).

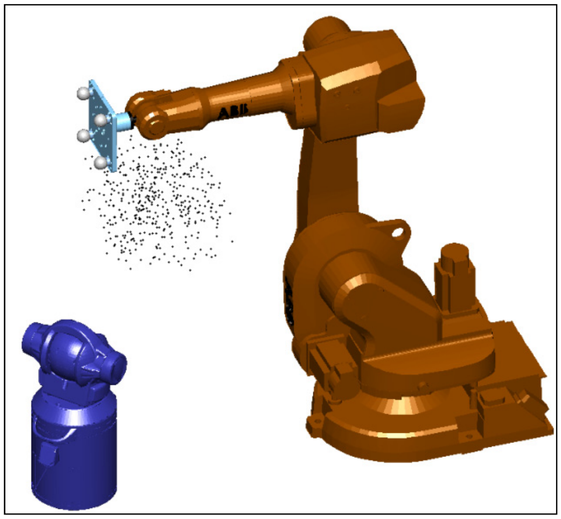


Figure 7.12 Position of the 434 sphere points chosen for relative analysis

We took 434 measurements with any of our eight targets (for each measurement a random target is chosen) and with 3 kg of payload. We calculated the mean position vector (the mean coordinate errors for all measured points), which was {δx=0.018 mm; δy=−0.117 mm; δz=0.020 mm } with respect to the robot base frame. Statistics are shown in Table 7.10.

Table 7.10 Relative position errors (from virtual frame, 3 kg)

Model	μ (mm)	σ (mm)	max (mm)	μ+3σ (mm)
34-parameter	0.179	0.048	0.462	0.323

Figure 7.13 shows the dispersion of errors.

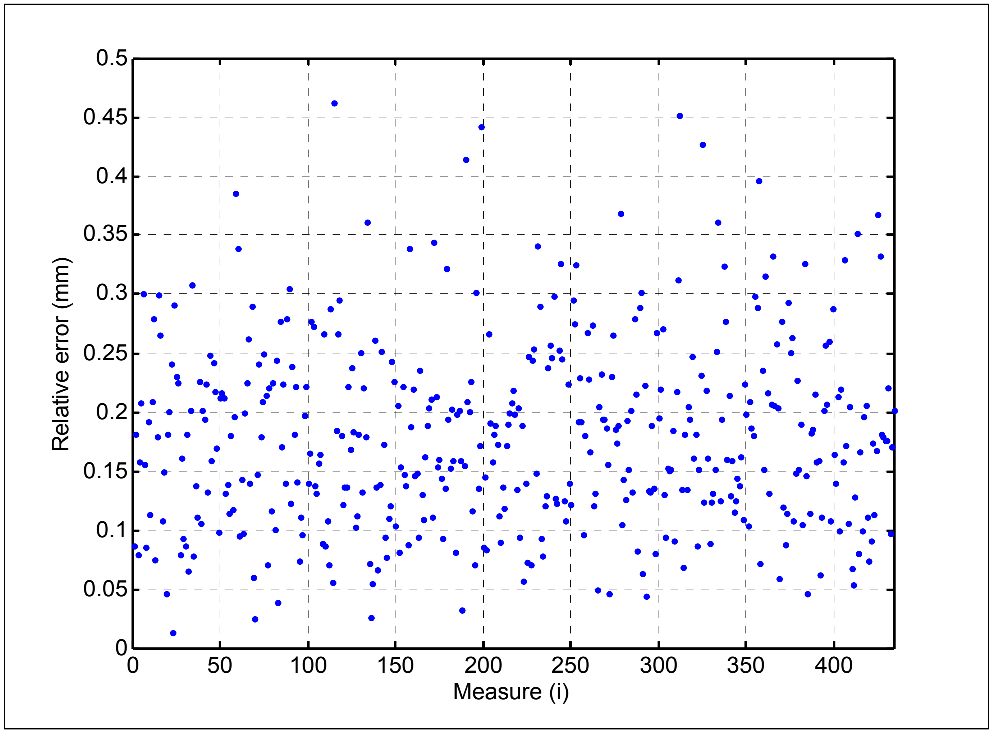


Figure 7.13 Relative position errors from a virtual frame (3 kg)

As we can see, results are much better if we work relatively in a small area. In the 99.73 % of the cases we will find that the error is smaller than 0.323 mm.

7.7.3.7 Relative analisys from 3 points

We repeated the test performed in the previous section but instead of using a virtual frame we used a coordinate system by measuring 3 targets for 3 positions of the robot (in comparison with the relative analisys from the preious section, errors should be added by the imperfections of the first 3 points). Results shown in Figure 7.14 are obtained.

Table 7.11 Relative position errors (from 3 targets, 3 kg)

Model	μ (mm)	σ (mm)	max (mm)	μ+3σ (mm)
34-parameter	0.257	0.052	0.502	0.412

Figure 7.14 shows the dispersion of errors.

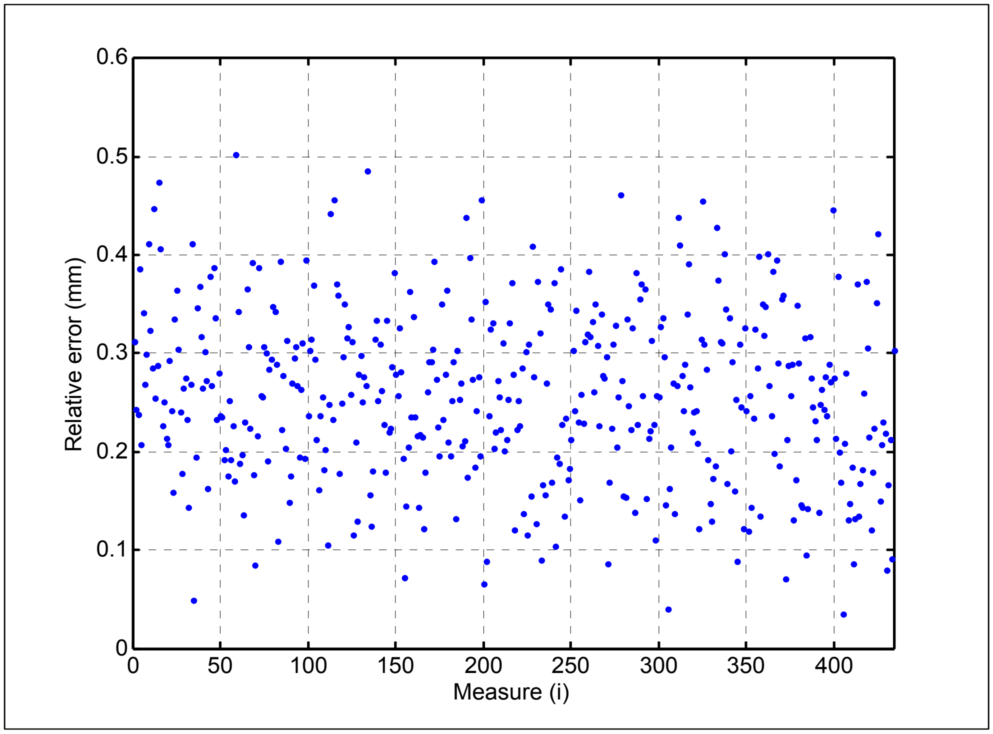


Figure 7.14 Relative position errors from 3 targets (3 kg)

These errors will strongly depend on the first 3 measured points to generate the coordinate system. If the position error for these 3 first points is bigger, the position error for all other points will be worse. In these case we find that the 99.73 % of times we will have an error smaller than 0.412 mm.

7.7.4 Results found for 34-parameter calibrations from 80 to 200 measurements

The 34-parameter calibration results shown in Section 7.7.3 correspond to a procedure found from a calibration of 200 points. We performed other three 34-parameter calibration procedures with 80, 120 and 180 measurements (using the same 34-parameter model). All measurements are taken randomly and the identification measurements used in one procedure are not used in another procedure.

The verification is performed by pose measurements (from tool targets 1, 2 and 3). The payload is 3 kg. We show position and orientation errors for each of these three 34-parameter procedures found plus the 34-parameter results obtained in Section 7.7.3 (the test performed with the 34-parameter procedure in Section 7.7.3.5 is repeated from different identification measurements). Statistics obtained are shown in Table 7.12.

Table 7.12 Position and orientation errors for the 34-parameter procedures (3 kg)

Procedure from:	Position errors (mm)				Orientation errors (°)			
	μ	σ	max	μ+3σ	μ	σ	max	μ+3σ
80 measures	0.345	0.056	0.568	0.512	0.090	0.023	0.185	0.157
120 measures	0.291	0.043	0.472	0.419	0.097	0.018	0.171	0.150
180 measures	0.292	0.043	0.474	0.422	0.081	0.019	0.164	0.138
200 measures	0.292	0.040	0.453	0.412	0.080	0.021	0.174	0.144

The history of all position and orientation errors is shown in Figure 7.15 and Figure 7.16.

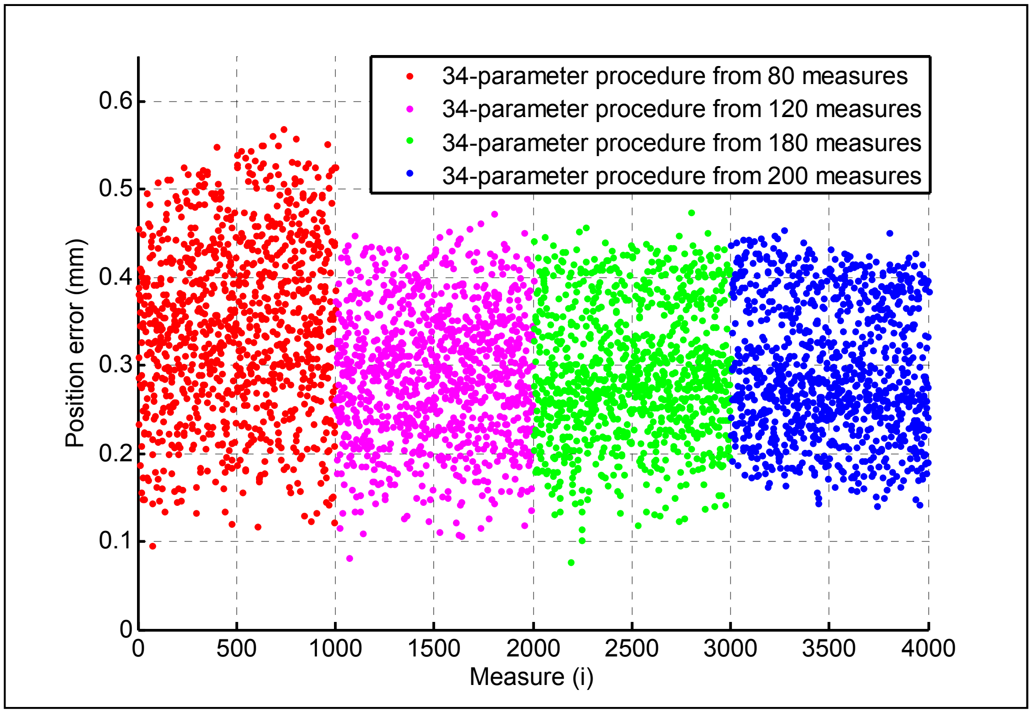


Figure 7.15 Position errors reduced to the 6th axis for the four 34-parameter procedures (3 kg)

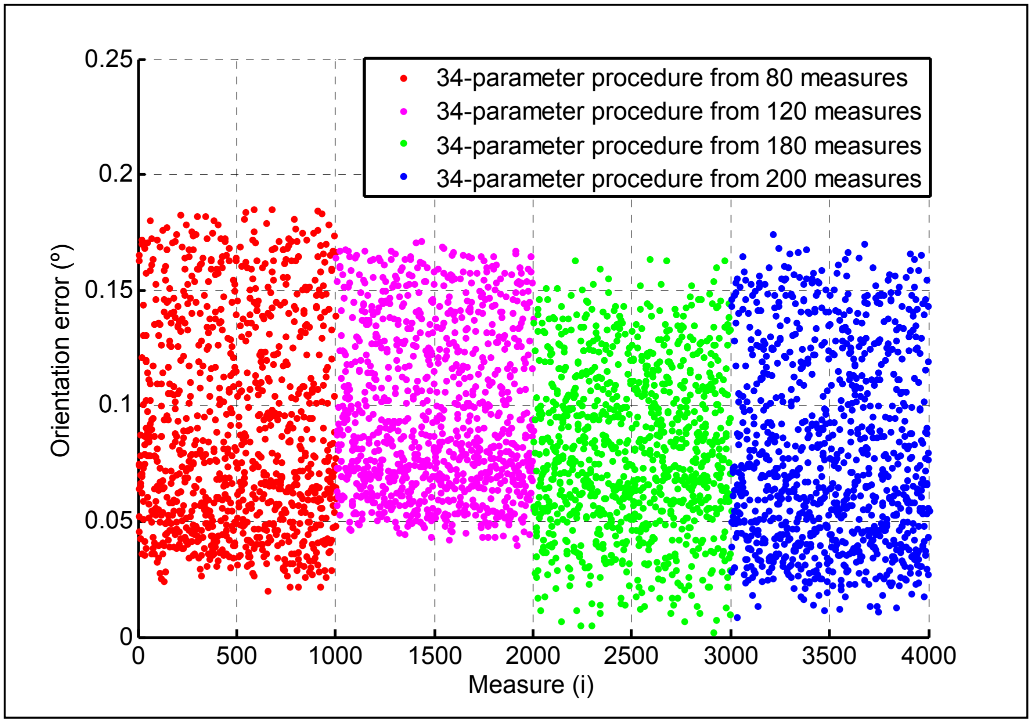


Figure 7.16 Orientation errors for the four 34-parameter procedures (3 kg)

We can see that for the first three calibration procedures we obtain similar results. Although taking more points may be more reliable, taking a look at these figures we see that even from a calibration of 80 measurements we obtain satisfactory results.

7.7.5 Worst random 34-parameter procedures from 80, 120, 180 and 200 measurements

Since we did not perform any observability analysis to choose the position measurements we wanted to see if the random effect had any influence on the final results. We generated 20 different combinations of identification measurement groups: five combinations of a total of 80 measurements, five combinations of a total of 120 measurements, etc.

For each of the four groups (80, 120, 180 and 200 measurements) we tried five 34-parameter procedures using different groups of measurements, obtaining a total of 20 different procedures. For each of the 34-parameter procedures tried we evaluated the maximum expected error for the ISO cube and for all robot range area (8000 points each, for all 8 tool targets and with 3 kg payload), as we did in Section 7.7.1.

Here we show the procedures that gave the worst results from these expected errors. We also show the procedure that gave best results from all the 20 procedures. Finally, we also give the objective function error value obtained for each procedure.

Table 7.13 Worst expected errors from pose measurements for the 34-parameter procedures (3 kg)

Procedure	Number of measures	Max. error expected for the ISO cube area (mm)	Max. error expected for all robot range area (mm)	Objective function (mm)	Observations:
A	80	0.952	1.152	0.533	Worst of the 80 measurement procedures
B	120	0.7	0.837	0.438	Best of all the 20 procedures
C	120	0.683	0.945	0.508	Worst of the 120 measurement procedures
D	180	0.894	0.996	0.546	Worst of the 180 measurement procedures
E	200	0.837	1.211	0.555	Worst of all the 20 procedures

Looking at Table 7.13 we can see that there is some correlation between the objective function and the worst expected errors.

We validated these procedures with two types of tests: with 1000 pose measurements each in the ISO cube area at 3 kg of payload and with other 1000 measurements for all range area at 6 kg of payload.

7.7.5.1 Validation in the all range area (6 kg)

With the same analysis that we performed in Section 7.7.3, we show the position error statistics found from measuring 1000 points for the five mentioned procedures. We used the same 1000 positions for each of the procedures, using the eight targets of the tool (125 positions per tool target).

Table 7.14 Position statistics for the five 34-parameter procedures chosen (6 kg)

Procedure	μ (mm)	σ (mm)	max (mm)	μ+3σ (mm)
A	0.421	0.108	1.132	0.745
B	0.329	0.088	0.916	0.592
C	0.357	0.102	1.084	0.661
D	0.423	0.109	1.056	0.749
E	0.421	0.108	1.093	0.745

The history of all position error measurements is shown in Figure 7.17.

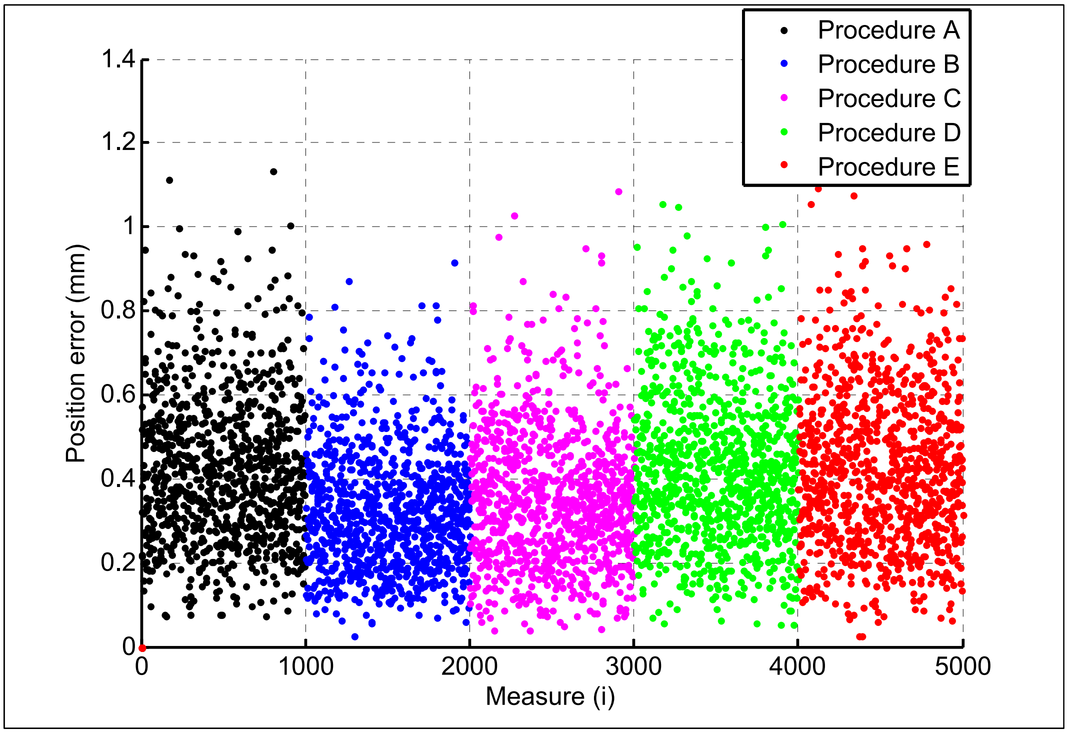


Figure 7.17 Position errors for the five 34-parameter procedures chosen (6 kg)

The results obtained are close to the expected values. For the best procedure found (procedure B in blue) the maximum error is 0.916 mm. All other procedures are around 0.1 mm worse concerning the mean values and 0.2 mm worse concerning the maximum and 99.73% CI values.

7.7.5.2 Validation from pose measurements in the ISO cube area (3 kg)

Once again, with the same analysis that we performed in Section 7.7.3, we show the position error statistics found from measuring 1000 poses for the five mentioned procedures. We used the same 1000 poses for each of the procedures, using the first three targets of the tool to generate the pose.

Table 7.15 Position statistics reduced to the 6th axis for the five 34-parameter procedures (3 kg)

Procedure	μ (mm)	σ (mm)	max (mm)	μ+3σ (mm)
A	0.174	0.038	0.351	0.289
B	0.202	0.041	0.379	0.326
C	0.220	0.040	0.401	0.341
D	0.231	0.044	0.410	0.364
E	0.371	0.045	0.610	0.505

Table 7.16 Orientation statistics for the five 34-parameter procedures (3 kg)

Procedure	μ (º)	σ (º)	max (º)	μ+3σ (º)
A	0.102	0.021	0.200	0.166
B	0.108	0.026	0.222	0.185
C	0.111	0.027	0.241	0.192
D	0.089	0.022	0.201	0.156
E	0.091	0.025	0.194	0.166

The history of all position error measurements reduced to the 6th axis and orientation errors are shown in Figure 7.18 and Figure 7.19.

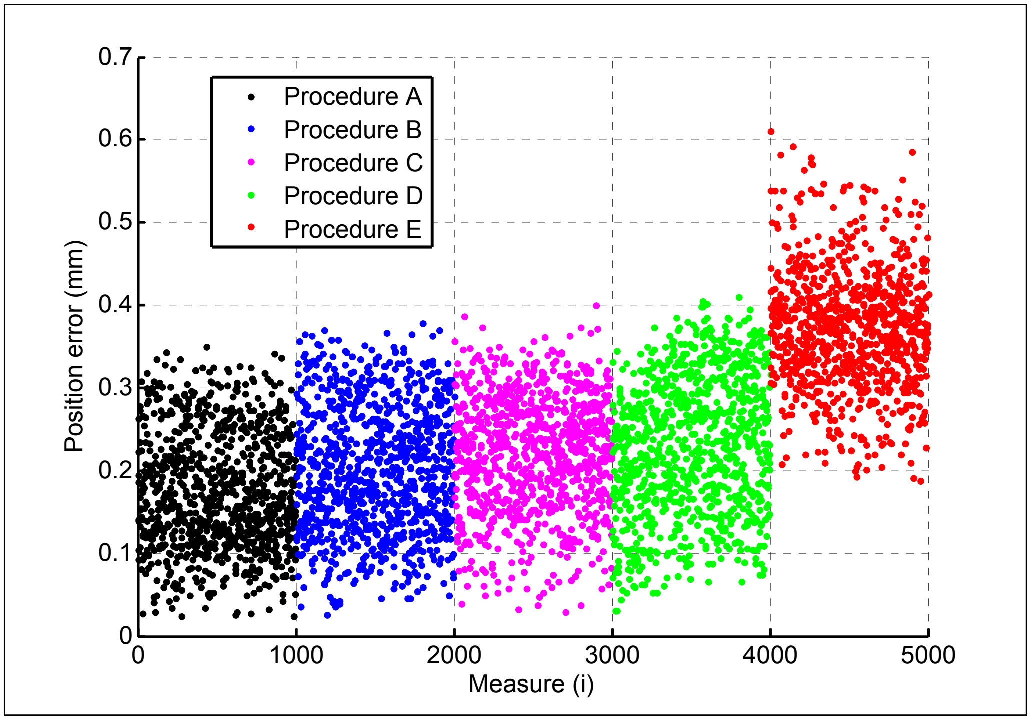


Figure 7.18 Position errors reduced to the 6th axis for the five 34-parameter procedures (3 kg)

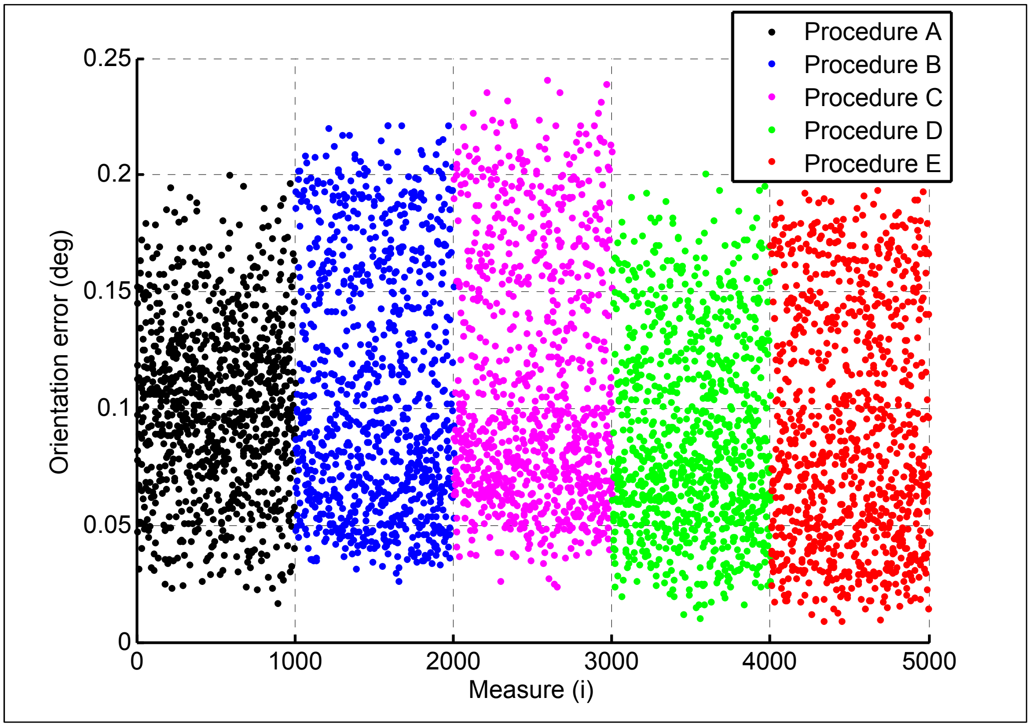


Figure 7.19 Orientation errors for the five 34-parameter procedures (3 kg)

Results obtained from pose measurements are different from the expected values as we divided the final error measurement in position errors reduced on the 6th axis and orientation errors. However, concerning these results, we find that the procedure obtained from 80 measurements (procedure A) is much better than expected. We also find that the position error for the procedure E is much worse than the other procedures (compared to results obtained in Section 7.7.5.1), but orientation errors are better.

CONCLUSION

It was demonstrated that the proposed 34-parameter model, and the procedure for identifying the 34 parameters of that model, lead to improving the accuracy of the ABB IRB 1600-6/1.45 robot in terms of mean and maximum position errors, throughout the robot workspace. Also, it was shown that the proposed iterative inverse kinematics solution scheme for the 34-parameter model works flawlessly. This solution scheme is the core of the software filter required for replacing the desired poses into so-called “fake targets”.

When stating the positioning performance of an industrial robot, it is imperative to specify the coordinates of the measurement points (with respect to the robot flange reference frame), since position errors are significantly dependent on the choice of measurement point (due to the lever effect). Ideally, this measurement point should not lie on axis 6 of the robot, and should represent the location of the tip of a commonly used tool. In this project, the eight targets (i.e., the centers of the eight SMRs) on the robot end-effector were approximately 110 mm away from the 6th axis, and about 100 mm away from the plane of the robot flange.

When stating the absolute accuracy of an industrial robot, it is also important to specify the exact location of the base reference frame used in the nominal kinematic model and in the 34-parameter model, and how this frame can be measured. In the case of our robot, we chose the base frame as explained in Section 3.2 (from kinematic calibration using the information of the first two axes). Thus, for the nominal model (i.e., before robot calibration), the mean/maximum position error values are 0.961 mm / 2.268 mm for 8000 measurements, throughout the whole workspace, and with a 3-kg payload (as shown in Table 7.1).

For the 34-parameter model (at 3-kg payload), if each of the eight targets is considered, the mean/maximum position errors are reduced to 0.329 mm / 0.703 mm, respectively, for 1000 measurements throughout the whole workspace, or 125 measurements for each of the eight targets.

If we limit the workspace to the ISO cube only, for the same 34-parameter model (at 3-kg payload), if each of the eight targets is considered, the mean/maximum position errors are further reduced to 0.277 mm / 0.647 mm, respectively, for 1000 measurements inside the ISO cube, or 125 measurements for each of the eight targets, as shown in Table 7.4.

Recall that in the identification phase, we only measure the coordinates of targets 1, 2 and 3, for validation we use all of our 8 targets. However, the position error found for targets 4 to 8 is very similar to targets 1, 2 and 3.

Regarding the orientation errors, in Section 7.7.3.5, we find that mean/maximum orientation error values, measured using the reference frame defined by targets 1, 2 and 3, are improved from 0.108º / 0.229º (using the nominal model) to 0.082º / 0.166º (using the 34-parameter model), at a 3-kg payload, in the ISO cube. This relatively small improvement is due to the fact that the objective function used in the calibration process takes into account only the position errors.

Although the robot calibration was performed with a 3-kg payload, we obtained satisfactory results for validation tests at a 6-kg payload (the maximum one), with the maximum deflection due to the extra 3 kg being only about 0.3 mm (as shown in Table 4.3). In Section 7.7.5.1, we find that for the 34-parameter model the mean/maximum position error values are 0.329 mm / 0.916 mm for 1000 measurements (125 measurements for each of the eight targets), throughout the whole workspace at a 6-kg payload.

We can see that the full analysis that we performed to our robot (computing the expected errors for the 8000 measurements and more) allowed us to find good results. However, if we do not perform this huge analysis, results turn out to be around 0.1 mm worse concerning the mean values and 0.2 mm worse concerning the maximum and 99.73% CI values (as found in Section 7.7.5.1). On the other hand, if we look at the pose results obtained (in Section 7.7.5.2), where we divided the end-effector position error in the error reduced on the 6th axis and orientation error, we find that the best procedure was not the expected one. This is because the objective function takes into account the error position at the tool target. Otherwise, if we had taken into account the reduced position error on the 6th axis in our objective function we may have obtained other expected values.

As for the various models that were tested, although the 34-parameter model led to best accuracy results, the second best model was surprisingly the simple 11-parameter model (which takes into account only the joints offsets). In both of these models, optimization is used in identifying the error parameters. In the 26-parameter model, no optimization is used and the parameters are identified directly by rotating the axes of the robot, one by one. Indeed, this so-called kinematic (or direct) calibration is even less effective than the calibration based on a simple 11-parameter model, and while the former improves a lot the mean values for the ISO cube area position error results, the maximum error values are much worse compared to the nominal model. Therefore, this direct calibration technique is hardly of any use for today's industrial robots.

We did not try the end-effector calibration explained in Section 3.4.3 because to apply this method we need to dismount the end-effector from the robot to measure it in a CMM and we realized that there is a lot of play between the robot tool flange and the end-effector (due to mechanical tolerances).

The use of constant motor slope (k_i in equation 3.37) error parameters was also tested but was not retained as it slightly improves the precision locally but gives worse results outside the studied range.

Finally, all the stiffness models that we can propose, as well as potential models that consider backlash, can be improved a lot if we know the real applied torque to each motor and the inverse kinematics can be computed in real time. The torque is proportional to the applied current. We can find this measurement somewhere in the robot control feedback. If we can access to torque measurements we can store the torque applied to each joint at each measurement. In this way it would be much easier to analyze the stiffness of the robot.

ANNEX I

POSITION REPEATABILITY OF THE ABB IRB 1600-6/1.45

The repeatability of the ABB IRB 6/1.45 robot has been measured by a Faro laser tracker ION. The laser tracker repeatability was found to be between 5 µm and 8 µm at $\mu+3\sigma$ (Annex III). The robot repeatability is measured at each of the five poses of the inclined plane defined in ISO 9283 as the plane *a*. The dimensions of this cube are 650×650×650 mm. Starting from P_1, the cycle $P_5 \rightarrow P_4 \rightarrow P_3 \rightarrow P_2 \rightarrow P_1$ is repeated 30 times taking measurements at each point of the cycle. Each valid measurement is the mean of 500 measurements for the laser tracker, taken during 500 ms (the laser tracker measures at a sample rate of 1000 Hz).

Only one target is measured at each position (orientation is not measured). But the measured target is at a distance of 120 mm from the 6th axis, so orientation repeatability affects the position repeatability. Payload was 2 kg.

In accordance to ISO 9283:1998, repeatability is defined as

$$RP_q = \pm 3S_q = \pm 3\sqrt{\frac{\sum_{i=1}^{n}\left(q_i - \overline{q}\right)^2}{n-1}}\,. \tag{I.1}$$

In the case of the laser tracker, where 3D measurements are performed, the position repeatability formula defined in Section 7.2.2 of the ISO 9283 (1998) is used.

Surprisingly, the robot repeatability at each of the five poses was found to be the same for both approach TCP speeds (100 mm/s or up to 5000 mm/s). The repeatability was found to be between 17 µm and 25 µm for poses P_1 (despite its closeness to a wrist singularity), P_4 and P_5, and between 30 µm and 37 µm for poses P_2 and P_3, in which the robot is even more stretched. Our test results slightly exceed the 20 µm repeatability quoted in the product specification that comes with our IRB1600-6/1.45 industrial robot. However, neither the measurement point, nor the five poses, nor the warm-up procedure, nor the laser tracker model and its location with respect to the robot are the same as those used in the tests performed by ABB, so no direct comparison is possible.

The five points used to test the repeatability (which correspond to the five ISO 9283 robot points for the cube a), correspond to next robot joints:

Table A I-1 Robot joints for the five positions

Position	θ_1 (º)	θ_2 (º)	θ_3 (º)	θ_4 (º)	θ_5 (º)	θ_6 (º)
P_1	0	−14	−14	0	0	0
P_2	17.690	18.084	−46.521	−33.814	33.095	29.298
P_3	−17.690	18.084	−46.521	33.814	33.095	29.298
P_4	−41.374	−24.789	47.145	−66.643	−46.052	58.108
P_5	41.373	−24.788	47.145	66.643	−46.052	58.108

For each test the mean repeatability error, maximum repeatability error and the sigma values are extracted. It is also shown in the Table the 99.73% within confidence interval which corresponds to the average error plus three times sigma. Table A I-2 shows the unidirectional repeatability error found with a linear movement so that the laser tracker never loses the target. Also, before taking the measurements, we let the robot stabilize for 4 seconds.

Table A I-2 Unidirectional repeatability statistics for the five tested points

Position	Mean error (mm)	Max error (mm)	$\mu+3\sigma$ (99.73% CI)
P_1	0.009	0.021	0.022
P_2	0.014	0.034	0.035
P_3	0.015	0.034	0.035
P_4	0.01	0.02	0.023
P_5	0.012	0.027	0.032

If we repeat the test but we change the cyclical order to a random order, without linear movement (the laser tracker looses the target most of times) and without waiting any time for the robot to stabilize (actually the robot is stopped for at least 0.5 seconds before taking the measurements, which is the time for the laser tracker to detect the presence of the target). We obtain the values shown in Table A I-3.

Table A I-3 Repeatability statistics for the five tested points in worse conditions

Position	Mean error (mm)	Max. error (mm)	$\mu+3\sigma$ (99.73% CI)
P_1	0.052	0.112	0.162
P_2	0.028	0.049	0.066
P_3	0.045	0.105	0.115
P_4	0.022	0.034	0.043
P_5	0.023	0.048	0.052

ANNEX II

FARO ION LASER TRACKER ACCURACY

The accuracy of the laser tracker has been tested in a robot cell area with an INVAR bar of nominal distance 1050.688 mm. The measured targets are 1.5” SMR laser tracker spherical targets.

During the test the temperature was between 23.1 and 23.9 ºC. The pressure was between 767.8 and 768.2 mm Hg and the relative humidity between 21.8 % and 23.2 %.

The robot has been moved to 12 different positions. At each position the robot stops and takes 30 distance measurements. For each measurement the laser tracker makes the average of 500 measurements at a sample rate of 1000 Hz, so 500 ms are needed for each measurement. The laser tracker starts measuring 5 seconds after the robot reaches the desired position, after two minutes the robot activates the brakes (same state as an emergency stop state) for energy saving. That happens exactly after 15 or 16 distance measurements, the robot is supposed to be more stable in this state.

Position measurements are taken cyclically from one side of the bar to the other side of the bar, for each pair a distance measurement is calculated from these two points. The distance error is calculated as the measured error minus the nominal error. Distance error measurements are sequentially displayed, so the 30 measurements for the first robot position are sorted from 1 to 30, the 30 measurements for the second robot position are sorted from 31 to 60, etc.

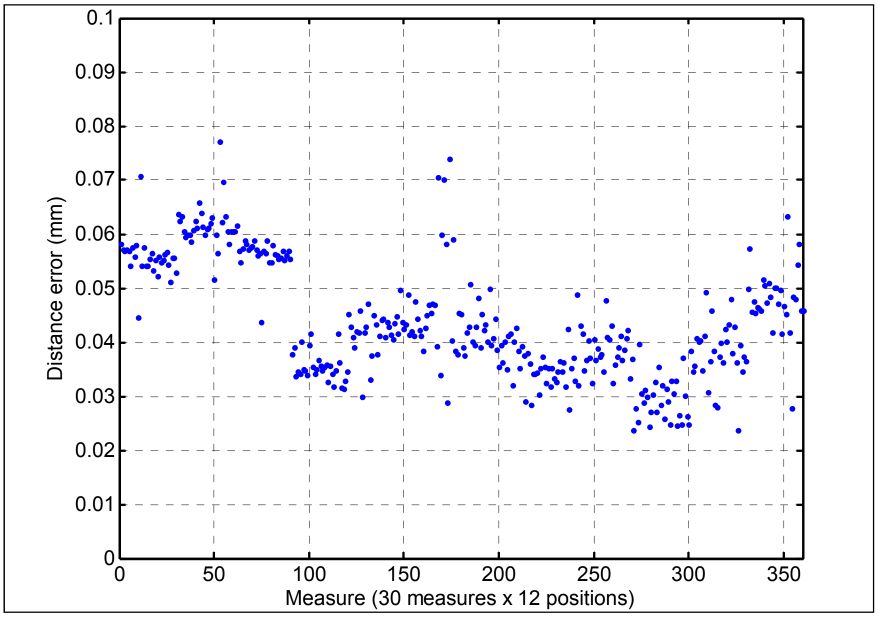


Figure A II-1 Distance error for all 360 distance measures

As we can see in this Figure, the effect of the brakes is very small (comparing the first 15 first points versus the last 15 points of each cycle test). Therefore, the stability that the robot brakes can add is not important at this level of precision.

Table A II-1: Robot joint values for each position of the test

Test	θ_1 (º)	θ_2 (º)	θ_3 (º)	θ_4 (º)	θ_5 (º)	θ_6 (º)
1	−25	0	0	0	0	0
2	−25	0	0	0	45	0
3	−25	0	0	0	−45	0
4	0	0	−90	0	0	−45
5	−25	0	−90	0	45	−10
6	−45	0	−90	0	−45	10
7	0	−15	−60	−30	75	5
8	0	15	−120	−30	−75	5
9	110	0	0	0	0	180
10	110	0	0	0	45	180
11	110	0	0	0	−45	180
12	80	0	0	0	−90	220

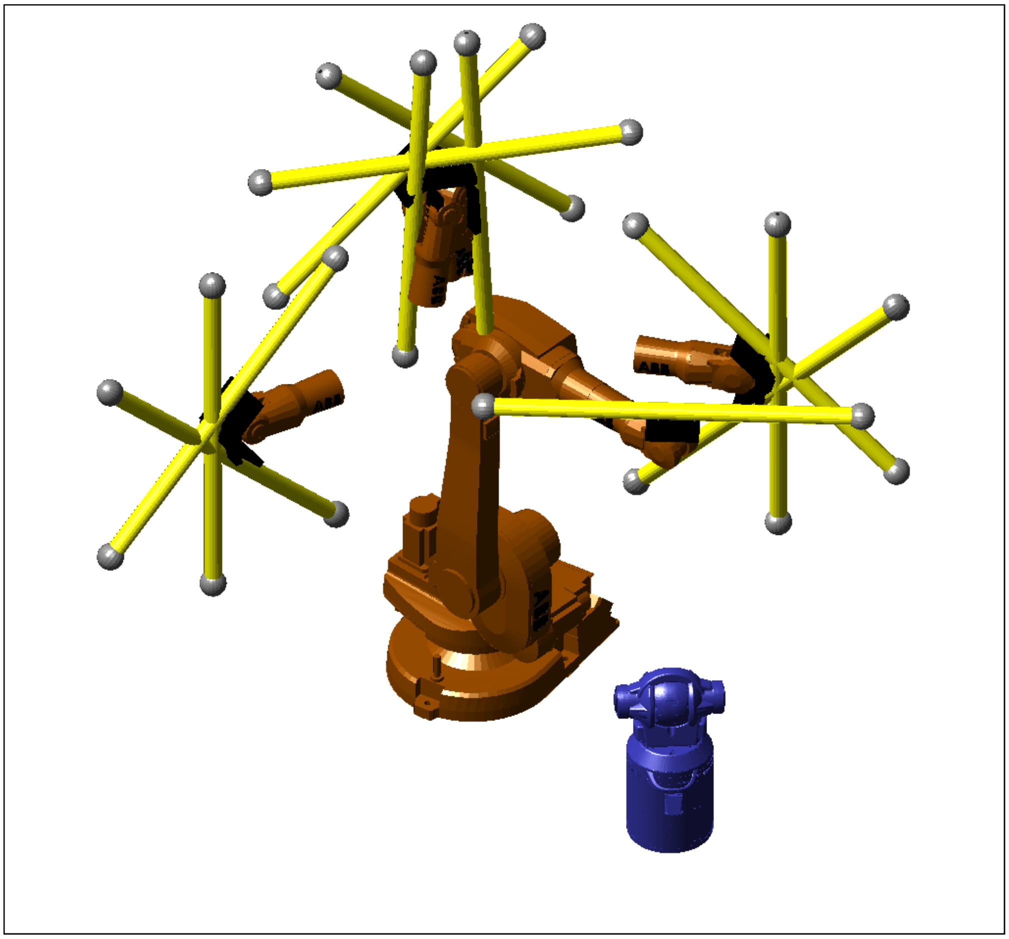


Figure A II-2: Representation of the 12 position tests for measuring the INVAR distance

ANNEX III

FARO ION LASER TRACKER REPEATABILITY

To prove the laser tracker repeatability we used similar tests to the ISO 9283:1998 that are applied to robots. One target is measured 50 times and the error of each measurement is considered as the distance between each measurement and the mean of all 50 measurements. Each valid measurement is the mean of 500 measurements for the laser tracker, taken during 500 ms (the laser tracker measures at a sample rate of 1000 Hz).

Between every measurement the laser tracker is moved to a random position where there is no target, after it is moved back to a close position (but not exact) from the known target of the first measurement. This position is close enough so that the laser tracker is capable of finding the target without executing its search algorithm. The Faro laser tracker ION is capable of doing that always that the guessed position is not farther than 1.5-2 mm from the real target.

The test is repeated five times for five different points in a robot cell working space. The robot is stopped in the "emergency-stop" mode during the test. These five positions correspond to the five ISO 9283 robot points (plane a), which correspond to next points referenced on the laser tracker coordinate system (given values are the mean of the 50 measurements for each point):

Table A III-1 Mean laser tracker coordinates for each test

Position	X (mm)	Y (mm)	Z (mm)
P_1	1568.937	753.352	−234.817
P_2	1184.504	700.874	31.782
P_3	1512.934	1135.976	36.515
P_4	1952.791	806.731	−502.204
P_5	1625.478	372.956	−505.890

For each test the mean error, maximum error and sigma values are extracted. It is also shown in the Table the 99.73% within confidence interval which corresponds to the average plus three times sigma.

Table A III-2 Statistics for the five repeatability tests

Position	Mean error (mm)	Max. error (mm)	σ (mm)	μ+3σ (99.73% CI)
P_1	0.002	0.004	0.001	0.005
P_2	0.001	0.006	0.001	0.004
P_3	0.002	0.005	0.001	0.005
P_4	0.003	0.007	0.002	0.008
P_5	0.002	0.007	0.001	0.006

Figure A III-1 shows the 50 error measurements for each of the five tests.

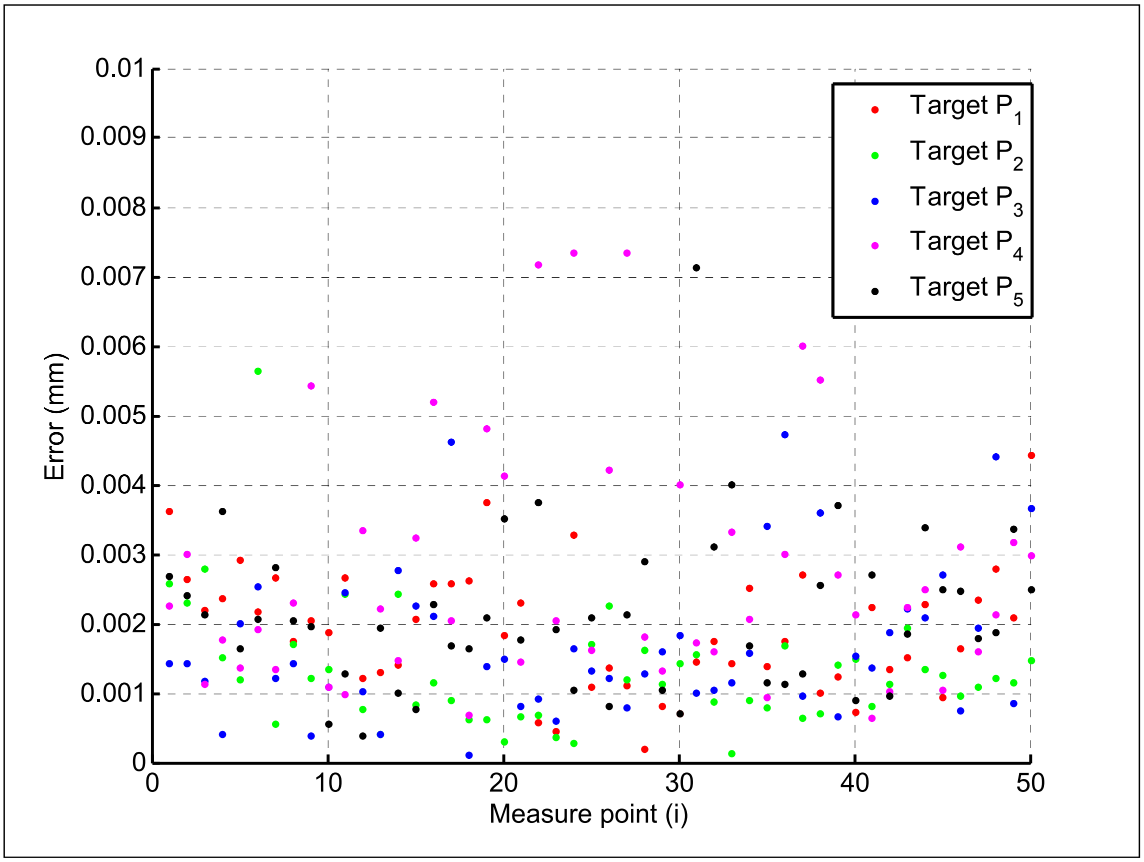


Figure A III-1 Measurement error for each of the five tests

ANNEX IV

FARO ION LASER TRACKER 24-HOUR TEST

To prove the laser tracker measurement stability through time we took measurements during 24 hours to a fixed target. Next figures show the measurement noise of taking measurements to a fixed target in space (a 1.5” SMR target, mounted on a robot tool). During 26 hours and 38 minutes a measurement has been taken every 17 seconds (exact cycle time varies from 15 to 20 seconds), making a total of 5871 measurements where we saved time, position, temperature, pressure and humidity. Each saved position measurement is the mean of 500 measurements for the laser tracker, taken during 500 ms (the laser tracker measures at a sample rate of 1000 Hz). Test started the 12th of December 2010 at 15h 50min and 43 seconds and ended the next day at 18h 28 min and 4 seconds. The test was performed at local A-3569 in the École de Technologie Supérieure, Montréal.

Between measurements the laser tracker is moved to a random position where there is no target, after it is moved back to a close position (but not exact) from the known target of the first measurement. This position is close enough so that the laser tracker is capable of finding the target without executing its search algorithm. The laser tracker Faro ION is capable of doing that always that the guessed position is not farther than 1.5-2mm from the real target.

The position error is calculated as the distance of each measurement to the mean of all measurements. As seen in the Figures there is no correlation between position error and temperature, pressure or humidity.

The mean position coordinates are $\{X=1586.937;\ Y=517.974;\ Z=-216.203\}$ mm referenced to laser tracker coordinate system. The mean and maximum error values are 0.012 mm and 0.038 mm respectively. Standard deviation of error is 0.005 mm.

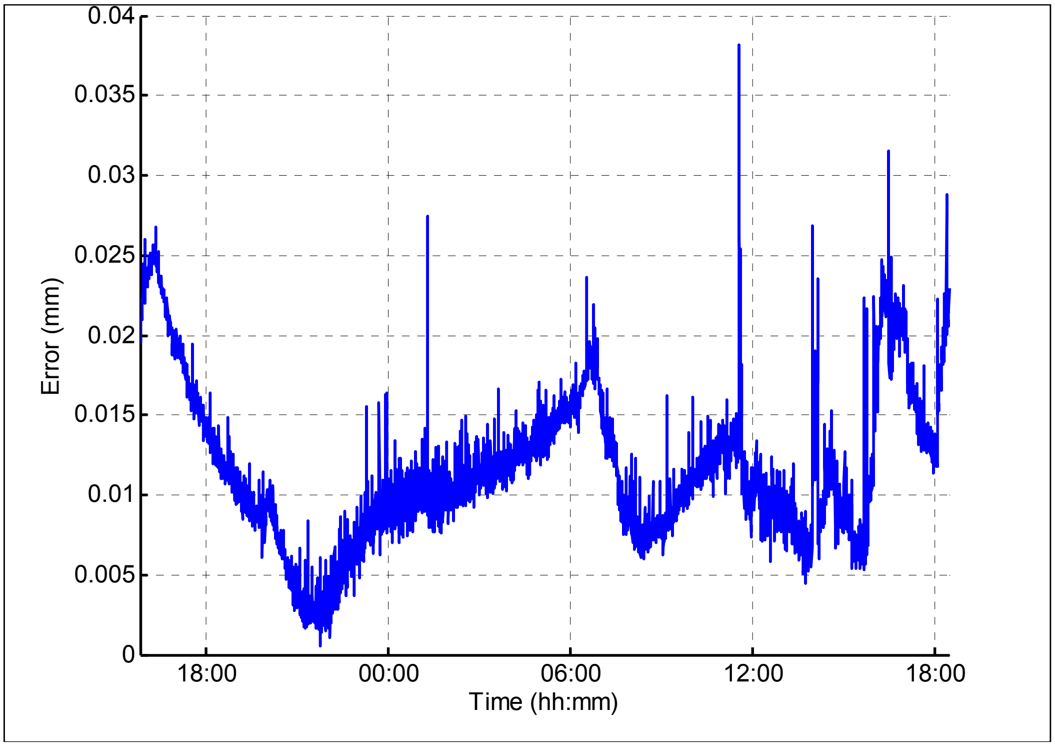


Figure A IV-1 Evolution of error

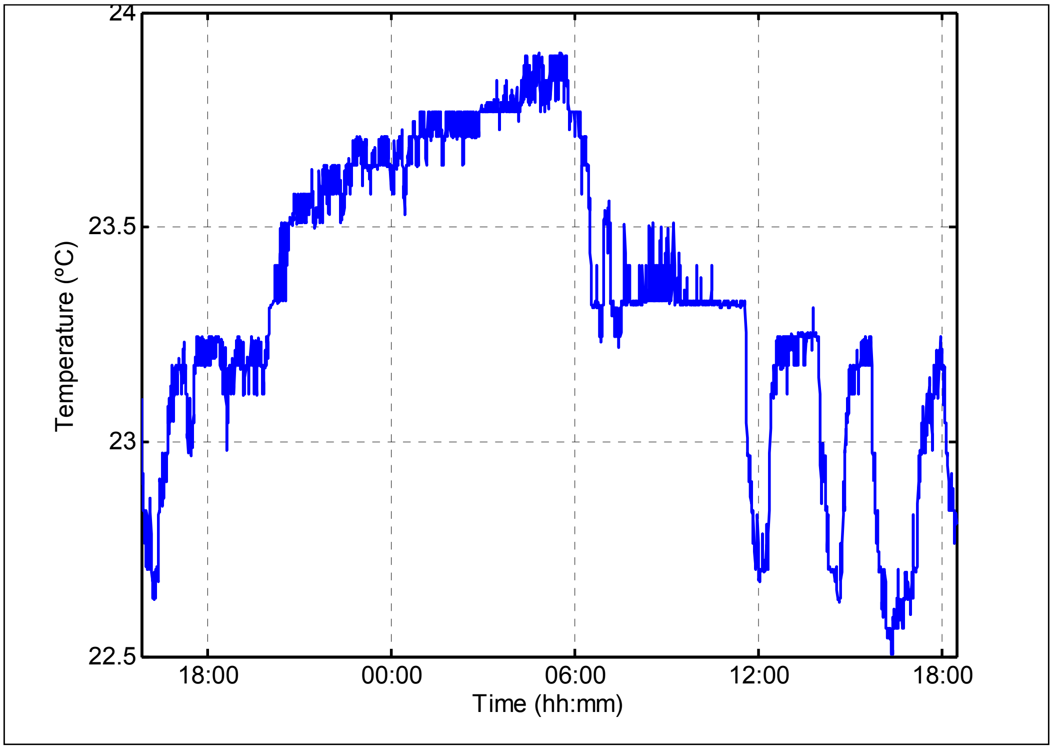


Figure A IV-2 Evolution of temperature

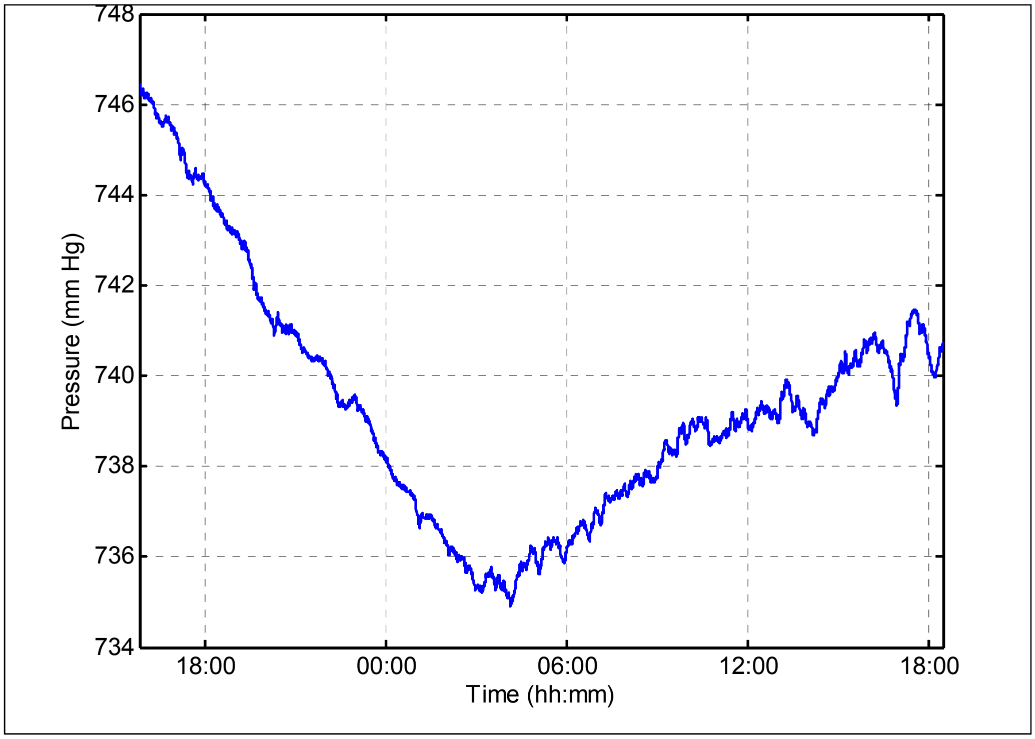


Figure A IV-3 Evolution of pressure

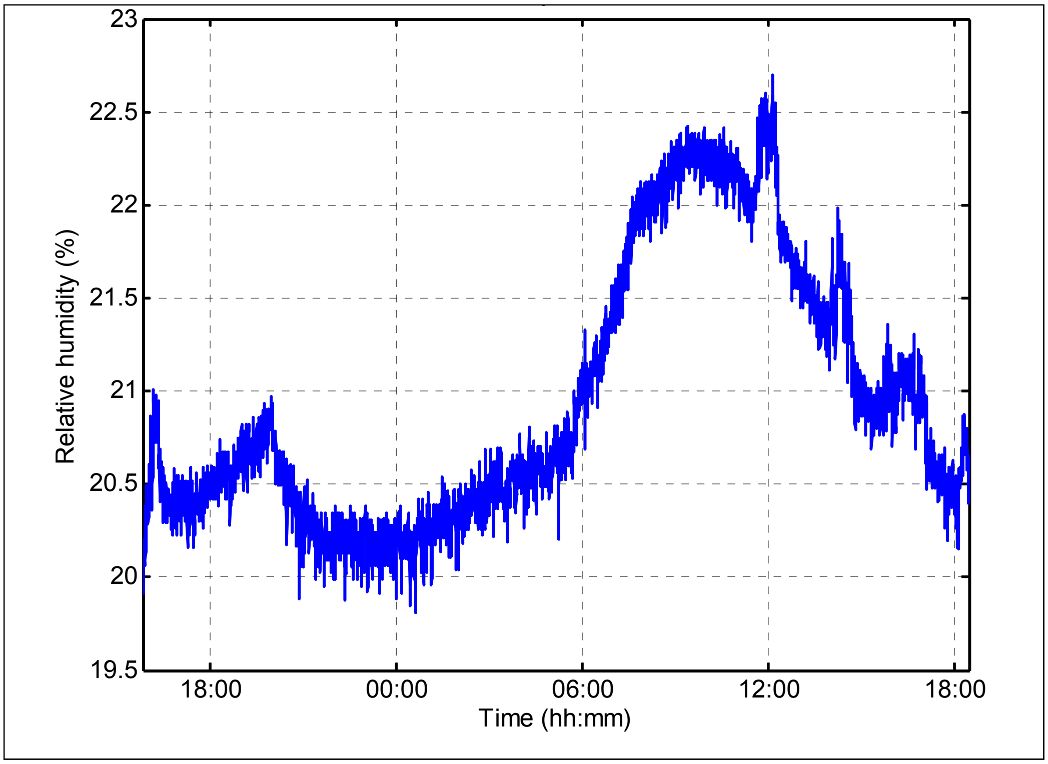


Figure A IV-4 Evolution of humidity

Matlab code used for tests

```
% Taking XYZ measurement of the present target
PREF = FARO_Take_Measure;
% Opening file to store data
fid = fopen('HISTORY.txt','w');
for i=1:n
    % Moving laser tracker at a random position
    randalpha = (rand)*5;
    randbeta = (rand*2-1)*5;
    point = [1000 0 0 1]';
    movepoint = rotx(randbeta*pi/180)*roty(randalpha*pi/180)*point;
    randmove = movepoint(1:3)';
    FARO_Move_XYZ(randmove);
    % Moving laser tracker 1.5mm close from the real target
    randvect = (rand(1,3)*2-1)*(1.5/sqrt(3));
    FARO_Move_XYZ(PREF+randvect);
    % Taking measurement
    p=FARO_Take_Measure;
    % Taking environment measurements
    temp = FARO_Weather_Conditions;
    % Taking computer time
    timei = clock;
    % Storing data to file
    fprintf(fid, '%4.5f %3.0f %3.0f %3.0f %3.0f %3.3f %3.5f %3.5f %3.5f
%3.5f %3.5f %3.5f\n', timei, temp.temperature,temp.pressure,temp.humidity,
p);
    pause(10);
end
% Closing file
fclose(fid);
```

ANNEX V

ABB IRB 1600-6/1.45 POSITION STABILIZATION TIME

To measure the ABB robot stabilization time the robot was moved linearly from the ISO point P_2 to P_1 three times at maximal speed as described on ISO 9283 (Section 7.4). Payload was 2.5 kg.

The error is calculated as the distance between every point and a “final-target” point. This final-target point is considered as the mean of 100 measurements after 10 seconds of stabilization.

Measurements are taken as fast as possible from the Faro ION laser tracker. The laser tracker spends 1 ms to take each measurement but the average time for treating the data and transferring this measurement to Matlab is around 40 ms.

The laser tracker starts measuring as the robot starts moving from point P_2 to P_1 and stops measuring after 20 seconds that the robot reached the point P_1. The time to execute this path is known (but can have small variations). The zero-time (used to plot graphics) is considered as the start moving instant plus the mean path time of 10 paths.

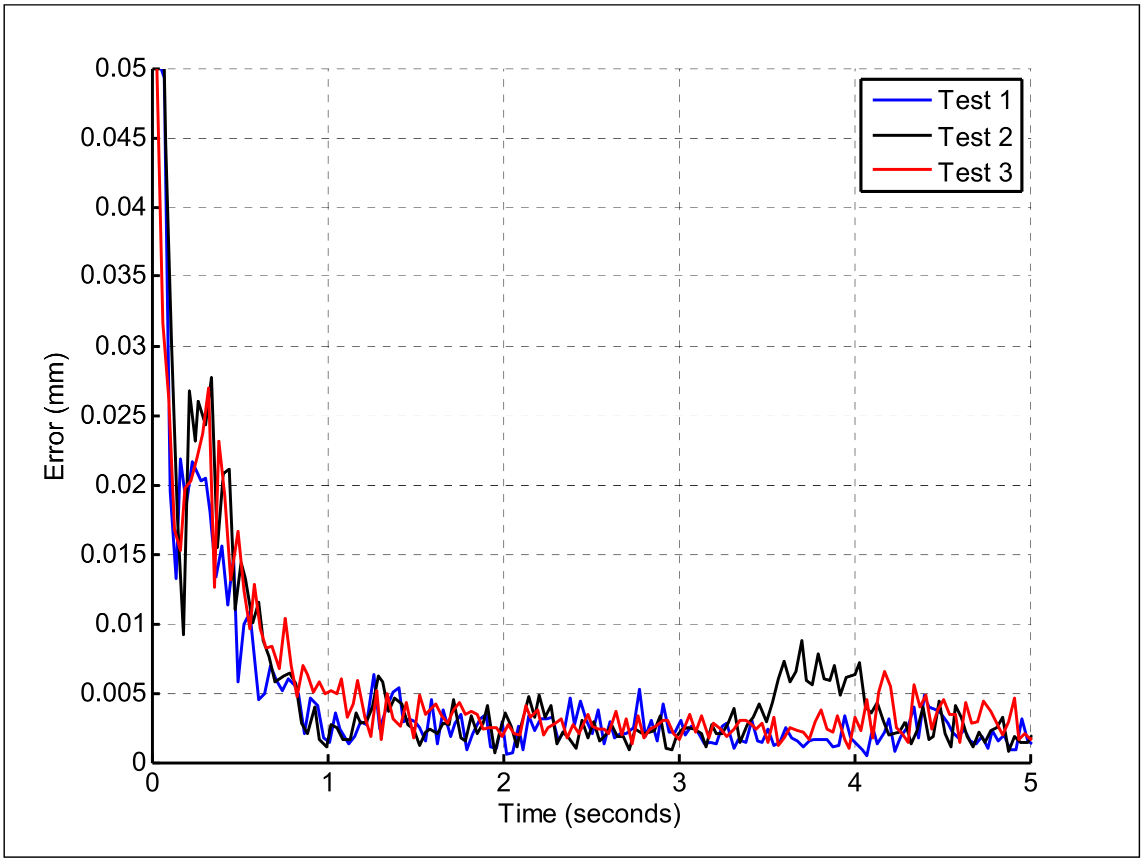


Figure A V-1 Stabilization error

ANNEX VI

CIRCLE PATH ACCURACY TESTS WITH A TELESCOPIC BALLBAR

Three circular path tests have been performed to an ABB IRB 1600-1.45/6 robot with a telescopic ballbar to prove the improvement of precision. The measurement instrument is a QC20-W telescopic ballbar by Renishaw with Bluetooth wireless technology. It consists of a wireless telescoping ballbar, ballbar extensions, a tool cup, two measuring balls and a pivot assembly. The tool cup is mounted on the robot end-effector. The base of the pivot assembly is magnetic and is solidly attached to a heavy steel table. The ballbar sensor accuracy (at 20°C) is ±0.5 μm and the measuring range is merely ±1.0 mm (which is probably too small for large non-calibrated industrial robots).

The same end-effector, weighing approximately 2 kg, was used in all ballbar tests. Circular tests were performed in clockwise (CW) and counterclockwise (CCW) direction at radius of 150 mm, 300 mm and 400 mm, and at the constant feedrate of 50 mm/s (*i.e.*, TCP linear speeds). The coordinates of the measurement point (*i.e.*, the center of the tool cup) with respect to the flange reference frame is approximately {0 mm, 65 mm, 149 mm}.

Before starting a telescoping ballbar test, the robot was warmed up by repeating the actual circular trajectory during one hour (from cold start).

Graphics and tables are shown for each of the three tests. The position of the center of these three tests describing the end-effector with respect to the *robot base frame* (*i.e.*, *wobj0*) are approximately {808.234 mm, −56.311 mm, 1011.205 mm} which correspond to {−3.984º, 5.733483º, 4.519º, −0.017º, 79.885º, −273.988º} in the robot joint space.

For the first test the circle radius is 300 mm as shown in Figure A VI−3. For the second test the circle radius is 400 mm and the tool is inclined 8º as shown in Figure A VI−6. For the third test the circle radius is 150 mm and the tool is inclined 45º as shown in Figure A VI−9.

When the path of the circle has to be calculated we must choose a number of arcs to discretize the circle for the improved path. For the first test the circle is divided in 6 arcs. For the third test the circle is divided in 32 arcs (which correspond to a division each 30 mm of path or less). For the second test a comparison is made between 6 arcs and 84 arcs (which correspond to a division each 30 mm of path or less).

For the nominal path (calculated with nominal kinematics) the discretization of the path does not affect the final result so the full 360 º circle path is divided in two arcs of 180º.

As we can see in next results, the error of the improved model gives better results. Also, as shown in Figure A IV−4, the effect of dividing the circle in smaller arcs gives better results although the final result is not so much better.

We can also see that always that there is a local maximum or minimum for one joint axis (which means that the joint changes direction) a perturbation is produced in the circle path, probably due to backlash.

Test 1: 300 mm radius

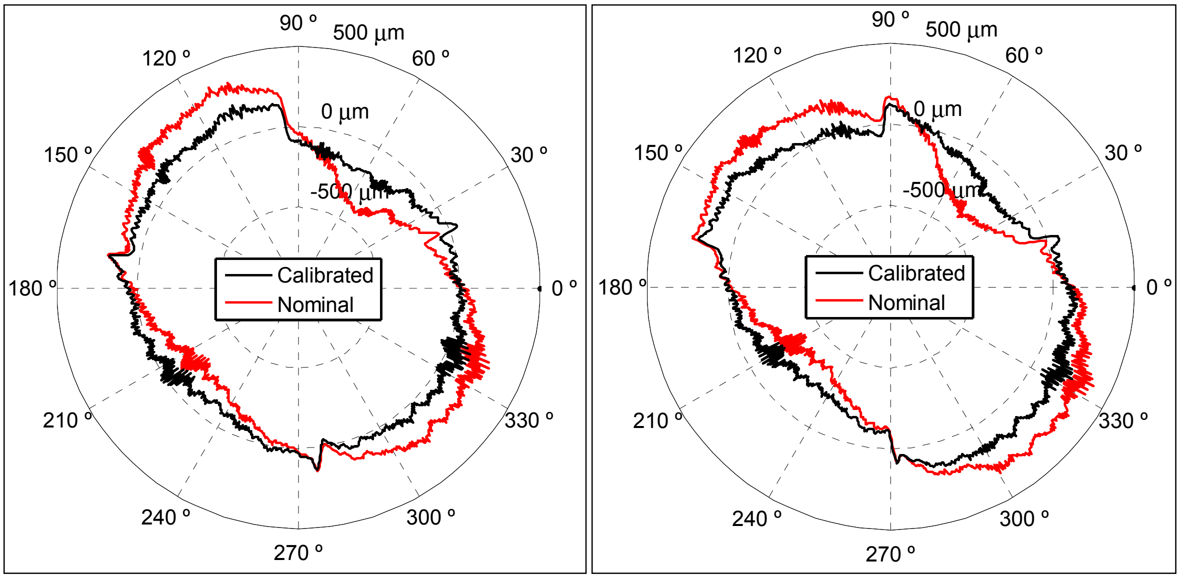


Figure A VI-1 300 mm circular path (CW left, CCW right)

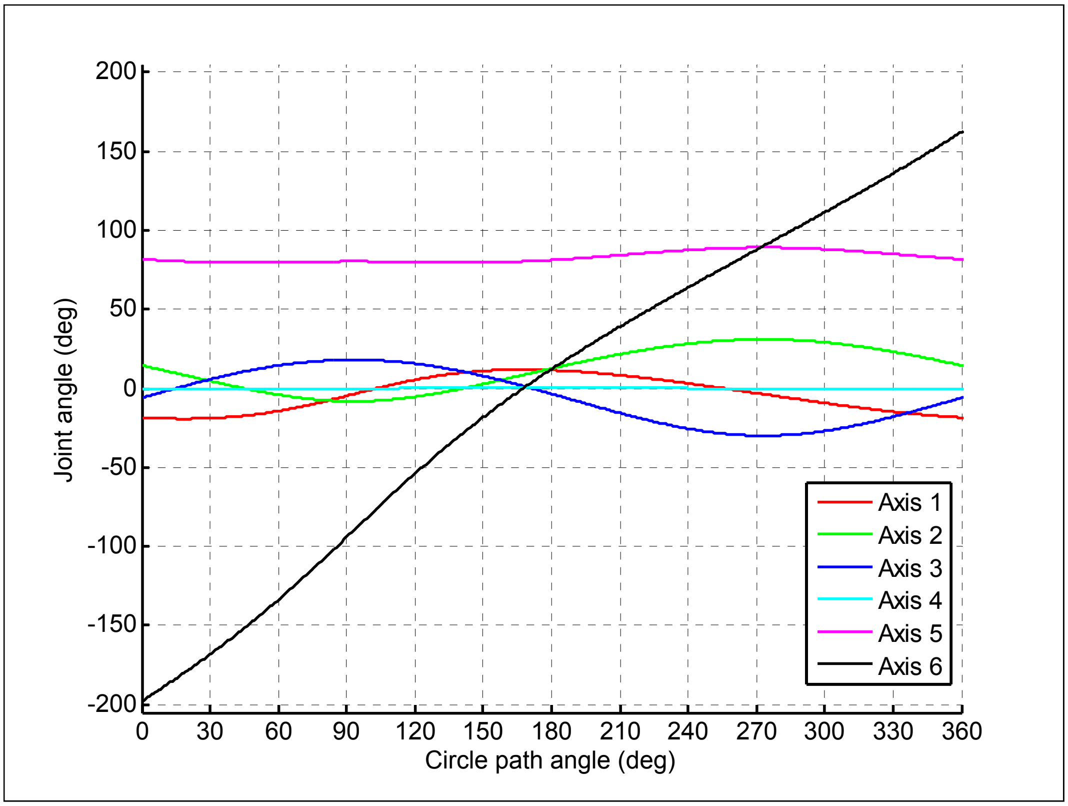


Figure A VI-2 Joint history simulation for the 300 mm-circle path

Table A VI-1 Local minimum and maximum values for all joints

id	Circle path angle (º)	Joint value (º)	Joint (axis number)	Local min/max
1	18.7	−19.2	1	min
2	19.3	-0.1	4	min
3	44.7	79.6	5	min
4	93.3	−8.2	2	min
5	93.3	18.1	3	max
6	93.3	80.3	5	max
7	141.3	79.6	5	min
8	167.3	11.8	1	max
9	167.3	0.0	4	max
10	273.3	31.0	2	max
11	273.3	−29.9	3	min
12	273.3	89.1	5	max

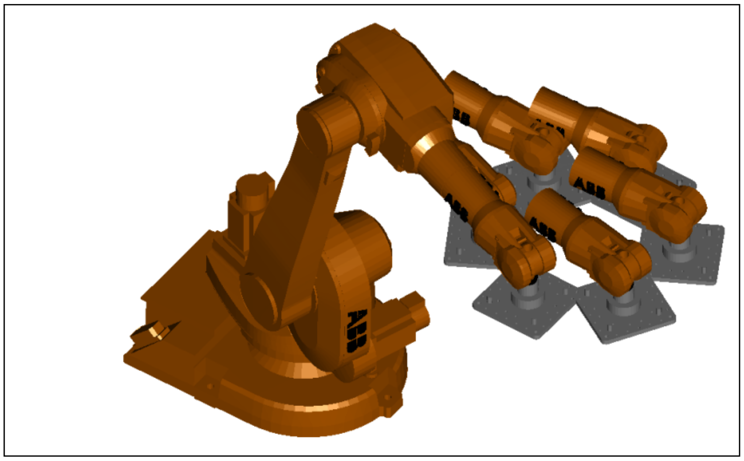


Figure A VI-3 Representation of 300mm circle path

Test 2: 400 mm circle radius / 8 degrees

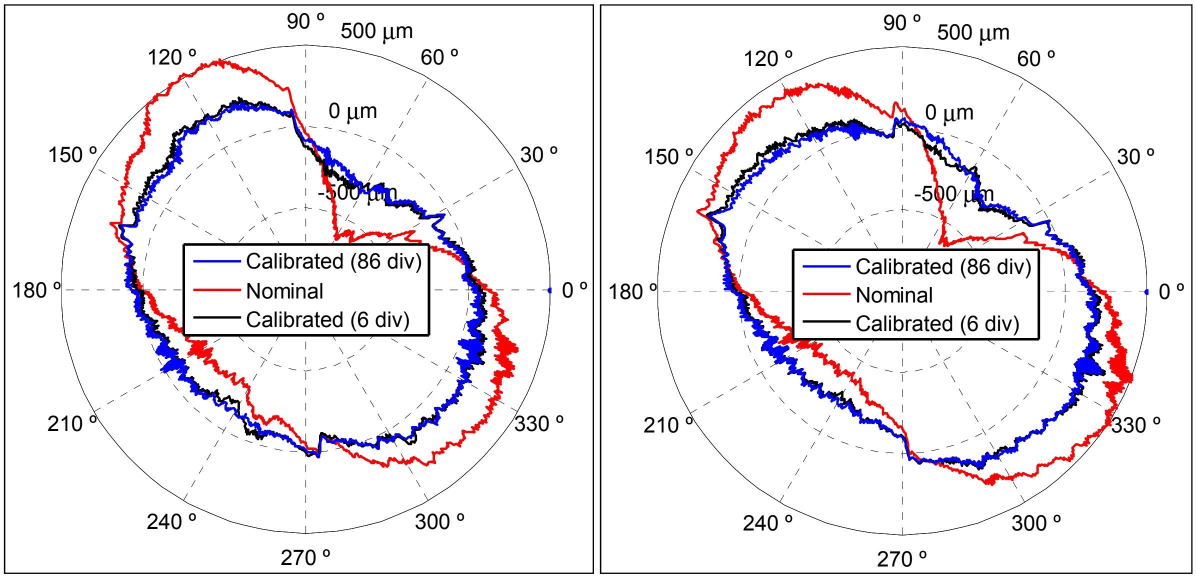


Figure A VI-4: 400 mm circular path (CW left, CCW right)

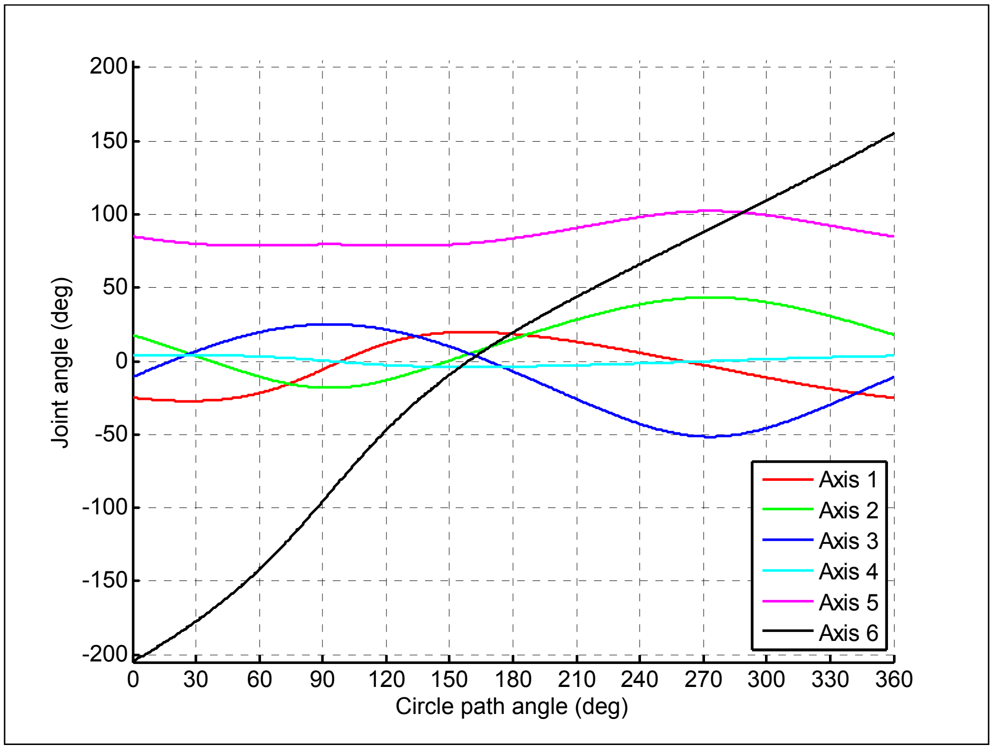


Figure A VI-5 Joint history simulation for the 400 mm-circle path

Table A VI-2 Local minimum and maximum values for all joints

id	Circle path angle (º)	Joint value (º)	Joint (axis number)	Local min/max
1	26.5	−27.2	1	min
2	28.0	4.0	4	max
3	55.0	78.6	5	min
4	93.0	−18.3	2	min
5	93.0	25.0	3	max
6	93.0	79.4	5	max
7	131.5	78.6	5	min
8	158.5	−4.0	4	min
9	159.5	19.8	1	max
10	273.0	43.4	2	max
11	273.0	−51.5	3	min
12	273.0	102.2	5	max

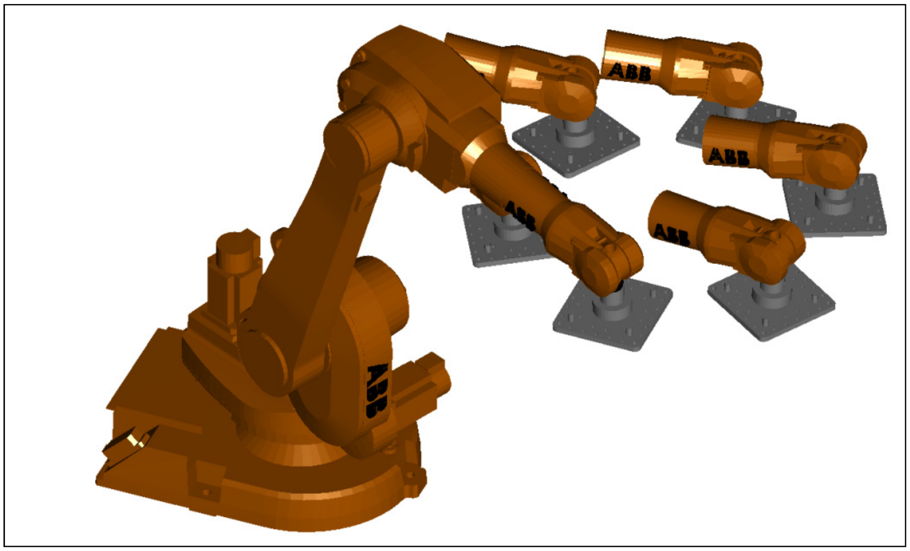


Figure A VI-6 Representation of 400mm circle path

Test 3: 150 mm circle radius / 45 degrees

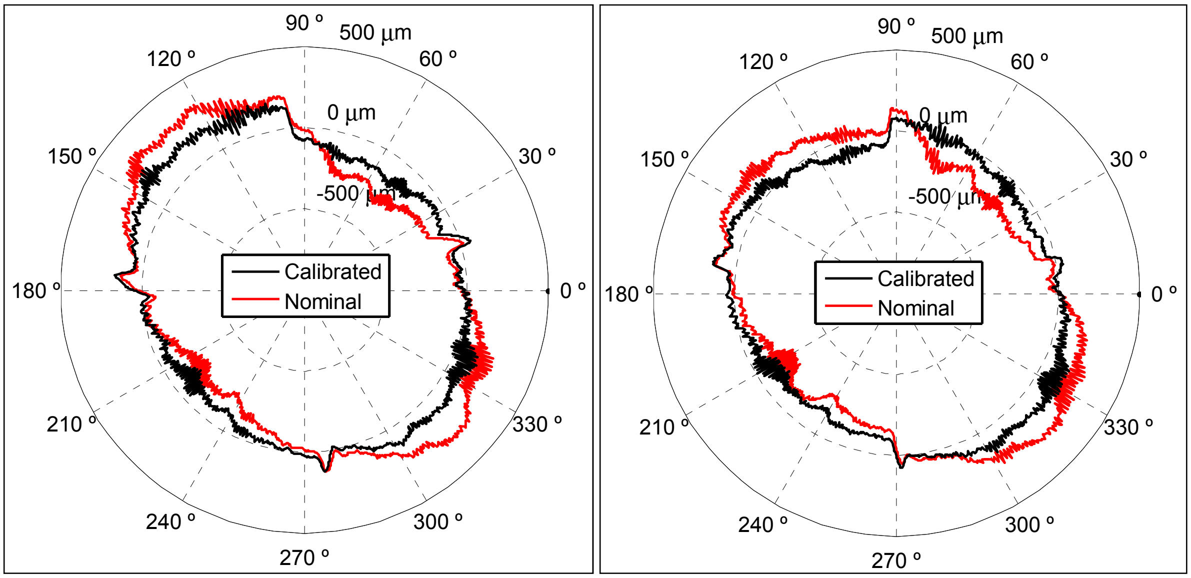


Figure A VI-7 150 mm circular path (CW left, CCW right)

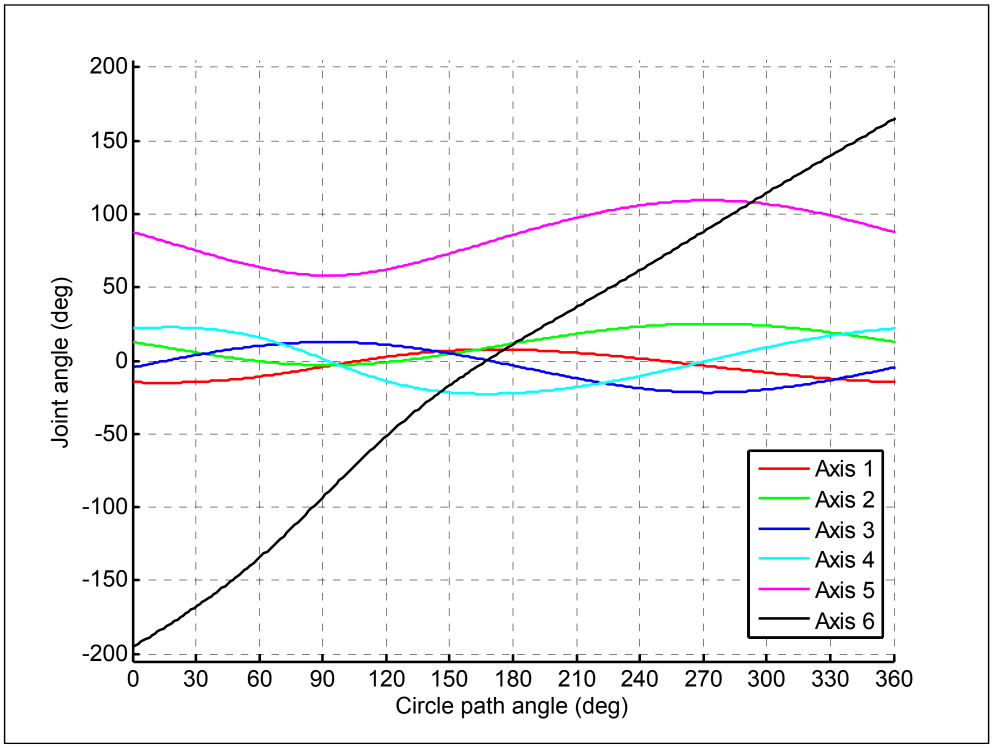


Figure A VI-8 Joint history simulation for the 150 mm-circle path

Table A VI-3 Local minimum and maximum values for all joints

id	Circle path angle (º)	Joint value (º)	Joint (axis number)	Local min/max
1	13.2	−15.0	1	min
2	17.2	22.7	4	max
3	92.6	−3.0	2	min
4	92.6	12.7	3	max
5	92.6	58.0	5	min
6	166.8	−22.8	4	min
7	170.7	7.6	1	max
8	272.6	25.2	2	max
9	272.6	−21.8	3	min
10	272.6	109.3	5	max

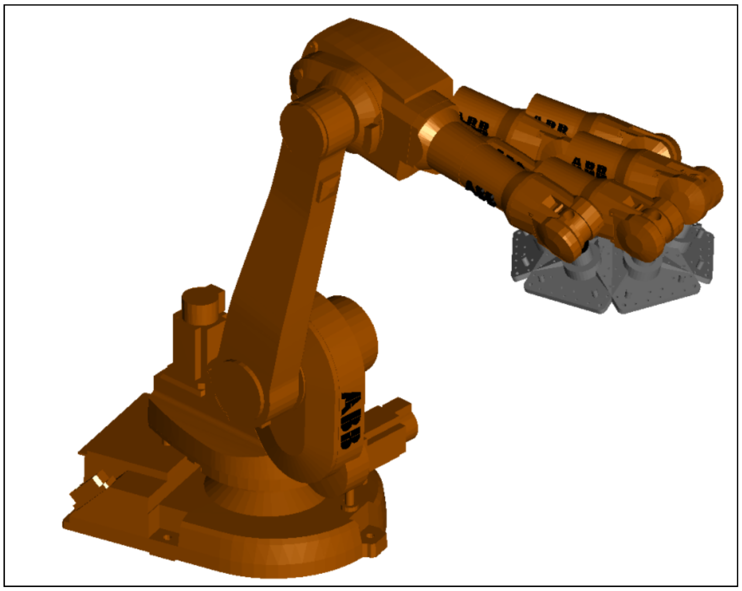


Figure A VI-9 Representation of 150 mm circle path

REFERENCES

Abderrahim, M., A. Khamis, S. Garrido and L. Moreno. 2007. "Accuracy and calibration issues of industrial manipulators", *Industrial Robotics: Programming, Simulation and Applications*, p. 131.

Andrew Liou, Y.H., P.P. Lin, R.R. Lindeke and H.D. Chiang. 1993. "Tolerance specification of robot kinematic parameters using an experimental design technique--the Taguchi method", *Robotics and Computer-Integrated Manufacturing,* vol. 10, n^{o} 3, p. 199-207.

Angeles, J. 1985. "On the numerical solution of the inverse kinematic problem", *The International Journal of Robotics Research,* vol. 4, n^{o} 2, p. 21-37.

Bai, Ying, Hanqi Zhuang and Z.S. Roth. 2003. "Experiment study of PUMA robot calibration using a laser tracking system", In *Soft Computing in Industrial Applications, 2003. SMCia/03. Proceedings of the 2003 IEEE International Workshop on* (23-25 June 2003). p. 139-144.

Barker, L.K. 1983. "Vector-algebra approach to extract Denavit-Hartenberg parameters of assembled robot arms", *NASA Tech. Paper 2191.*

Beyer, L., and J. Wulfsberg. 2004. "Practical robot calibration with ROSY", *Robotica,* vol. 22, n^{o} 05, p. 505-512.

Bonnans, JF, and C Lemaréchal. 2006. *Numerical optimization: theoretical and practical aspects*. Springer-Verlag New York Inc, 493 p.

Caenen, J.L., and J.C. Angue. 1990. "Identification of geometric and nongeometric parameters of robots", In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on* (13-18 May 1990). Vol. 2, p. 1032-1037.

Chen, N., and G.A. Parker. 1994. "Inverse kinematic solution to a calibrated puma 560 industrial robot", *Control Engineering Practice,* vol. 2, n^{o} 2, p. 239-245.

Craig, J.J. 1986. *Introduction to robotics: mechanics and control*, 3rd edition. Addison-Wesley, 400 p.

De Smet, P. 2001. "Method for calibration of a robot inspection system".

Denavit, J., and R.S. Hartenberg. 1955. "A kinematic notation for lower-pair mechanisms based on matrices", *Trans. ASME J. Appl. Mech,* vol. 22, n^{o} 1, p. 215-221.

Driels, M.R., W. Swayze and S. Potter. 1993. "Full-pose calibration of a robot manipulator using a coordinate-measuring machine", *The International Journal of Advanced Manufacturing Technology,* vol. 8, nº 1, p. 34-41.

Gander, W., G.H. Golub and R. Strebel. 1994. "Least-squares fitting of circles and ellipses", *BIT Numerical Mathematics,* vol. 34, nº 4, p. 558-578.

Goldenberg, A.A., J.A. Apkarian and H.W. Smith. 1987. "A new approach to kinematic control of robot manipulators", *Journal of Dynamic Systems, Measurement, and Control,* vol. 109, p. 97.

Goldenberg, A.A., and D.L. Lawrence. 1985. "A generalized solution to the inverse kinematics of robotic manipulators", *Journal of Dynamic Systems, Measurement, and Control,* vol. 107, p. 103.

Greenway, B. 2000. "Robot accuracy", *Industrial Robot: An International Journal,* vol. 27, nº 4, p. 257-265.

Hayati, S., and M. Mirmirani. 1985. "Improving the absolute positioning accuracy of robot manipulators", *Journal of Robotic Systems,* vol. 2, nº 4, p. 397-413.

Hoppe, W.C. 2008. "Method and System to Provide Improved Accuracies in Multi-Jointed Robots Through Kinematic Robot Model Parameters Determination". WIPO Patent No. WO/2006/086021.

Houde, Geneviève. 2006. "A closed-loop method for the geometric calibration of serial robots", Montréal, McGill University, 80 p.

Ikits, M., and J.M. Hollerbach. 1997. "Kinematic calibration using a plane constraint", In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on* (20-25 Apr 1997). Vol. 4, p. 3191-3196.

Jang, J.H., S.H. Kim and Y.K. Kwak. 2001. "Calibration of geometric and non-geometric errors of an industrial robot", *Robotica,* vol. 19, nº 03, p. 311-321.

Judd, R.P., and A.B. Knasinski. 2002. "A technique to calibrate industrial robots with experimental verification", *Robotics and Automation, IEEE Transactions on,* vol. 6, nº 1, p. 20-30.

Karan, B., and M. Vukobratovic. 1994. "Calibration and accuracy of manipulation robot models--An overview", *Mechanism and Machine Theory,* vol. 29, nº 3, p. 479-500.

Khalil, W., and S. Besnard. 2002. "Geometric calibration of robots with flexible joints and links", *Journal of Intelligent and Robotic systems,* vol. 34, nº 4, p. 357-379.

Knoll, A., and P. Kovacs. 2001. "Method and device for the improvement of the pose accuracy of effectors on mechanisms and for the measurement of objects in a workspace". US Patent No. 6,529,852 filed on May 14, 2001, issued on March 4, 2003.

Lightcap, C., S. Hamner, T. Schmitz and S. Banks. 2008. "Improved positioning accuracy of the PA10-6CE robot with geometric and flexibility calibration", *Robotics, IEEE Transactions on,* vol. 24, n° 2, p. 452-456.

Meggiolaro, M.A., S. Dubowsky and C. Mavroidis. 2005. "Geometric and elastic error calibration of a high accuracy patient positioning system", *Mechanism and Machine Theory,* vol. 40, n° 4, p. 415-427.

Meggiolaro, M.A., G. Scriffignano and S. Dubowsky. 2000. "Manipulator calibration using a single endpoint contact constraint", In *Proceedings of DETC2000: 2000 ASME Design Engineering Technical Conference* (2000). p. 1-8. Baltimore.

Meng, Y., and H. Zhuang. 2001. "Self-calibration of camera-equipped robot manipulators", *The International Journal of Robotics Research,* vol. 20, n° 11, p. 909.

Mooring, B, M Driels and Z Roth. 1991. *Fundamentals of manipulator calibration*. John Wiley & Sons, Inc. New York, NY, USA, 329 p.

Mooring, B.W., and S.S. Padavala. 1989. "The effect of kinematic model complexity on manipulator accuracy", In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on* (14-19 May 1989). Vol. 1, p. 593-598.

Motta, J.M.S.T., and R.S. McMaster. 1999. "Modeling, optimizing and simulating robot calibration with accuracy improvement", *Journal of the Brazilian Society of Mechanical Sciences,* vol. 21, p. 384-401.

Nelder, J.A., and R. Mead. 1965. "A simplex method for function minimization", *The computer journal,* vol. 7, n° 4, p. 308.

Newman, W.S., C.E. Birkhimer, R.J. Horning and A.T. Wilkey. 2000. "Calibration of a Motoman P8 robot based on laser tracking", In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on* (2000). Vol. 4, p. 3597-3602.

Paul, RP. 1981. *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. The MIT Press, 279 p.

Puskorius, G, and L Feldkamp. 1987. "Global calibration of a robot/vision system", In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on* (Mar 1987). Vol. 4, p. 190-195.

Schneider, P.J., and D.H. Eberly. 2003. *Geometric tools for computer graphics*. Morgan Kaufmann Pub.

Schröer, K. 1999. "Precision and calibration".

Shirinzadeh, B. 1998. "Laser-interferometry-based tracking for dynamic measurements", *Industrial Robot: An International Journal,* vol. 25, n^o 1, p. 35-41.

Slotine, JJE, and H Asada. 1992. *Robot analysis and control*. Wiley-Interscience, 266 p.

Snell, J.E. 1997. "Method and device for calibration of movement axes of an industrial robot". US Patent No. 5,687,293, filed October 25, 1994, issued November 11, 1997.

Stone, H, and A Sanderson. 1987. "A prototype arm signature identification system", In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on* (Mar 1987). Vol. 4, p. 175-182.

Stone., Henry W. 1986. "Kinematic modeling, identification and control of robot manipulators", Boston: Kluwer Academic, 175 p.

Summers, M. 2005. "Robot capability test and development of industrial robot positioning system for the aerospace industry", *SAE transactions,* vol. 114, n^o 1, p. 1108-1118.

Tsai, Y.T., and D.E. Orin. 1987. "A strictly convergent real time solution for inverse kinematics of robot manipulators", *Journal of Robotic Systems,* vol. 4, n^o 4, p. 477-501.

Vincze, M., J.P. Prenninger and H. Gander. 1994. "A laser tracking system to measure position and orientation of robot end effectors under motion", *The International Journal of Robotics Research,* vol. 13, n^o 4, p. 305.

Wang, K. 2009. "Application of genetic algorithms to robot kinematics calibration", *International Journal of Systems Science,* vol. 40, n^o 2, p. 147-153.

Wang, L.C.T., and C.C. Chen. 1991. "A combined optimization method for solving the inverse kinematics problems of mechanical manipulators", *Robotics and Automation, IEEE Transactions on,* vol. 7, n^o 4, p. 489-499.

Wiest, U. 2003. "(EN) 3D Coordinate measure system". WO Patent WO/2003/035,333.

Ye, SH, Y. Wang, YJ Ren and DK Li. 2006. "Robot Calibration Using Iteration and Differential Kinematics", In *International Symposium on Instrumentation Science and Technology* (2006). Vol. 48, p. 1. IOP Publishing.

Zhuang, H., and Z.S. Roth. 1992. "Robot calibration using the CPC error model", *Robotics and Computer-Integrated Manufacturing,* vol. 9, n° 3, p. 227-237.

Zhuang, H., and Z.S. Roth. 2002. "A linear solution to the kinematic parameter identification of robot manipulators", *Robotics and Automation, IEEE Transactions on,* vol. 9, n° 2, p. 174-185.

Zhuang, H., L.K. Wang and Z.S. Roth. 1993. "Error-model-based robot calibration using a modified CPC model", *Robotics and Computer-Integrated Manufacturing,* vol. 10, n° 4, p. 287-299.