IDENTIFICATION OF KINEMATIC PARAMETERS USING POSE MEASUREMENTS AND BUILDING A FLEXIBLE INTERFACE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALİCAN BAYRAM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

SEPTEMBER 2012

Approval of the thesis:

IDENTIFICATION OF KINEMATIC PARAMETERS USING POSE MEASUREMENTS AND BUILDING A FLEXIBLE INTERFACE

submitted by **ALİCAN BAYRAM** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ____________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Süha Oral ____________________
Head of Department, **Mechanical Engineering**

Asst. Prof. Dr. E. İlhan Konukseven ____________________
Supervisor, **Mechanical Engineering Dept., METU**

Prof. Dr. Tuna Balkan ____________________
Co-Supervisor, **Mechanical Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Reşit Soylu ____________________
Mechanical Engineering Dept., METU

Asst. Prof. Dr. E. İlhan Konukseven ____________________
Mechanical Engineering Dept., METU

Prof. Dr. Tuna Balkan ____________________
Mechanical Engineering Dept., METU

Prof. Dr. Kemal İder ____________________
Mechanical Engineering Dept., METU

Dr. Selçuk Himmetoğlu ____________________
Mechanical Engineering Dept., Hacettepe University

Date: 12.09.2012

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Alican BAYRAM

Signature:

ABSTRACT

IDENTIFICATION OF KINEMATIC PARAMETERS USING POSE MEASUREMENTS AND BUILDING A FLEXIBLE INTERFACE

Bayram, Alican
M.S., Department of Mechanical Engineering
Supervisor: Asst. Prof. Dr. E. İlhan Konukseven
Co-Supervisor: Prof. Dr. Tuna Balkan

September 2012, 74 pages

Robot manipulators are considered as the key element in flexible manufacturing systems. Nonetheless, for a successful accomplishment of robot integration, the robots need to be accurate. The leading source of inaccuracy is the mismatch between the prediction made by the robot controller and the actual system. This work presents techniques for identification of actual kinematic parameters and pose accuracy compensation using a laser-based 3-D measurement system. In identification stage, both direct search and gradient methods are utilized. A computer simulation of the identification is performed using virtual position measurements. Moreover, experimentation is performed on industrial robot FANUC Robot R-2000*i*B/210F to test full pose and relative position accuracy improvements.

In addition, accuracy obtained by classical parametric methodology is improved by the implementation of artificial neural networks. Neuro-parametric method proves an enhanced improvement in simulation results. The whole proposed theory is reflected by developed simulation software throughout this work while achieving accuracy nine times better when comparing before and after implementation.

Keywords: Industrial robots, System Identification, Artificial Neural Networks, Relative Position, Laser Sensor

ÖZ

POZ ÖLÇÜMLERİ KULLANILARAK ENDÜSTRİYEL ROBOTUN KİNEMATİK PARAMETRELERİNİN TANIMLANMASI VE ESNEK BİR ARAYÜZÜN OLUŞTURULMASI

Bayram, Alican
Yüksek Lisans, Makine Mühendisliği Bölümü
Tez Yöneticisi: Yrd. Doç. Dr. E. İlhan Konukseven
Ortak Tez Yöneticisi: Prof. Dr. Tuna Balkan

Eylül 2012, 74 sayfa

Robot manipulatörler, esnek üretim sistemlerinin yeri doldurulamaz bileşenleridir. Ancak robot entegrasyonunun fayda sağlaması için robotun yüksek doğrulukla çalışması gerekir. Robot hatalarının en önemli sebebi, robot kontrolcüsü hesaplamalarının gerçek sistemi karşılamamasıdır. Bu çalışmada amaç, lazer bazlı kinematik parametre tanılama ve poz hatalarını düzeltme yöntemlerini sunmaktır. Parametre tanılama işleminde direkt arama ve gradyent bağımlı eniyileme teknikleri kullanıldı. Üretilen sanal konum ölçümleri kullanılarak bilgisayar ortamında tanılama benzetimleri gerçekleştirildi. Bunun yanı sıra, FANUC Robot R-2000*i*B/210F robotu üzerinde tam poz ve bağıl konumlama hatalarının iyileştirilmesi deneylerle sağlandı.

Bu tez kapsamında ayrıca, parametrik yöntemler ile elde edilen poz doğrulukları, yapay sinir ağları uygulaması ile geliştirildi. Geliştirilen sinir ağları yönteminin sağladığı poz doğrulukları benzetim sonuçları kullanılarak gösterildi. Bu tez kapsamında önerilen teori ve elde edilen iyileştirilmiş sonuçlar, geliştirilen benzetim yazılımı kullanılarak sunulmaktadır.

Anahtar Kelimeler: Endüstriyel robotlar, Sistem Tanılama, Yapay Sinir Ağları, Bağıl Konumlama, Lazer Algılayıcı

ACKNOWLEDGEMENTS

I express my gratitude to all the people that have contributed and inspired me to successful conclusion of this study. Special thanks to my supervisor Asst. Prof. Dr. E. İlhan Konukseven and my co-supervisor Prof. Dr. Tuna Balkan for their valuable guidance during my thesis.

I would like to state my appreciation to Şenol Meriç from Oyak Renault for supporting equipments in experiments and hospitality in the company, Berk Yurttagül for his useful comments; my labmates in Mechatronics Laboratory Serter Yılmaz and Özgür Başer for their useful advices, Gökhan Bayar for his endless support, Masoud Latifi-Navid and Matin Ghaziani for enjoyable attitude throughout this study.

Finally my last, but not least, I am deeply grateful to mechanical engineer Yiğit Koyuncuoğlu and software engineer Birkan Polatoğlu for their enlightening comments and very special thanks go to my valuable family.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS

a_i	Effective link lengths
d_i	Joint offsets
dP^T	Differential error vector of the origin O_T;
dR^T	Differential rotation vector of the origin R_T;
D	Difference vector
e_i	Residue in i^{th} measurement
J	Generalized identification Jacobian matrix
J^p	Identification Jacobian for position functions
J_a^p	Component of identification Jacobian for position functions
$J_a(q,p)$	Component of identification Jacobian
L	Cost function
L_e	Cost function value at expansion point
p	Parameter vector
p_r	Parameter vector at reflection point
p_1, p_2, p_3	Components of parameter vector
$\overline{p}$	Centroid of simplex
${}_B P^T(y)$	Measured location of the end-effector frame T
${}_B P^T(q,p)$	Theoretical location of end-effector frame T
P_x, P_y, P_z	End-effector positions

${}_R POSITION$	Generalized position vector
q	Generalized vector of system inputs
Q	Vector of random configurations
Q_Φ	Orthogonal matrix from decomposition of Φ
R^i	Covariance matrix element of i^{th} measurement
R_Φ	Upper triangular matrix from decomposition of Φ
${}_B R^T_{measured}$	Orientation matrix from measurements
${}_B R^T_{computed}$	Orientation matrix from calculations
${}_R ROTATION^T$	Generalized rotation matrix of frame T
S	Diagonal matrix from single value decomposition of Φ
${}_{i-1}T^i$	Transformation matrix
U	Orthogonal matrix from single value decomposition of Φ
V	Conjugate transpose from single value decomposition of Φ
y	Generalized vector of observations
Y	Observations vector corresponds random configurations
~	Angle about z axis
~$_i$	Twist angles
~$_s$	Reflection coefficient
~$_N$	Momentum factor
	Angle about y axis
${}_i$	Hayati parameters
${}_s$	Expansion coefficient
¥$_s$	Contraction coefficient
¢$_s$	Shrink coefficient

Δp	Vector of the change in geometric parameters
ΔP	Position error in end-effector coordinates
ΔR	Rotation error in end-effector coordinates
¦$_N$	Learning rate
¯	Angle about x axis
¯$_i$	Joint variables
a	Adjustable scalar
°	Presentation of modeling errors and un-modeled factors
±	Standard deviation
Φ	Observation matrix
w	Scale matrix
Ω	Presentation of geometrical shape, simplex

CHAPTER 1

INTRODUCTION

1.1. General Background: Flexible Manufacturing

For several decades, flexibility is becoming irreplaceable for manufacturing systems. In order to respond to the fast-changing product designs due to the consumers' and manufacturers' tastes, producers should adapt their manufacturing systems to the trending demands. Recent academic research has been focused on a special type of flexibility called "product-design flexibility" which is becoming major concern of the producers while production volume flexibility is still popular [1]. By utilizing the product-design flexibility in the manufacturing systems, producers are able to adapt to trending demands in the market in a fast manner.

The degree of flexibility in manufacturing systems is a function of fluctuations in the amount of the product demands and variation in consumer preferences. In plain words, with stable consumer tastes, producer dedicates a mass production system to the desired one product. But in the markets where fluctuations in volume and variation in consumer preferences exist, a dedicated manufacturing system undoubtedly ends up in loss of market share or close down for rearranging the plant [2].

In history, several car manufacturers' production has been suffered due to the long close down periods for retooling. As a result of their dedicated manufacturing line, it was impossible to adapt to the trending product changes that are demanding in the market. Since the degree of the inflexibility had been high, cost of the transition was high for these car manufacturers [1].

A flexible manufacturing system (FMS) is consisting of a control center and a group of numerically controlled machines, as control center's clients. Simply a FMS is considered as an automated production system that produces several parts in a flexible manner. The evolution of FMSs has been experienced during 1960s when robots, controllers, and numerical control came to the factory environment [2]. All of these concepts have considerably increased the competitiveness of the producers, since higher efficiency in planning and production is resulted. Shorter production time, higher utilization of tool and machines is achieved and inevitably production cost is reduced [2].

It can be inferred that when the robots and controllers have higher performance, gain from the FMS is increased, and also competitiveness of the producer increased by satisfying demanded tight tolerances.

1.2. Performance of the Robots

Performance of the robots is characterized by concepts of resolution, precision and accuracy. Hardware and construction procedure of the robotic systems affect each of these concepts and they could be introduced in joint space and world space subsequently [3]:

- *Resolution* is the smallest incremental move that a robot is able to produce in end-effector coordinates. At the joint level, resolution is characterized by resolution of the encoders [3].
- *Precision or Repeatability* is a measure of the capability of the manipulator to reproduce same position in 3-D space when the same command is given repeatedly [3]. According to ISO 9283, repeatability refers to the ability to reproduce to same pose from the same direction in response to the same command [4].
- *Accuracy* is the measure of the ability of the robot to produce desired position in 3-D space. According to ISO 9283, accuracy refers to difference between the theoretical pose that is computed and the actual pose that is measured [4].

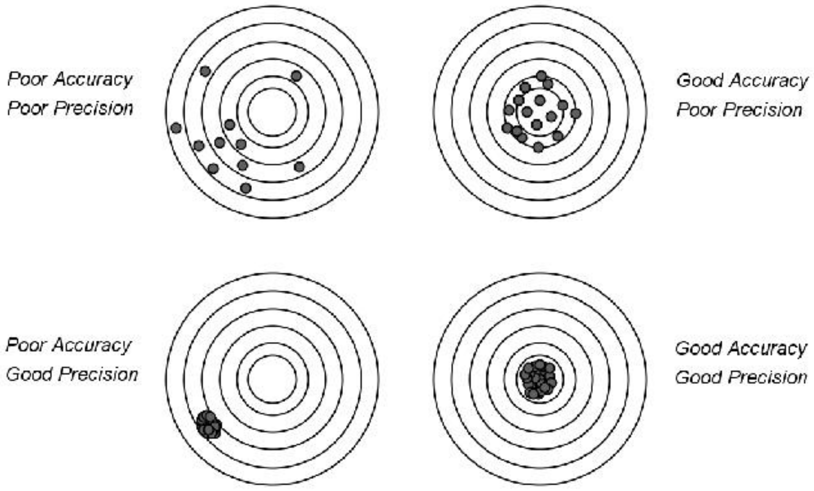


Figure 1.1: Repeatability and Accuracy Concepts from Shooter-Target Example

An illustrating analogy using shooter and target example explains the repeatability and accuracy concepts. Inherently resolution and precision are determined by robot components and their capabilities. However, robot models could be more accurate by identification of model parameters [3].

1.3. Basics of Robot Calibration

Accuracy of the robot is determined by geometric and non-geometric factors present in robot components and assembly. These error sources can be modeled and enhancement could be obtained in a process known as robot calibration [5].

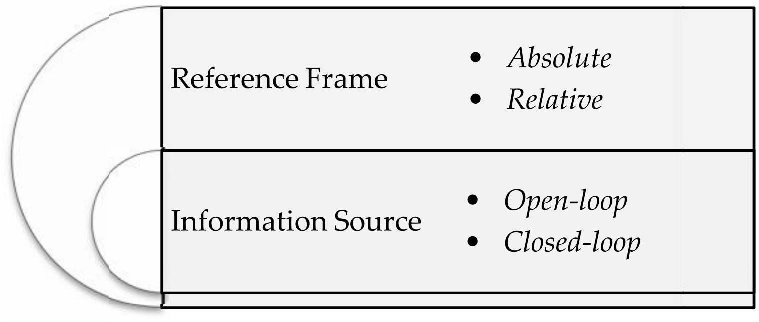


Figure 1.2: Classification of Methods

Absolute calibration refers to the ability of a robot to move a desired location in terms of the absolute reference frame of the system. In applications, in which several robot manipulators share the same workspace, defining a world reference frame is essential. On the other hand, *relative calibration* is the process means that errors are modeled relative to the object frame [5].

In literature, another classification on calibration methods based on the information sources of errors. In *open-loop calibration* procedures, an external sensor (e.g. camera, laser tracker systems and coordinate measuring machines) is used for the actual pose information. On the other side, *closed-loop calibration* does not require an external sensor. Instead end-effector frame is constrained to a fixed geometrical object and consequently difference between different encoder configurations corresponding to the end-effector frame is used as the information source.

1.4. Objective of the Thesis

The main objective of this thesis is to enable industrial methods for high accuracy tasks by performing a robust sensor-based calibration approach. To do so, identification approaches are examined by simulations and experiments. Besides modeling the robot geometrical errors by parametric means is achieved, sources of inaccuracy are also modeled by non-parameteric tools. The goals of this work could be classified as:

- Obtaining measurements on the industrial robot using a high precision external sensor.
- Improving accuracy of the case study robot using full-pose and relative position measurements.
- Developing modular simulation software based on the theory of calibration approach which will be used throughout this work and improving accuracy of the case study robot using virtual position measurements.
- Implying the gradient-free alternative identification algorithm into the approach.
- Implying non-parametric approach in compensation manipulator errors by simulations.

1.5. Outline of the Thesis

In Chapter 2, the literature survey is given to discuss the previous work proposed in the literature. In this chapter, efficient measurement systems in the literature are also given because the proposed approach is to be designed to handle them.

Since the approach needs models of the physical system, Chapter 3 focuses on the model development stage. In this stage analytical problems in notation and solutions are also introduced. When the model is obtained and data is gathered, model parameters should identified by using related approaches. Gradient methods and direct search algorithms are introduced throughout the Chapter 4. All proposed theory is reflected through developed modular simulation software; briefly features of the software are introduced in Chapter 5. In this study, experiments and simulations are applied to the case robot; results will be concisely introduced in Chapter 6 by using software's numerical and visual outputs. Consequently, in Chapter 7, discussion concludes the thesis.

CHAPTER 2

LITERATURE REVIEW

In this chapter calibration methods in the literature are introduced. More explicitly, complexity levels of calibration approaches, efficient sensors for measurements, methods of kinematic modeling and parameter identification algorithms and results are reviewed.

2.1. Off-line vs. teach-in programming

In mass production industries, the most part of the industrial robots are still programmed by a teach pendant. This approach is called *"Teach-in programming"* in which simple sequences of motion are recorded into the memory of the controller. However this requires specifying the trajectories by teaching reference points for each operation using teach-pendants [5].

Due to the fast-changing nature of operations in the industry, reprogramming of the robots via teach-by-doing ends up in machine downtime (consequences are explained in Section 1.1). *"Off-line programming"* strategy shortens the downtime by using models and simulations on the work cells, yet models need to be repeatable and very accurate.

To obtain the accuracy at desired level, calibration is necessary [5]. Eventually, reduction in maintenance expenditure is achieved by robot calibration since machine quickly returns from downtime [6].

2.2. Complexity of Error Models

Calibration procedures in literature could be classified according to the complexity level. Procedures vary from joint level methods to dynamic methods. Roth, Mooring and Ravani describe three complexity levels of manipulator calibration for classification of the approaches [7].

Level 1 calibration aims to determine the relationship between the input pulse supplied for desired joint displacement and the obtained joint displacement, termed as joint offsets [3]. Practically, process includes calibration of joint sensor readings. To do this, complex relationship between desired joint position and real joint position can be approximated with linear functions.

Level 2 calibration includes all errors sources of entire robot kinematics geometry. This process covers the calibration of joint-angle relations. In addition, non-kinematic error sources are also encountered by Level 3 calibration. Joint and link compliance, friction and backlash are the non-geometric factors that affect the end-effector positioning. Level 3 calibration is applied to robot in a combination with Level 2 and Level 1 calibration [7].

2.3. Effect of Kinematic Model Complexity

Differences between actual robot model and the theoretical model are sources of the manipulator errors. Possible sources of error and their impact on the overall accuracy are worth to know. Mainly sources of the inaccuracy are divided into geometric and non-geometric errors.

Geometric parameters are shortly refers to the parameters that are independent from load and motion type (e.g. link length, offset, twist, rotation) [7]. The geometric errors of a robot take their source from manufacturing tolerances, wear and misaligned assemblies [9]. On the other side, term "non-geometric errors" refers to joint compliance, flexibility errors of links, gear backlash and controller errors and propagated by self-gravity and load [3].

Impact effect of the errors sources have been investigated by a number of researchers. A disagreement on the significance of the error sources exist, while recent studies point out the geometric errors as the primary factor. Non-geometric errors appear as the most significant error source in the study of Whitney, Lozinski and Rourke [8]. On the contrary, Judd and Knasinski reduced 95% of the error by calibrating geometric errors and additional improvement of 0.5% was achieved by compensating gearing errors [10].

According to the Conrad, Shiakolas and Yih [11], inaccuracy issues are caused by robot zero offset with 97%. This study claims that shipping and installation affect robots' zero offset configuration. Similarly, Veitschegger and Wu took geometric parameters into consideration and reduced positioning errors from 21.746 mm to 0.3 mm [12]. In addition, Gong, Yuan and Ni have performed geometric and non-geometric calibration techniques step by step. Geometric errors correspond to the 88% of the total error where 3.6% additional improvement has obtained via calibrating stiffness factors of joint 2 and 3 (see Figure 2.1) [9].

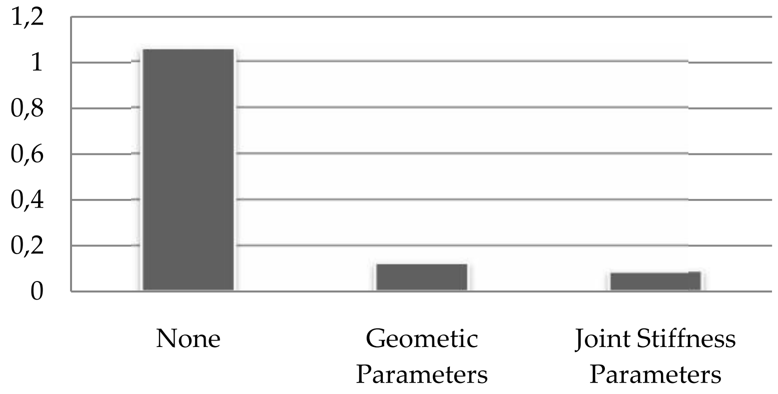


Figure 2.1: Reduction of Mean Positional Errors by Identifying Additional Parameters (mm) [9]

Identical study in figures out that by identification of geometrical parameters, absolute accuracy is changed from 2.82 mm to 0.69 mm and by including compliance error result reduced to 0.58 mm [13]. One more identical study has been done by Morring and Padavala, in their study several models have been obtained from simple to most detailed one. Using geometric parameters mean positioning error is reduced from 30.04 mm to 0.50 mm. Further improvement was obtained by adding compliance effect of joint 2 and 3 [14]. Chen and Chao reduced the positional error from 5.9 mm to 0.28 mm by identification of true angles of joint 2 and 3 in addition to geometric parameters [15].

Jang, Kim and Kwak introduced an alternative approach for robot calibration using radial basis functions for estimating the error, their approach results in improvement in mean error from 0.92% to 0.154% [16]. Authors claim that method estimates the error in a certain local region of the workspace [16]. Alternatively, Shamma and Whitney have built a inverse calibration method that contain a phenomena called "black box". The black box contains a nominal inverse model and approximating functions. Using nominal joint angles from inverse calculations, functions approximate to the desired joint angles. This method is verified by simulations, positioning error is reduced from 1.5 mm to 0.25 mm [17].

To sum up, all these studies show that errors caused by geometric factors dominate overall manipulator errors. For that reason, many of the researches totally neglect the influence of non-geometric errors or consider only some types of sources such as joint compliance and gear backlash. In recent researches, estimation and inference methods are trending for compensating non-geometric factors, since analytical methods augment the complexity of the model and robot controllers only contain geometric parameters.

2.4. Model Development

Model development stage, in calibration procedure, is to obtain a relation between inputs and outputs of the system. Essentially, two types of mapping required in order to complete the procedure. While forward model maps real system inputs (joint displacements) to the output (end-effector pose), inverse model works in the opposite direction.

Stone and Sanderson’s S-model is a popular modeling and identification approach, which is based on joint features [20]. Substance of this approach comes from the ease of handling where consecutive joint axes are parallel. On the other hand, robot controllers do not use S-model parameters. An efficient solution from [21] supplies direct identification of Denavit-Hartenberg (DH) kinematic model parameters when S-model is used in model development stage. He, Zhao, Yang and Yang provides an alternative modeling approach for configurations where consecutive joint axes are almost parallel, called product of exponentials formula, also eliminates the redundant parameters that are exist in other conventions [22].

The most common approach for the model development stage is the Denavit-Hartenberg (DH) method where four parameters used to transform one link to another. The product of the transformation matrices maps reference frame to the tool frame. Transformation matrix is obtained from three translations and three rotations about the frame axes [23]. Models based on DH, proportionality of the direct model is altered when adjacent joint axes are not perfectly parallel. Hayati and Mirmirani suggested a solution to be used in that case [24].

A kinematic model should respond three fundamental prerequisites in order to be useful in identification. Relation between joint and end-effector positions for all configurations should be met, this property called completeness. Secondly, model should have relations with alternative modeling approach, called equivalence or continuity. Thirdly, proportionality refers to that small variations in the structure should lead to small proportional variations in kinematic parameters, otherwise small variations in parameters end up in unexpected results [18, 19].

One alternative accuracy improvement approach is introduced in [25] by Bai and Zhuang. Errors are estimated by using fuzzy interpolation technique. Errors are calculated for measured locations then the error interpolation is concluded for a unmeasured point by fuzzy inference. In addition, genetic algorithm is utilized by Dolinsky, Jenkinson and Colquhoun in [26]. Geometric and non-geometric parameters are included by the nature so that mean error is reduced from 1.85 mm to 0.77 mm.

2.5. Measurement Systems

According to the information provided by the measurement system, efficient systems in the literature could be grouped as in [27]. Identically, in this thesis, classification has been done with respect to the independent quantities provided by the system.

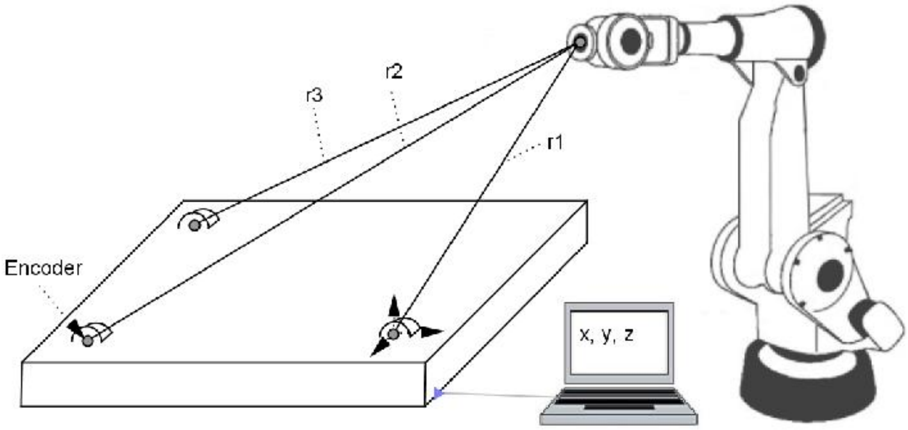


Figure 2.2: Measurement System Consisting of Three Cables

In order to determine independent coordinates of the tip point, coordinate measurement machines (CMMs), three (or four) cable systems and laser tracker systems could be used. In three cables system, on a plate there exist three encoders attached to the cables which are connected to the end-effector frame in one point. By reading cable's push-pull lengths (r1, r2, r3), coordinates of the end-effector tip is determined (see Figure 2.2). For CMMs, spherical object is used as the end-effector. Sensitive probe measures coordinates of the center by touching several points on the surface of the sphere. An advantage of this system is that measurement procedure could be automated. On the other hand, robots are large in dimension are inconvenient for this system since large CMMs are limited in the industry. Alternatively using two theodolites, angles from horizontal and vertical planes were measured and consequently position the end-effector was found as in [10]. Similarly, Chan and Chao employ three theodolites for measuring the location of the end-effector [15].

One of the novel studies in measurement systems was proposed by Vincze et al [28]. By using a single laser beam, Cartesian coordinates and orientation is determined. System is automated and all motions of the end-effector are tracked by the proposed device [28].

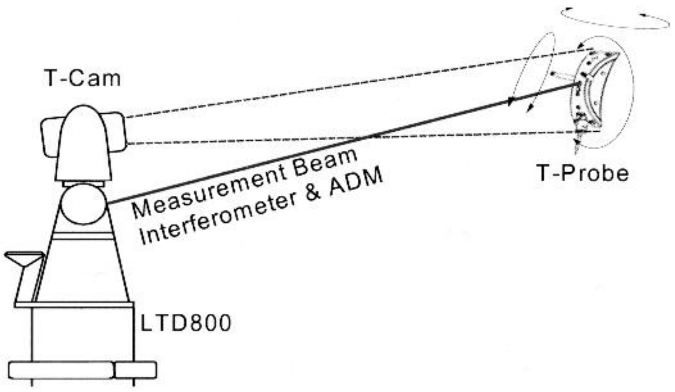


Figure 2.3: A Combination of Laser Interferometer and Handheld Probe [49]

For one component measurements, wire potentiometer [30] or linear variable differential transformer (LVDT) systems have been used to find distance measurements [29]. LVDT is a moveable device, attached between end-effector frame and reference frame, measures the actual distance of the ball-bar. In the meanwhile, nominal forward kinematic model is processed and expected ball-bar length is calculated. Difference between the actual and nominal model is tried to be minimized by regression [29]. Relative measurements are also reached from laser tracker system since these devices measure the pose of the manipulator as in [48].

An alternative approach by Lee et al. proposes the implementation of indoor global positioning system (IGPS) for robot calibration. Position and orientation (only roll and pitch) data is gathered by using LED and laser sensors [31].

2.6. Identification of Parameters

After developing nominal kinematic model, a set of parameters are to be adjusted so that model fits to the measured data. Consequently, process produces actual model by obeying directions of the data. In literature this process is called identification. More specifically, the parametric identification technique is used to be a data fitting problem. Technique uses nominal kinematic parameters as the initial guess then kinematic parameter errors are estimated. In the estimation stage, problem is defined as nonlinear parameter estimation problem in many works in literature. Two efficient solutions are available in this step: Direct search methods and gradient methods. Although gradient-free search methods converge relatively slow, application with suitable initial guesses increase the convergence properties and reduce the number of iterations. On the other side, gradient methods require the calculation of identification Jacobian. Both approaches have successful applications in the literature; a summary is given in the following part of context by adding efficient alternative methods at the end.

The Levenberg-Marquardt (LM) algorithm is one of the most efficient approaches among gradient methods. Because of the singularity problems with Gauss-Newton algorithm, Levenberg [33] and Marquardt [34] independently developed solutions. LM algorithm based on combination of two conventional algorithms; the steepest descent and the Gauss-Newton methods. For that reason, convergence properties inherently taken from Gauss-Newton algorithm when it converges to the solution. A successful application among many in literature for calibration procedure can be found in [32], in which Khalil and Besnard identify the parameters of geometric and compliance models.

In the last few years, intelligent identification techniques are trending in the literature while conventional methods keep their importance. Dolinsky et al. [26] identify kinematic parameters with genetic algorithms.

In addition, Bai and Zhuang propose a dynamic modeless calibration method which covers fuzzy inference of positional errors [25]. For a certain part of the workspace, fuzzy inference system interpolates the error from neighboring errors. Alternatively, neural networks are used to estimate the errors due to the non-geometric factors [35]. The inputs of the network are the joint readings and corresponding nominal pose results while the output is guess for the pose errors.

2.7. Implementation of Calibrated Parameters

In the first step, a nominal model, that maps inputs to output, is obtained. After identifying actual robot parameters, nominal robot model is calibrated in order to predict real robot motion. At this point, identified parameters should be updated instead of the nominal parameters in robot controller.

Due to the fact that robot controllers work with geometric models, some identified parameters from complex models could not be entered. For this situation, inverse Jacobian is used to find the joint angles that correspond to the pose errors. Then iteratively joint angles are supplied to the model until the error is reduced under the threshold. This method requires one inverse calculation and one inverse Jacobian manipulation in addition to the forward model.

On the other hand, fake targets approach requires only inverse calculation in addition to forward model [50]. This approach work simply as the desired pose is used as the input of the inverse model with identified parameters then resultant joint angles are used for fake targets in the nominal forward model (see Figure 2.4). Consequently, in order to reach target poses, fake targets are serving.

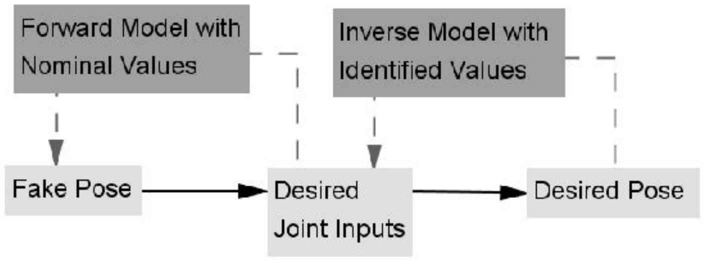

Forward Model with Nominal Values
Inverse Model with Identified Values
Fake Pose
Desired Joint Inputs
Desired Pose

Figure 2.4: Implementation of Results using Fake Pose

CHAPTER 3

MODEL DEVELOPMENT

3.1. Kinematic Model

3.1.1. Basic Descriptions: Frames and Transformation

Robot manipulators are mechanisms that are consisting of tools and parts, and designed for spatial motion. Before commanding a motion; position and orientation of the parts and tools, relative to each other, should be investigated. In order to obtain mathematical descriptions relating the system's inputs and outputs, several conventions could be found in literature [37]. By using proper convention for the case, a set of coordinate systems are defined and transformations between coordinate systems are obtained to develop the kinematic model of the manipulator. Following sections are designed to illustrate the relating concepts.

3.1.1.1. Describing Position Vector and Rotation Matrix

Any point in space could be represented with a 3x1 position vector relative to a coordinate system. If several coordinate systems are existed in the context, reference coordinate system should be indicated by the notation.

$$ {}_{R}POSITION = \begin{bmatrix} p_x \\ p_y \\ p_y \\ w \end{bmatrix} \tag{3.1} $$

Elements with subscripts x, y, z in of the vector represents distances along axes and w represent a scale factor, generally equals to the unity [37].

A physical object requires complete location to be specified. This needs the position vector presentation as well as the orientation presentation relative to the reference. Determining orientation of the body requires a frame to be assigned on the object.

$$ {}_{R}ROTATION^{T} = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \tag{3.2} $$

In Equation 3.2, columns represent the x, y, z components of the projections on the unit directions of the reference frame R, calculated by scalar product. This means, the position of a point is described by a column vector while the orientation is described with a matrix.

Four column vectors giving rotation and position information termed as frame in robotics. In order to change the description from one frame to another, mapping need to be used.

$$ \{T\} = \left\{ {}_{R}ROTATION^{T}, {}_{R}POSITION \right\} \tag{3.3} $$

In order to find the expression of inverse mapping from tool to reference frame, transpose of direct mapping would be sufficient.

3.1.1.2. Representation of Orientation

In the form of 4x4 matrix, Equation 3.4 is called homogenous transform including both translation and rotation;

$$\left[\begin{array}{c|c} {}_{R}ROTATION^{T} & {}_{R}POSITION \\ 0 \quad 0 \quad 0 & 1 \end{array}\right] \tag{3.4}$$

While translations along three perpendicular axes are easy to visualize, rotations need some effort. Considering pure rotational transformation for ease of understanding, three consecutive rotations about three axis possibly described by; firstly rotate with angle ~ about z axis of frame $\{R\}$, secondly with angle about y axis, thirdly with angle ¯ about x axis. This sequence called Z-Y-X Euler sequence and described as follows and will be used throughout this work;

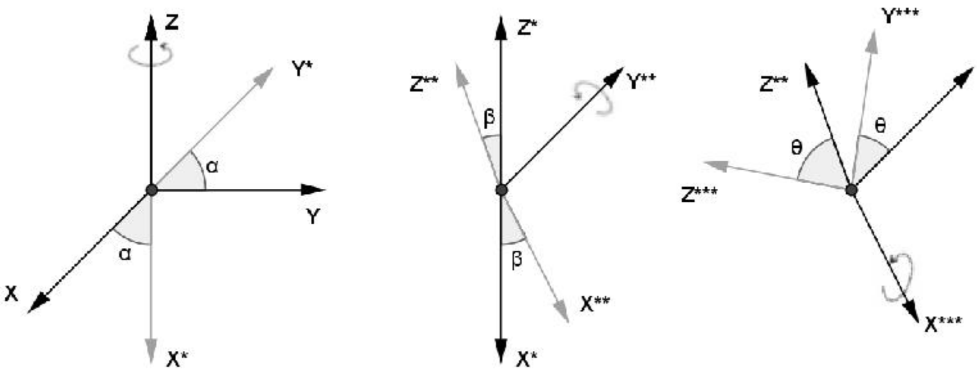


Figure 3.1: Z-Y-X Euler Angles

In matrix form, rotations are written as three separate rotations in multiplication as follows:

$${}_{R}ROTATION^{T} = R_{Z}(\sim)R_{Y}(\ \ )R_{X}(\bar{\ }) \tag{3.5}$$

Consequently as in [36],

$$R_Z(\sim)R_Y(\ \ )R_X(^{-}) = \begin{bmatrix} c\sim & -s\sim & 0 \\ s\sim & c\sim & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c & 0 & s \\ 0 & 1 & 0 \\ -s & 0 & c \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c^{-} & -s^{-} \\ 0 & s^{-} & c^{-} \end{bmatrix} \tag{3.6}$$

where $c^{-} = \cos^{-}$, $s^{-} = \sin^{-}$ and so on. When the resultant rotation matrix is known (Equation 3.7) and Euler angles need to be extracted;

$${}_{R}ROTATION^{T}(\sim,\ \ ,^{-}) = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \tag{3.7}$$

As long as $c\ \ \neq 0$ solution is found as [36];

$$\begin{aligned} &= A\tan 2(-n_z, \sqrt{n_x^{\ 2} + n_y^{\ 2}}) \\ \sim &= A\tan 2(\frac{n_y}{c}, \frac{n_x}{c}) \\ ^{-} &= A\tan 2(\frac{o_z}{c}, \frac{a_z}{c}) \end{aligned} \tag{3.8}$$

3.1.2. Assignment of Coordinate Frames

For kinematic analysis, motion of each link relative to its neighbors is described by using frames attached to the each link. For this purpose, several approaches and notation have been proposed in literature. The most popular method among others has been introduced by Denavit and Hartenberg [23]. Modified DH convention by Khalil and Kleinfinger is used in this study [38]. In this study, links and joints are assumed to be ideal since manipulator error source analysis (explained in Section 2.3: Effect of Kinematic Model Complexity) is stating that errors are mainly caused by geometric factors. Therefore backlash and elasticity are not modeled and study is focused on articulated robot configurations.

For n degrees of freedom robot, n+1 link frames are assigned where for the base frame letter "R" is used. Subsequently, moving bodies numbered from 1 to n where same situation is applied for the joints. Frame assignment starts with assigning z_i for the axis of joint i . Then, between adjacent z_{i-1} and z_i axes, common normal is obtained to locate x_{i-1} . The y_i axis is assigned by obeying the right-hand rule. The origin O_i is found, the point where z_i and x_i intersects [38].

After assigning frames, four parameters are used for defining the location of one frame relative to the neighboring frame as in Figure 3.2 [23, 38];

- a_i is the distance from z_{i-1} to z_i along x_{i-1},
- d_i is the distance from x_{i-1} to x_i along z_i,
- $\sim_i$ is the angle from z_{i-1} to z_i about x_{i-1},
- $\overline{\ }_i$ is the angle from x_{i-1} to x_i about z_i.

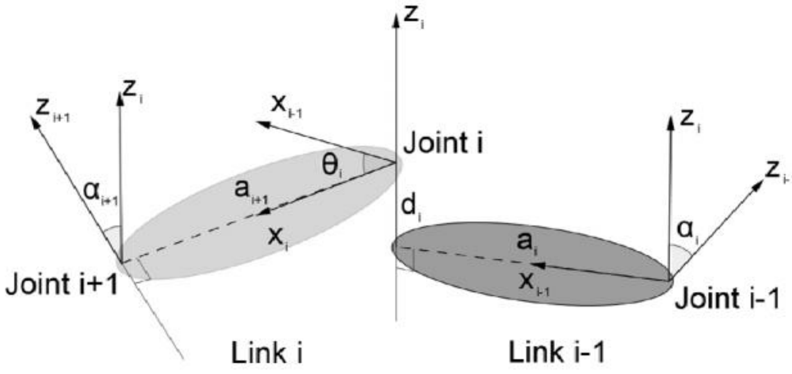


Figure 3.2: Representation of Geometric Parameters

In this convention, location of frame i relative to the frame $i-1$ is defined by using transformation ${}_{i-1}T^{i}$ [38];

$$ {}_{i-1}T^{i} = Rot(x,\sim_{i})Trans(x,a_{i})Rot(z,^{-}_{i})Trans(z,d_{i}) \tag{3.9} $$

Here 3x3 rotation matrix for adjacent frames $i-1$ and i could be extracted as;

$$ {}_{i-1}ROTATION^{i} = Rot(x,\sim_{i})Rot(z,^{-}_{i}) \tag{3.10} $$

If the neighboring axes are nearly parallel, the common normal between the joint axes is not defined properly. For that reason, x_{i-1} is chosen arbitrarily as one of the common normal of joints $i-1$ and i [38]. This can result in disproportionate change in the kinematic parameters, and ends up in instability in the parameter identification [24]. For this potential problem, Hayati proposed an extra rotational parameter $\ _{i}$ about the y_{i-1} axis as in Figure 3.3 [38].

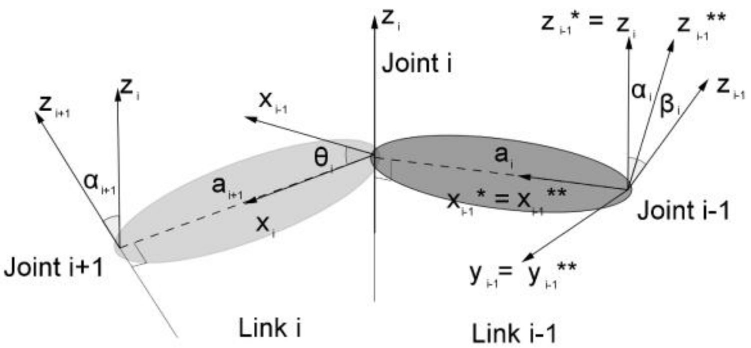


Figure 3.3: Representation of Hayati Parameter

Here, transformation from frame i relative to the frame $i-1$ becomes [38];

$$ {}_{i-1}T^{i} = Rot(y,\ _{i})Rot(x,\sim_{i})Trans(x,a_{i})Rot(z,^{-}_{i}) \tag{3.11} $$

3.1.3. The Kinematic Chain

Transformation matrices have been described for single frame transformation so far; however a robot manipulator contains a serial chain. Homogeneous transformations from reference coordinate frame to end-effector coordinate frame are termed as the kinematic chain. The kinematic chain is a sequential product of the homogenous transformations [38].

$$_{R}T^{T} = {_{R}T^{1}}\,{_{1}T^{2}}\,{_{2}T^{3}}\cdots{_{n-1}T^{n}}\,{_{n}T^{T}} \tag{3.12}$$

The forward model provides mapping from the joint coordinates to the location of the end-effector. Conversely, inverse model supplies joint variables for a specific pose of the manipulator. Inverse geometric model, when it compared to forward geometric model, is highly complex. Generally multiple solutions are available; on the other hand analytical solution is not possible for some configurations. For configurations have spherical wrist, that is the in the intersection point of three consecutive revolute joint axes, solution could be provided by analytical means. In the following section, model description is illustrated on the case robot FANUC Robot R-2000*i*B/210F for forward mapping, inverse mapping calculations could be found in Appendix B in detail.

3.1.4. Case Study: FANUC Robot R-2000*i*B/210F

The forward kinematics of a 6-DOF robot could be defined as; model input is a given set of joint angles $q = \left[q_1, q_2, \ldots, q_6\right]^T$, corresponding poses of the end-effector relative to the base frame are the model's output. In the first step, using the convention described in the previous section, frames are attached (Figure 3.5).

Table 3.1: Modified Denavit-Hartenberg Parameters

Link	$a_i(mm)$	~$_i(rad)$	$d_i(mm)$	¯$_i(rad)$	$_i(rad)$
R	-	-	0	0	-
1	0	0	670	¯$_1$	-
2	312	–® / 2	0	¯$_2$ –® / 2	-
3	1075	®	-	¯$_3$	0
4	225	–® / 2	-1280	¯$_4$	-
5	0	–® / 2	0	¯$_5$	-
6	0	® / 2	-235	¯$_6$ –® / 2	-
T	182	®	205	®	-

In the robot configuration of FANUC Robot R-2000*i*B/210F, the axes of second and third joints are parallel. Therefore, referring to the theory explained in the previous section, Hayati parameter $_3$ is needed to be used in the transformation matrix between these axes and d_3 is not used (see Figure 3.5). In Figure 3.4 joint axes of the case robot is given, frame assignment is started by determining joint axes correctly. The resultant Modified Denavit-Hartenberg parameters are shown in Table 3.1.

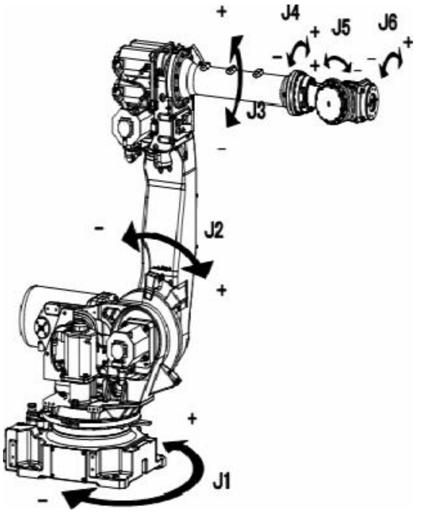


Figure 3.4: Joint axes of FANUC Robot R-2000*i*B/210F [39]

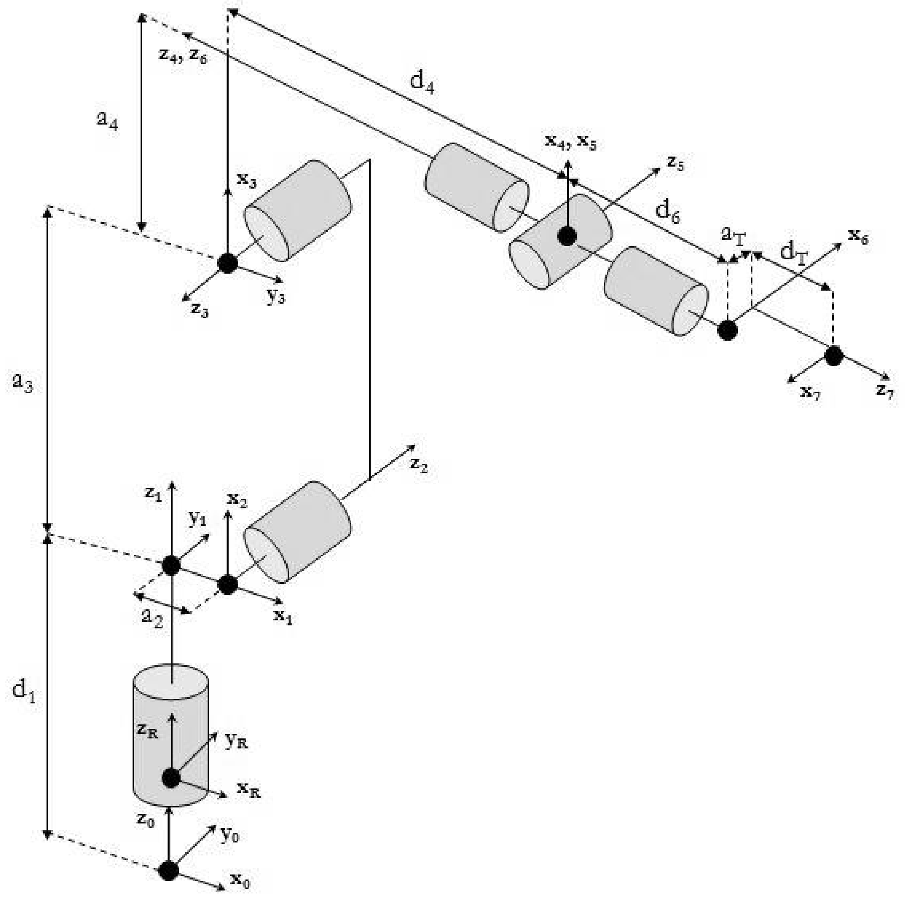


Figure 3.5: Robot Configuration

Modified Denavit-Hartenberg parameters are used in sequential product of the homogenous transformations leads forward kinematic solution.

$$ {}_{R}T^{T} = {}_{R}T^{1}\,{}_{1}T^{2}\,{}_{2}T^{3}\,{}_{3}T^{4}\,{}_{4}T^{5}\,{}_{5}T^{6}\,{}_{6}T^{T} \tag{3.13}$$

The inverse kinematics of a 6-DOF robot could be defined as; model input is the pose of the end-effector relative to the reference frame, a given set of multiple solution consisting of joint angles $q = \left[q_1, q_2, ..., q_6\right]^T$ are the model's output. For manipulator configurations with spherical wrist, analytical solution is available through the matrix manipulations and nonlinear equation solving. By virtue of the spherical wrist, position of the end-effector is determined by first three joints.

The forward and inverse models of the case robot are symbolically obtained for manipulator error minimization steps, optimization and implementation, respectively.

Detailed information about inverse model and calculations could be found in Appendix B.

3.2. Neuro-accuracy Compensator

The parametrical calibration approaches needs development of the kinematic model, to relate the joint reading to end-effector pose. These approaches improve performance with high rates and reach the robot repeatability [13, 14, 15]. However, almost all manipulator calibration methods do not simulate all error sources. Obtaining complicated models, identification of the model parameters and implementation of the identified model dictate several difficult analytical procedures. Besides, only geometric factors can be parameterized for implementation into the robot controller.

In order to reduce the complexity of analytical procedures, alternatively input-output relationships of robot is tried to be modeled by using multilayer neural networks (NN) for compensation in world coordinates. NN approaches are very useful due to their computation efficiency, however NN solutions are not able to reach accuracies of analytical approaches, by their nature, and therefore hybrid application is suggested by many researchers [35, 54]. If the resultant accuracy after parametrical calibration is not in the order of robot's repeatability which means developed model is not complex, further improvement could be achieved by neural-accuracy compensation. In this study, a neural network has been designed for predicting compensated location. It is desired that neural network encapsulates all error sources, taught by input-output data, which affect end-effector error without analytically analyzing and modeling all factors.

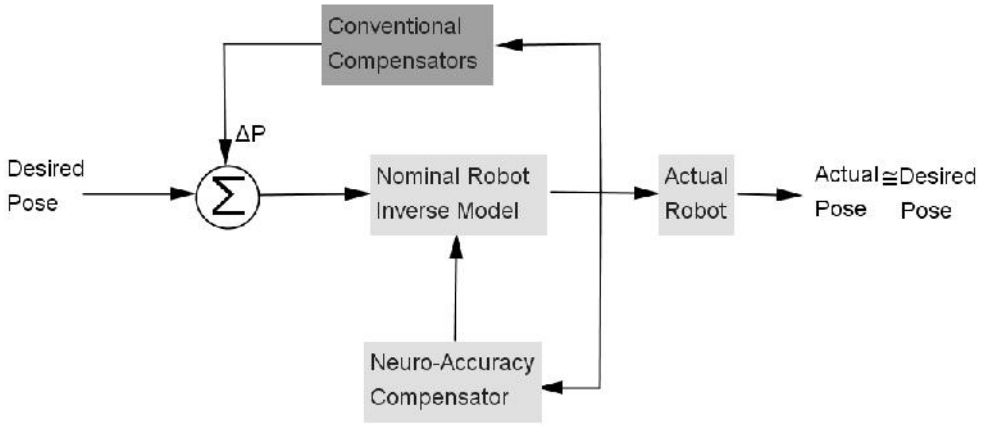

Conventional Compensators
ΔP
Desired Pose
Σ
Nominal Robot Inverse Model
Actual Robot
Actual ≅ Desired Pose Pose
Neuro-Accuracy Compensator

Figure 3.6: Neuro-accuracy Compensation Scheme (in comparision)

The neural network is developed for use in cooperation with nominal forward model in predicting compensated location coordinates in response to given joint configurations. However, in the literature [35, 54], task points are tried to be corrected by predicting error vectors belonging to one location, this process called forward compensation. Clearly, proposed approach is different than conventional methods, since it includes a nominal forward model containing geometric error factors.

A neural network contains neurons that compute a linear or nonlinear transfer function to compute the output. Transfer function is the sum of the weighted inputs and a bias where network weights are adjusted in training process.

In order to predict outputs of the system in response to new inputs, network has to be well-trained [48].

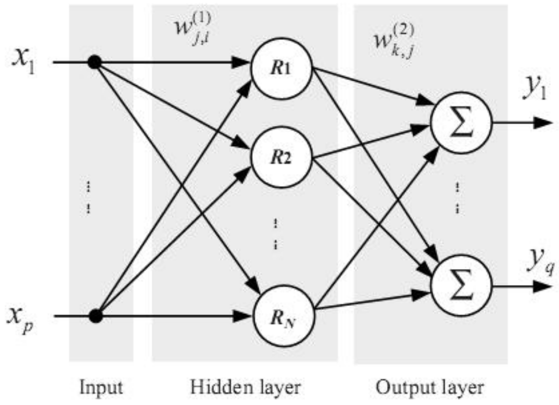


Figure 3.7: Typical Neural Network Structure [48]

Although different compensation schemes and neural network models have been developed, test results suggest the training parameters as: momentum ~$_N$ (0.9) and learning rate ¦$_N$ (0.01). These values proved to be suitable when Levenberg Marquardt (LM) type learning rule is used for weight refining phase. A multi-layer feed-forward NN is suitable when the input-output relation is highly non-linear, therefore typical structure as in Figure 3.7 is used in the study. Transfer functions in the hidden layer are sigmoidal functions, whereas output layer is containing linear functions.

CHAPTER 4

IDENTIFICATION

After describing the robot model and collecting the measurements data using sensors, model parameters that match measured data need to be determined. This stage of the process is called "identification" in the literature. Identification of the robot model has its source in estimating the parameters in order to eliminating the mismatch between the robot observations and the theoretical model [38]. The approach is to view problem as one of fitting a nonlinear regression model [7]. Definition of the nonlinear model for the manipulator is

$$y_i = f(q_i, p) \qquad i = 1,2,\ldots,n \tag{4.1}$$

where y_i is the exact robot end-effector pose and n is the number of observations. The residuals between the observations and model are

$$e_i = y_{i,measured} - f(q_i, p) \qquad i = 1,2,\ldots,n \tag{4.2}$$

where $y_{i,measured}$ is the vector containing measured pose of the robot at position i.

$$L = \sum_{i=1}^{n} e_i^T e_i \tag{4.3}$$

The least squares estimation of parameter vector, p, is the value that minimizes Equation 4.3. Following sections provide different approaches for parameter estimation. In Section 4.1, a direct search algorithm is introduced that does not require any derivative terms; however the rate of convergence is slow, like all direct search algorithms, by their nature. Subsequently, in Section 4.2, gradient method that based on identification Jacobian is presented.

4.1. Direct Search Algorithm

$L:\mathbb{R}^d \rightarrow \mathbb{R}$ is called the cost function, desired to be minimized, and d is the dimension. The name of the algorithm comes from the geometric figure in d dimensions, the simplex. The simplex has $d+1$ vertices are the possible solutions for parameter vector, denoting $p_1, p_2, ..., p_{d+1}$ and the simplex is denoted by Ω.

Nelder-Mead simplex algorithm produces an approximation to the optimal point of L by iterations. At each iteration step, all vertices $\{p_k\}_{k=1}^{d+1}$ are reordered according to the cost function values [42].

$$L(p_1) \leq L(p_2) \leq \ldots \leq L(p_{d+1}) \tag{4.4}$$

Vertex p_1 refers to the best and vertex p_{d+1} refers to the worst. Some vertices have the possibility of same value of the cost function, if this is the case tie-breaking rules should be included into the algorithm [42].

The Nelder-Mead simplex algorithm is based on the four possible operations: reflection, expansion, contraction, and shrink; and all operations is associated with some scalars: $\sim_s$ (reflection), $_s$ (expansion), ¥$_s$ (contraction), ¢$_s$ (shrink). $\sim_s > 0$, $_s > 1$, $0 <$ ¥$_s < 1$, and $0 <$ ¢$_s < 1$.

In [42], the parameters are chosen as;

$$\left[\sim_s,\ {}_s,\text{¥}_s,\text{¢}_s\right]=\left[1,2,0.5,0.5\right] \tag{4.5}$$

The centroid of the best n vertices is calculated using,

$$\overline{p}=\frac{1}{d}\sum_{k=1}^{d}p_k \tag{4.6}$$

Steps in one iteration of the simplex algorithm from the reference [42] are given below;

1. *Sort.* Compute the value of function L at the $p+1$ vertices of Ω and sort the vertices according to;

   $$L\left(p_1\right)\le L\left(p_2\right)\le\ldots\le L\left(p_{d+1}\right) \tag{4.7}$$

2. *Reflection.* Calculate the reflection point p_r from

   $$p_r=\overline{p}+\sim_s\left(\overline{p}-p_{d+1}\right) \tag{4.8}$$

   Evaluate $L_r=L\left(p_r\right)$. If $L_1\le L_r<L_d$, replace p_{d+1} with p_d.

3. *Expansion.* If $L_r<L_1$ then compute the expansion point p_e from

   $$p_e=\overline{p}+\beta_s(p_r-\overline{p}) \tag{4.9}$$

   Evaluate $L_e=L(p_e)$. If $L_e<L_r$, replace p_{d+1} with p_e; otherwise replace p_{d+1} with p_r.

4. *Outside Contraction.* If $L_d\le L_r<L_{d+1}$, compute the outside contraction point

   $$p_{oc}=\overline{p}+\text{¥}_s(p_r-\overline{p}) \tag{4.10}$$

   Evaluate $L_{oc}=L(p_{oc})$. If $L_{oc}\le L_r$, replace p_{d+1} with p_{oc}; otherwise skip to 6.

5. *Inside Contraction.* If $L_r\ge L_{p+1}$, compute the inside contraction point p_{ic} from

$$p_{ic} = \overline{p} - ¥_s(p_r - \overline{p}) \tag{4.11}$$

Evaluate $L_{ic} = L(p_{ic})$. If $L_{ic} < L_{d+1}$, replace p_{d+1} with p_{ic}; otherwise skip to 6.

6. *Shrink.* For $2 \le i \le d+1$, define

$$p_i = p_1 + ¢_s(p_i - p_1) \tag{4.12}$$

4.2. Gradient Methods

The nonlinear model (Equation 4.1) can be linearized using Taylor expansion around the kinematic parameter values till the second derivative term.

$$f(q_i)_{j+1} = f(q_i)_j + \frac{\partial f(q_i)_j}{\partial p}\Delta p \tag{4.13}$$

where j is the initial guess, $j+1$ is the prediction made, where $\Delta p = p_{j+1} - p_j$. By combining this equation with observation vector [33];

$$y_i - f(q_i)_j = \frac{\partial f(q_i)_j}{\partial p}\Delta p \tag{4.14}$$

In matrix form;

$$D = J(q,p)\Delta p \tag{4.15}$$

where J is the parameter Jacobian matrix of partial derivatives of the objective function, with respect to variables, evaluated at the initial guess j.Utilizing linear least-squares theory, concluded in Equation 4.16 [40];

$$(J^T J)\Delta p = J^T D \tag{4.16}$$

Equation can be solved for parameter deviations through matrix inversion;

$$\Delta p = (J^T J)^{-1} J^T D \qquad (4.17)$$

Steps till Equation 4.17 are referred as Gauss-Newton algorithm in which matrix inversion is not possible due to singularity and ill-conditions. Subsequently, desired parameter deviation vector could not be obtained.

A more robust approach for matrix inversion was developed by Levenberg [33] and Marquardt [34], independently. Tull and Lewis [44] and Chen and Chan [43] have successfully adopted this algorithm to synthesis of mechanisms.

In the Levenberg-Marquardt algorithm, equation above is modified as;

$$\Delta p = (J^T J + \text{ª} I)^{-1} J^T D \qquad (4.18)$$

In this equation ª is an adjustable scalar and *I* is an identity matrix with has rows as many as the number of measurements. Choosing a sufficient scalar, ª , inversion becomes possible for parameter estimation.

Iterations are continued till the predefined performance criteria concerning convergence is meet. This algorithm is regarded as the combination of two methods, the steepest descent and Gauss-Newton methods. For a large scalar value at the start, algorithm's behavior is closed to the steepest descent method. After successful iterations, convergence to measured data is achieved, scalar value is lowered. Thus, algorithm begins to resemble as the Gauss-Newton method [29].

4.3. Sensor-based Calibration Methods

In this study, most common sensor-based calibration methods are used. In the first approach an external sensor device provides position or location coordinates of the tool frame while the second approach focuses on improving performance of the relative position traveled between two configurations.

4.3.1. General Differential Model of the Methods

Robust approaches are performed using a linearized model that is based on low geometric errors. The generalized differential model of the robot maps differential change in geometric parameters to the differential change in the end-effector pose. Simply denoted by [40];

$$\begin{bmatrix} dP^T\left(y,q,p\right) \\ dR^T\left(y,q,p\right) \end{bmatrix} = J\left(q,p\right)\Delta p + \circ \tag{4.19}$$

The columns corresponding to the parameters of the generalized identification Jacobian matrix J relative to the reference frame can be computed symbolically using following expressions, using alternative robust approaches resultant matrix has been checked [38, 40, 41];

$$J\left(q,p\right) = \left[J_a\left(q,p\right) \;\; J_{\sim}\left(q,p\right) \;\; J_d\left(q,p\right) \;\; J_{-}\left(q,p\right) \;\; J\;\;\left(q,p\right) \right] \tag{4.20}$$

Here, each column corresponds to the derivatives relative to the each parameter assuming that all parameters exist in the model and all are identifiable. For instance, Equation 4.21 shows derivatives of each model output (3 position and 3 orientation components) relative to the $a = \left[a_1, a_2, \ldots a_8\right]$ parameters.

$$J_a\left(q,p\right)=\begin{bmatrix}\frac{\partial f_1}{\partial a_1} & \frac{\partial f_1}{\partial a_2} & \cdots & \frac{\partial f_1}{\partial a_8}\\ \frac{\partial f_2}{\partial a_1} & \frac{\partial f_2}{\partial a_2} & \cdots & \frac{\partial f_2}{\partial a_8}\\ \vdots & \vdots & \cdots & \vdots\\ \frac{\partial f_6}{\partial a_1} & \frac{\partial f_6}{\partial a_2} & \cdots & \frac{\partial f_6}{\partial a_8}\end{bmatrix} \tag{4.21}$$

In order to obtain generalized identification Jacobian matrix J same procedure is applied to $\left[\sim_1,\sim_2,\ldots\sim_8,d_1,d_2,\ldots d_8,\bar{}_1,\bar{}_2,\ldots\bar{}_8,\ _1,\ _2,\ldots\ _8\right]$ parameters.

In order to solve the system and estimate Δp, a set of sufficient number of configurations $Q=\left\{q_1,....,q_m\right\}$ is applied, equation become [38, 40];

$$\begin{bmatrix}\Delta P^1(Y^1,Q^1,p)\\ \Delta R^1(Y^1,Q^1,p)\\ \vdots\\ \Delta P^m(Y^m,Q^m,p)\\ \Delta R^m(Y^m,Q^m,p)\end{bmatrix}=\Phi(Q,p)\Delta p+{}^\circ \tag{4.22}$$

Equation 4.22 also figures out how the observation matrix Φ is obtained, which will be used for determination of identifiable parameters (see Section 4.3.4). ° is the generic presentation of un-modeled error factors and modeling errors. Observation matrix is simply shown as;

$$\Phi=\begin{bmatrix}J^1(q^1,p)\\ \vdots\\ J^m(q^m,p)\end{bmatrix} \tag{4.23}$$

Sections 4.3.2 and 4.3.3 introduces details on parameter identification procedure for different pose components.

4.3.2. Using Endpoint Pose Measurements

Difference between the theoretical and measured terminal frame location (or only position) is intended to be minimized [38].

$$\begin{bmatrix} \Delta P^T(y,q,p) \\ \Delta R^T(y,q,p) \end{bmatrix} = 0 \tag{4.24}$$

Here ΔP is the (3x1) vector of the position error between measurement and model;

$$\Delta P = P^T(y) - P^T(q,p) \tag{4.25}$$

and ΔR is the (3x1) vector of orientation error;

$$\Delta R = u\sim \tag{4.26}$$

Here u and ~ could be calculated using;

$$R^T_{measured} = Rot(u,\sim)R^T_{computed} \tag{4.27}$$

$R^T_{measured}$ is the measured (3x3) orientation matrix and $R^T_{computed}$ is the computed orientation matrix using forward model.

The linear differential model (similar to Equation 4.19) defining the deviation of the end-effector's pose in terms of differential errors in geometric parameters [38];

$$D = \begin{bmatrix} \Delta P^T(y,q,p) \\ \Delta R^T(y,q,p) \end{bmatrix} = J(q,p)\Delta p \tag{4.28}$$

In this equation, $J(q,p)$ is the generalized Jacobian matrix so that if sensor provides only position measurements, first three rows of Equation 4.21 and 4.24 are used for identification algorithm [38].

Since position and orientation measurements have different units for representing the physical magnitude and different accuracy, theory implements a scale matrix w in order to compare them.

$$\Delta p = (J^T wJ)^{-1} J^T wD \tag{4.29}$$

Matrix w is consisting of standard deviations ,± , provided by the measurements system or calculated by the appropriate theory. For $i = 1,...,n$ number of measurements and c components [38];

$$R^i = \begin{bmatrix} \pm_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \pm_c^2 \end{bmatrix} \tag{4.30}$$

$$w = \begin{bmatrix} R^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R^n \end{bmatrix}^{-1} \tag{4.31}$$

4.3.3. Using Relative Position Measurements

The relative position constituted by the endpoint of the robot moving between two sequential configurations is utilized. Assuming the position of the manipulator calculated using nominal parameters is $P^T(q,p)$, and the real position of the manipulator is $P^T(y)$, relation between them is expressed as;

$$P^T(y) = P^T(q,p) + dP \tag{4.32}$$

Small errors in the position coordinates of the manipulator can be modeled as a function of small changes in the kinematic parameters, as in the following equation [47]:

$$dP = \begin{bmatrix} dP_x \\ dP_y \\ dP_z \end{bmatrix} = \begin{bmatrix} J_a^{\ p} J_{\sim}^{\ p} J_d^{\ p} J_{-}^{\ p} J^{\ \ p} \end{bmatrix} \begin{bmatrix} \partial a \\ \partial \sim \\ \partial d \\ \partial^{-} \\ \partial \end{bmatrix} \tag{4.33}$$

The equation of first position command a and second position command b can be written as;

$$P^T\left(y_a\right) = P^T\left(q_a, p\right) + dP_a \tag{4.34}$$

$$P^T\left(y_b\right) = P^T\left(q_b, p\right) + dP_b \tag{4.35}$$

Small differences of the manipulator position coordinates between two sequential robot configurations can be written as [47];

$$dP_a - dP_b = \left(P^T\left(y_a\right) - P^T\left(y_b\right)\right) - \left(P^T\left(q_a, p\right) - P^T\left(q_a, p\right)\right) \tag{4.36}$$

Where; $dP_a - dP_b$ is the relative position error, $P^T\left(y_a\right) - P^T\left(y_b\right)$ is the real robot's relative position, and $P^T\left(q_a, p\right) - P^T\left(q_a, p\right)$ is the nominal robot relative position.

Consequently the linear model can be written as;

$$D = J_{a-b}^{p}(p)\Delta p \tag{4.37}$$

where;

$$D = dP_a - dP_b = \left(P^T\left(y_a\right) - P^T\left(y_b\right)\right) - \left(P^T\left(q_a, p\right) - P^T\left(q_a, p\right)\right) \tag{4.38}$$

$$J_{a-b}^{p}(p) = [J_a^{\ p} - J_b^{\ p}] \tag{4.39}$$

$$\Delta p = \begin{bmatrix} \partial a \ \partial\sim \ \partial d \ \partial^{-} \ \partial \ \ \end{bmatrix}^T \tag{4.40}$$

4.3.4. Identifiability of the Geometric Parameters

When the theory is tried to be applied to a case robot, it is clear that some parameters do not have effect on calibration model and effect of some parameters may be grouped with others. An exercise may clarify that identification Jacobian matrix can have linearly dependent columns and corresponding parameters for that columns vary arbitrarily satisfying the linear dependence. This situations can be examined by calculating the rank of the identification Jacobian matrix, J, this procedure is outlined by Khalil et al. [38]. After applying sufficient number of random configurations $Q=\{q_1,....,q_m\}$, observation matrix Φ is obtained;

$$\begin{bmatrix} \Delta P(Y,Q,p) \\ \Delta R(Y,Q,p) \end{bmatrix} = \Phi(Q,p)\Delta p \tag{4.41}$$

Unidentifiability of some parameters is originated from zero and a linearly dependent column of the observation matrix and these parameters depend on the notation and measurements components used for the model. Determination of unidentifiable parameters starts with determining the zero columns of Φ; by eliminating these parameters, the number of the columns of observation matrix is reduced. Although, matrix dimension is reduced with some order, in the context same symbol is used in order to keep unity [38].

The rank of Φ determines the number of independent columns, an efficient set of identifiable parameters should be chosen as the maximum number of parameters in set is equal to the rank. This process can be formulated as determination of zero and independent columns using QR decomposition of the observation matrix Φ [38];

$$\Phi = Q_{\Phi} \begin{bmatrix} R_{\Phi} \\ 0_{(r-c)\times c} \end{bmatrix} \tag{4.42}$$

Orthogonal matrix of (rxr) and upper triangular matrix of (cxc), Q_{Φ} and R_{Φ} respectively, and the null matrix of (r-c)xc are obtained [38]. In theory, the non-identifiable parameters are the numeric zero diagonal elements of the upper triangular matrix R_{Φ} [38]. In the identification step, the parameters are already defined in robot controller have the priority. Starting from the parameters with high priority, a set of parameters is chosen to continue with estimation algorithm. This step clearly figures out that the set of identifiable parameters are not unique.

In literature, singular value decomposition (SVD) technique is alternatively used instead of QR decomposition. Similarly, technique is applied to set of equations to determine numerical singularity. Theory decomposes the matrix Φ in to a product of orthogonal matrix U, diagonal matrix S and conjugate transpose of V (see Equation 4.43). Here again, zero off diagonal elements diagonal matrix S is corresponding to singular elements of Φ. Details of this SVD procedure for robot model identification and discussion could be found in [53].

$$\Phi_{r\times c} = U_{r\times c} S_{c\times c} V^{T}_{c\times c} \tag{4.43}$$

When these steps are applied to the case robot FANUC Robot R-2000*i*B/210F for location measurements, parameter vector p reduces to;

$$p = [a_1, \ldots, a_T, \sim_1, \ldots, \sim_T, d_1, d_2, d_4, \ldots, d_T, \overline{\ }_1, \ldots, \overline{\ }_T, \ {}_3]^T \tag{4.44}$$

This could be inferred that effects for manipulator error of the parameters $\overline{\ }_R$ and d_R are regrouped with $\overline{\ }_1$ and d_1, respectively. In addition, in the case of position measurements, $\overline{\ }_T$, $\sim_T$ and d_T are not included into the identification step in simulations.

CHAPTER 5

SIMULATION SOFTWARE

In this study, simulation software has been developed that reflecting the theory introduced so far. To do this, the object-oriented programming approach has been utilized due to its advantages such as flexibility, maintainability, data integration and expandability. The object-oriented paradigm approximates a program as an accumulation of discrete objects that are data structures and methods showing relations.

Responsibilities are distributed as Model-View-Controller type. View present the models, take the information from model in order to visualize it in textual and/or graphical forms, transmit prompts to controller and render view in response to controller demands [51]. Model is the physical and mathematical representation of system and process so it responds to state queries. Controller designates the behavior of the software, interact user commands to model and selects the render type of the output [52].

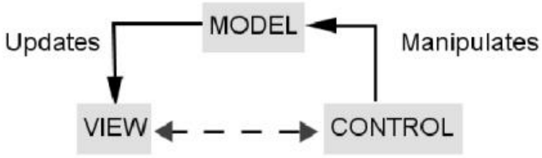
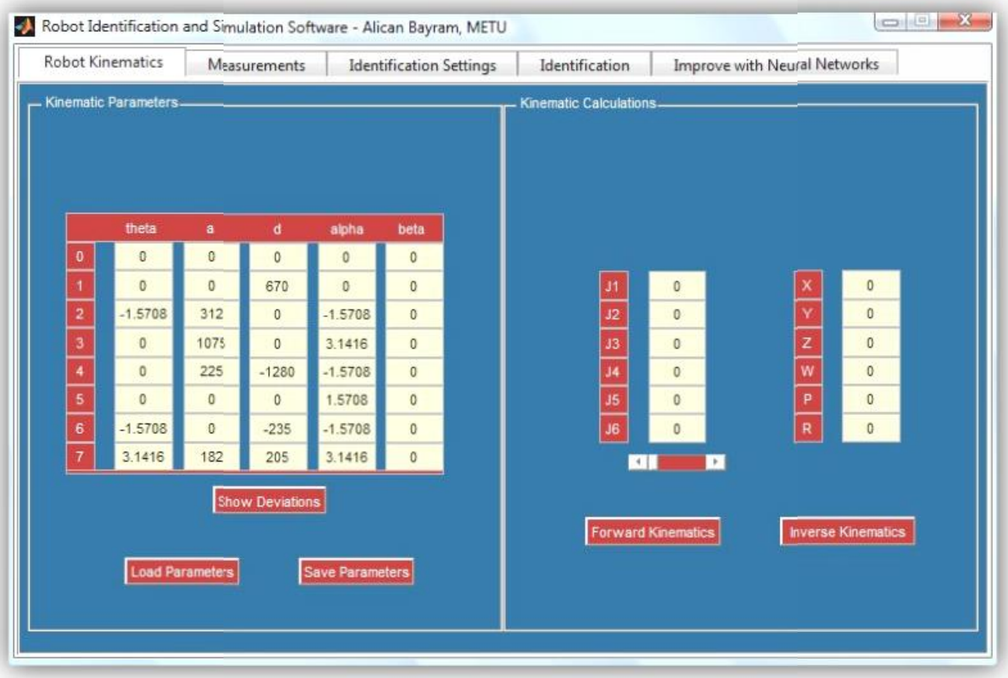


Figure 5.1: The Model-View-Controller (MVC) Topology

The software has a functional graphical user interface consisting of module tabs that represents each step in the calibration procedure. All components are explained in this section.

5.1. Robot Kinematics Tab

The robot kinematic tab is divided into two parts. By using the left hand portion, kinematic parameters that would be used for the model development are displayed. Kinematic parameters could be specified by entering as well as by loading them from a saved file. Creating a parameter file for future use and updating an existent parameter file could be done by save parameters button (see Figure 5.2). In addition, deviations for each parameter could be specified for building a virtual robot that is used in simulations.



	theta	a	d	alpha	beta
0	0	0	0	0	0
1	0	0	670	0	0
2	-1.5708	312	0	-1.5708	0
3	0	1075	0	3.1416	0
4	0	225	-1280	-1.5708	0
5	0	0	0	1.5708	0
6	-1.5708	0	-235	-1.5708	0
7	3.1416	182	205	3.1416	0

J1	0	X	0
J2	0	Y	0
J3	0	Z	0
J4	0	W	0
J5	0	P	0
J6	0	R	0

Figure 5.2: Robot Kinematics Tab

The right hand portion is consisting of spaces for joint angles and end-effector coordinates. For forward kinematic calculations, after specifying kinematic parameters, joints angles (in degrees) are entered from one to six and forward kinematics button is pushed. Consequently, end-effector location will be displayed as three positions (in millimeters) and three rotation angles (in degrees). Reversely, inverse calculation can be processed by entering end-effector location and pushing inverse kinematics button. Obviously, inverse kinematics produce multiple solutions, all of these solutions can be accessible by changing the slider position.

5.2. Measurements Tab

The left hand portion of the measurements tab is specially designed for display purposes. Table shows data concerning joint angles, nominal and measured end-effector coordinates, relative distances and generated noise (see Figure 5.4).

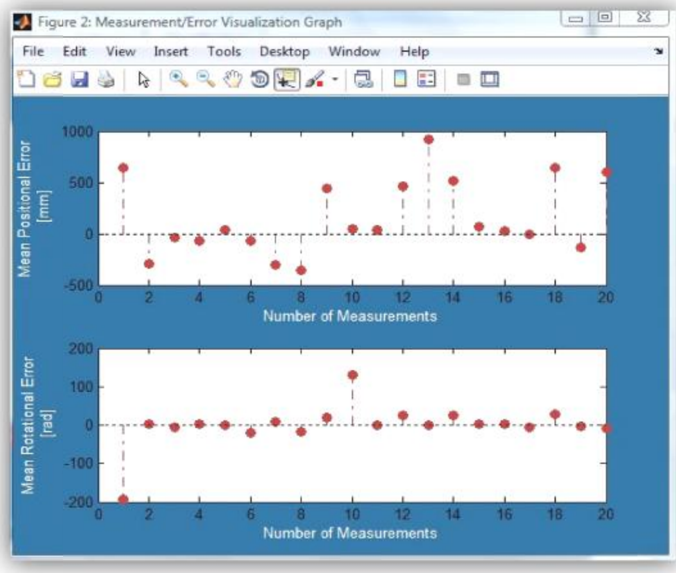
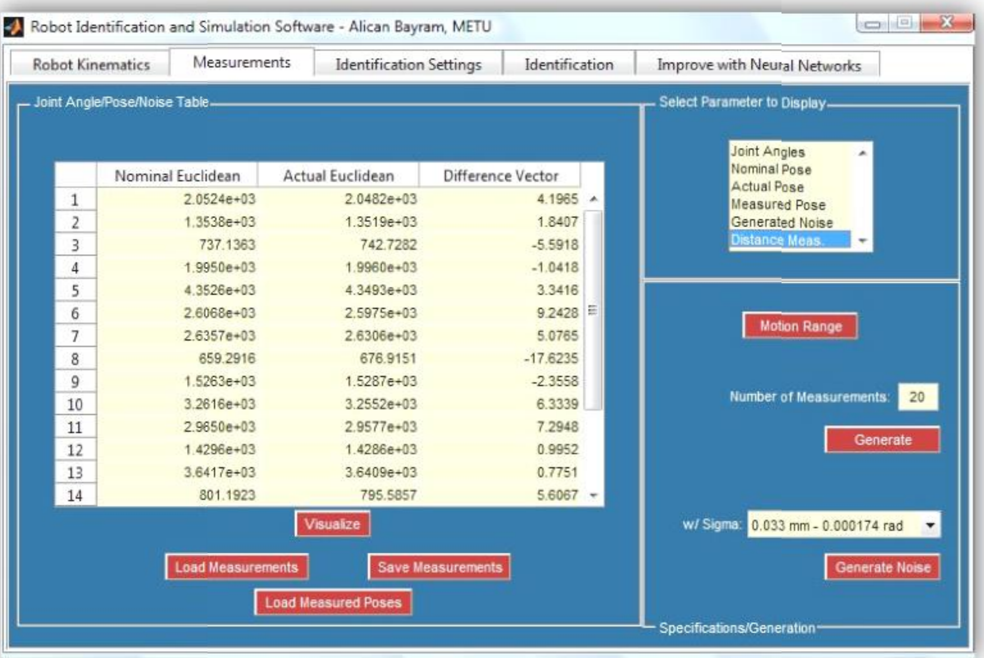


Figure 2: Measurement/Error Visualization Graph
File Edit View Insert Tools Desktop Window Help
Mean Positional Error [mm]
1000
500
0
-500
0 2 4 6 8 10 12 14 16 18 20
Number of Measurements
Mean Rotational Error [rad]
200
100
0
-100
-200
0 2 4 6 8 10 12 14 16 18 20
Number of Measurements

Figure 5.3: Measurement Visualization

Measurements could be calculated automatically by entering joint angles as well as by loading them from a saved file. For simulation purposes virtual measurements could be generated by using nominal parameters and parameter deviations defined in robot kinematics tab. Joint angles are produced randomly within the lower and upper limits of the joint angles as specified by the robot manufacturer. These values are asked via a dialog box before generating joint angles. Generated virtual measurements could also contain noise, this factor is simulated according to the standard deviations supplied by the measurements system.



Robot Identification and Simulation Software - Alican Bayram, METU

Joint Angle/Pose/Noise Table

	Nominal Euclidean	Actual Euclidean	Difference Vector
1	2.0524e+03	2.0482e+03	4.1965
2	1.3538e+03	1.3519e+03	1.8407
3	737.1363	742.7282	-5.5918
4	1.9950e+03	1.9960e+03	-1.0418
5	4.3526e+03	4.3493e+03	3.3416
6	2.6068e+03	2.5975e+03	9.2428
7	2.6357e+03	2.6306e+03	5.0765
8	659.2916	676.9151	-17.6235
9	1.5263e+03	1.5287e+03	-2.3558
10	3.2616e+03	3.2552e+03	6.3339
11	2.9650e+03	2.9577e+03	7.2948
12	1.4296e+03	1.4286e+03	0.9952
13	3.6417e+03	3.6409e+03	0.7751
14	801.1923	795.5857	5.6067

Select Parameter to Display: Distance Meas.

Number of Measurements: 20

w/ Sigma: 0.033 mm - 0.000174 rad

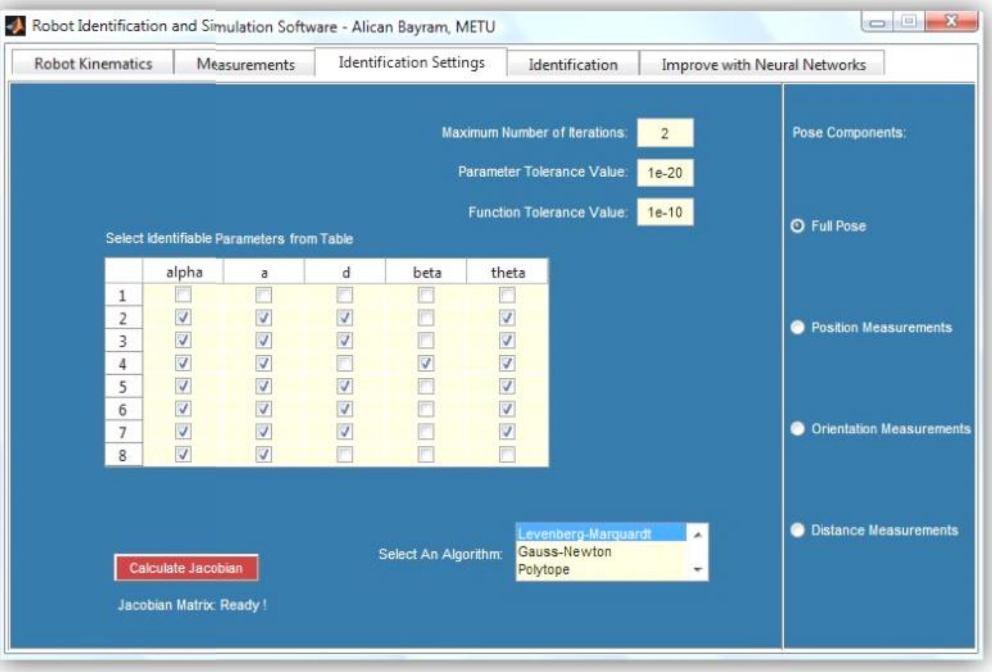
Figure 5.4: Measurements Tab

Controlling the residue between the nominal pose and measured pose is required for some reasons before applying the identification process. Thus, a visualization module is defined in measurements tab. By using this, mean residual errors in position and in orientation coordinates are displayed in pop-up window.

5.3. Identification Settings Tab

Identification settings tab is designed to control identification settings before starting to the identification procedure. On the left hand portion, parameters table is located in order to select parameters that are desired to be included or excluded in the identification. Identifiable parameters are selected by a simple click on the corresponding box. Default settings are defined while software is launching.

Termination criteria are set by defining function tolerance, parameter tolerance and maximum iteration number. When any of these limits are reached, iterations will stopped and results will be displayed. The features concerning termination criteria are inspired by Matlab® function lsqcurvefit through its use and adopted to other algorithms.


Robot Identification and Simulation Software - Alican Bayram, METU
Robot Kinematics
Measurements
Identification Settings
Identification
Improve with Neural Networks
Calculate Jacobian

Maximum Number of Iterations: 2

Parameter Tolerance Value: 1e-20

Function Tolerance Value: 1e-10

Select Identifiable Parameters from Table

	alpha	a	d	beta	theta
1	□	□	□	□	□
2	☑	☑	☑	□	☑
3	☑	☑	☑	□	☑
4	☑	☑	□	☑	☑
5	☑	☑	☑	□	☑
6	☑	☑	☑	□	☑
7	☑	☑	☑	□	☑
8	☑	☑	□	□	□

Select An Algorithm: Levenberg-Marquardt
Gauss-Newton
Polytope

Jacobian Matrix: Ready !

Pose Components:

◉ Full Pose

○ Position Measurements

○ Orientation Measurements

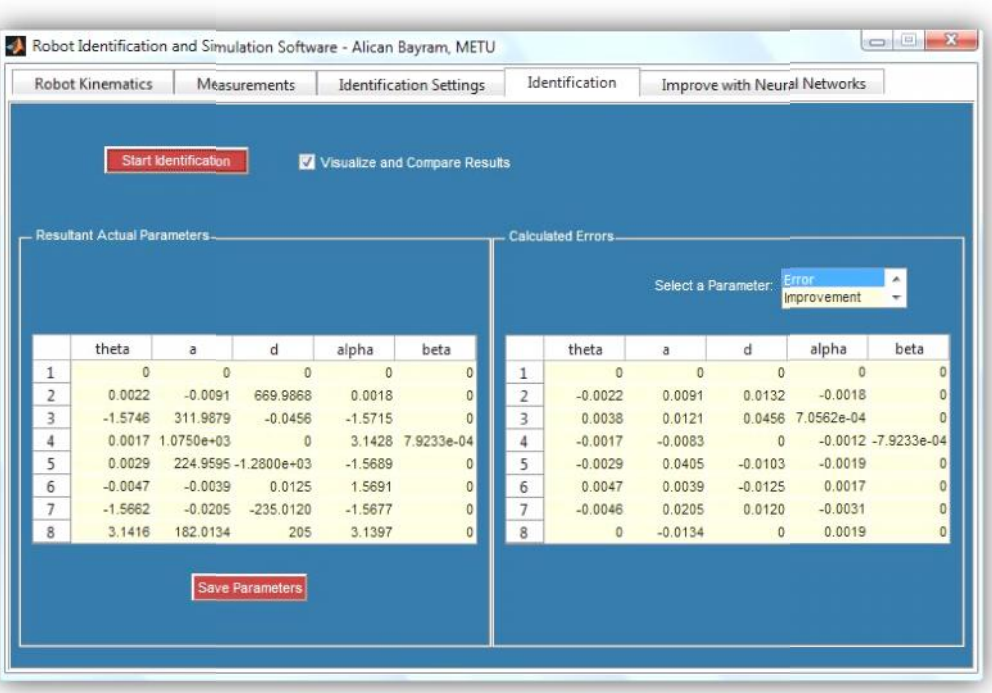
○ Distance Measurements

Figure 5.5: Identification Settings Tab

Pose components could be selected by selecting the appropriate button such as full pose, position measurements, orientation measurement and relative position measurements. To do this, Jacobian matrix need to be ready since pose component selection updates the Jacobian matrix. For identification, three methods could be used. Gauss-Newton algorithm, Levenberg-Marquardt (based on Matlab® function lsqcurvefit) and Nelder-Mead simplex algorithm are available for identification purposes.

5.4. Identification Tab and Result Visualization

In the identification tab, identification is started by pressing the corresponding button. Before identification starts, mean residual errors of position and orientation components are saved for comparison.


Robot Identification and Simulation Software - Alican Bayram, METU
Robot Kinematics
Measurements
Identification Settings
Identification
Improve with Neural Networks
Start Identification
☑ Visualize and Compare Results
Save Parameters

Resultant Actual Parameters

	theta	a	d	alpha	beta
1	0	0	0	0	0
2	0.0022	-0.0091	669.9868	0.0018	0
3	-1.5746	311.9879	-0.0456	-1.5715	0
4	0.0017	1.0750e+03	0	3.1428	7.9233e-04
5	0.0029	224.9595	-1.2800e+03	-1.5689	0
6	-0.0047	-0.0039	0.0125	1.5691	0
7	-1.5662	-0.0205	-235.0120	-1.5677	0
8	3.1416	182.0134	205	3.1397	0

Calculated Errors

Select a Parameter: Error

Improvement

	theta	a	d	alpha	beta
1	0	0	0	0	0
2	-0.0022	0.0091	0.0132	-0.0018	0
3	0.0038	0.0121	0.0456	7.0562e-04	0
4	-0.0017	-0.0083	0	-0.0012	-7.9233e-04
5	-0.0029	0.0405	-0.0103	-0.0019	0
6	0.0047	0.0039	-0.0125	0.0017	0
7	-0.0046	0.0205	0.0120	-0.0031	0
8	0	-0.0134	0	0.0019	0

Figure 5.6: Identification Tab

When the termination criteria are reached, automatically results are displayed on the corresponding tables. Resultant parameters could be saved for further investigations by save parameters button. Consequently, results are plotted by comparing them with the state before the calibration. Mean residual errors figure out how much improvement has been achieved in all measurements individually.

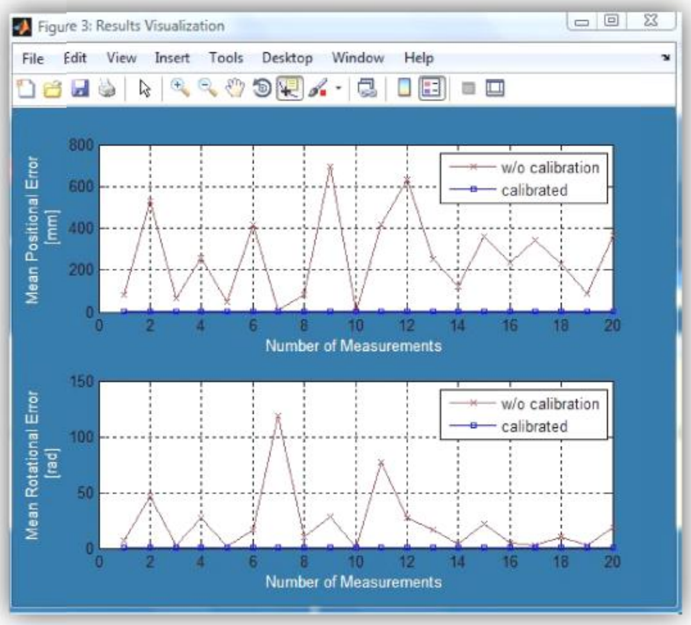


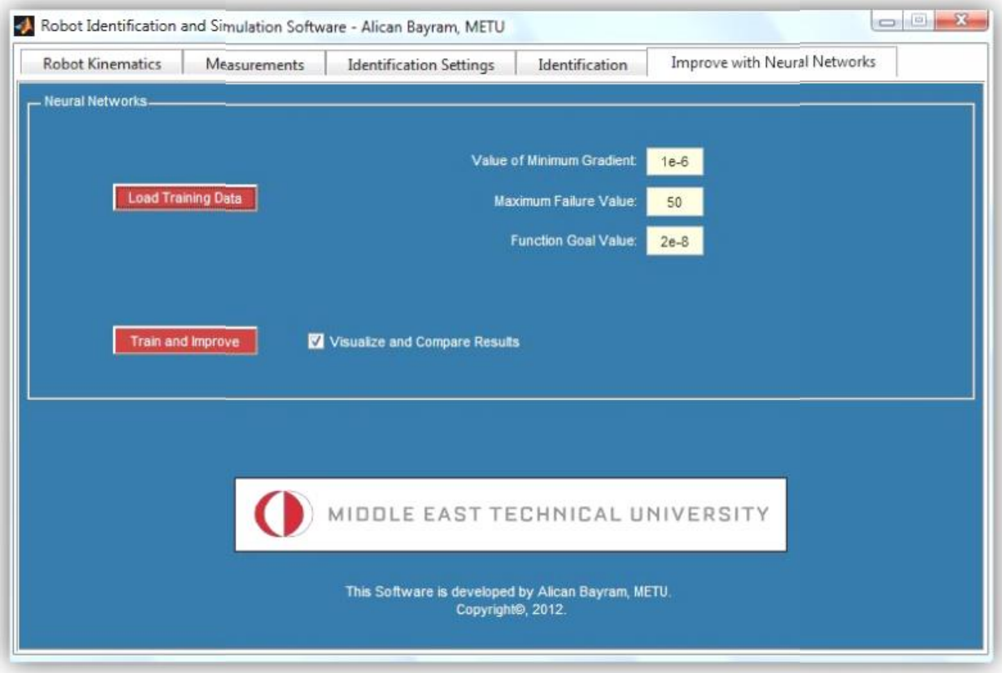
Figure 5.7: Results Visualization

5.5. Neural Networks Tab

In the neural networks tab, network training data could be loaded by pressing the corresponding button. For efficient training of the network, parameters of minimum gradient value, number of the maximum failure and goal value for performance function should be entered (see Figure 5.8). After pressing train and improve button, Matlab Neural Network Toolbox® will train the network till the performance criteria reached.

When the performance criteria are reached, automatically results are displayed in a comparison with nominal model and calibrated model.

Detailed information concerning compensation scheme and neural networks structure, and efficient design parameters are given in Section 3.2 Neuro-accuracy compensation model.


Robot Identification and Simulation Software - Alican Bayram, METU
Robot Kinematics
Measurements
Identification Settings
Identification
Improve with Neural Networks
Neural Networks
Load Training Data
Train and Improve
MIDDLE EAST TECHNICAL UNIVERSITY

Value of Minimum Gradient: 1e-6

Maximum Failure Value: 50

Function Goal Value: 2e-8

☑ Visualize and Compare Results

This Software is developed by Alican Bayram, METU.
Copyright©, 2012.

Figure 5.8: Neural Networks Tab

CHAPTER 6

CASE STUDY RESULTS

6.1. Implementing the Theory

In order to examine the features of the software and the introduced theory, experiment and simulation studies are processed with different pose components. In Case A, virtual position measurements are generated by the developed software and theory is applied on the case robot. Furthermore, in Case B and C, measurements from laser sensor are used in full pose and relative position performance investigations of FANUC Robot R-2000*i*B/210F, respectively. Subsequently, a neuro-accuracy compensation approach is performed in Case D. Primarily parametric zero joint-offset calibration is applied to virtual full-pose measurements, and then neuro-accuracy compensator is applied to enhance the improvement rates.

6.2. CASE A: Position Measurements

In Case A, virtual pose measurements are generated by the developed software. Nominal kinematic parameters of FANUC Robot R-2000*i*B/210F, and generated joints angles within the upper and lower bounds of the joint axes are used for the forward model. Simultaneously, predefined deviations (see Table 6.1) on each parameter are utilized in order to obtain the actual robot model. Consequently, noise that characterizes the measurement system is added with zero mean, and standard deviation, ± , of 33 μm and 0.174 mrad.

Table 6.1: Defined Parameter Deviations for Simulation

Link	$a_i(mm)$	$\sim_i(mrad)$	$d_i(mm)$	$^{-}_i(mrad)$	$_i(mrad)$
R	-	-	-0.02	1.31	-
1	1.41	-1.72	2.23	-1.21	-
2	0.42	-0.82	-0.32	-3.64	-
3	-0.63	-1.32	-	-1.44	0.37
4	2.64	1.63	-2.21	2.73	-
5	-0.85	-1.34	1.63	-3.82	-
6	0.75	3.11	0.82	3.44	-
T	-0.44	-1.26	0.52	-2.11	-

By using only position coordinates of 20 virtual measurements, mean positional error between measurements and robot model is reduced from 4.0715 mm to 0.0179 mm.

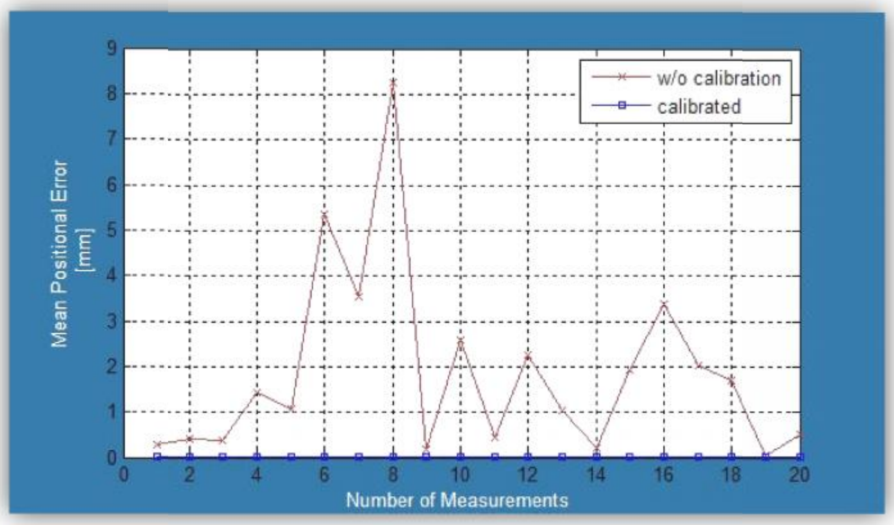


Figure 6.1: Results for Virtual Position Measurements

Estimated parameter errors on each parameter could be seen in Table 6.2. Simulation shows that identification using only position measurements concluded in 99.56% improvement for calibrated robot model (see Figure 6.1).

Table 6.2: Estimated Parameter Errors

Link	a_i(«m)	~$_i$($mrad$)	d_i(«m)	¯$_i$($mrad$)	$_i$($mrad$)
R	-	-	-0.02	1.31	-
1	-2.90	1.72	17.35	-0.11	-
2	-26.21	-0.83	8.03	-3.67	-
3	-4.88	1.32	-	1.40	-0.36
4	22.02	-1.62	24.7494	-2.73	-
5	1.20	1.32	-45.7834	3.78	-
6	6.61	-3.17	-3.4176	-2.08	-
T	-3.34	-1.26	0.52	-2.11	-

Additional 15 virtual measurements are used as the validation set, random configurations throughout the workspace are utilized for this purpose. Mean positional error for validation set is reduced from 4.3464 mm to 0.0279 mm.

6.3. CASE B and CASE C: Full Pose and Relative Position Measurements

6.3.1. Measurement System and Procedure

The measurement system is consisting of Leica LTD500 laser tracker and Leica T-Mac probe. While Leica LTD500 is directed to the robot with its front, Leica T-Mac Probe is mounted on the end-effector frame (Figure 6.2).

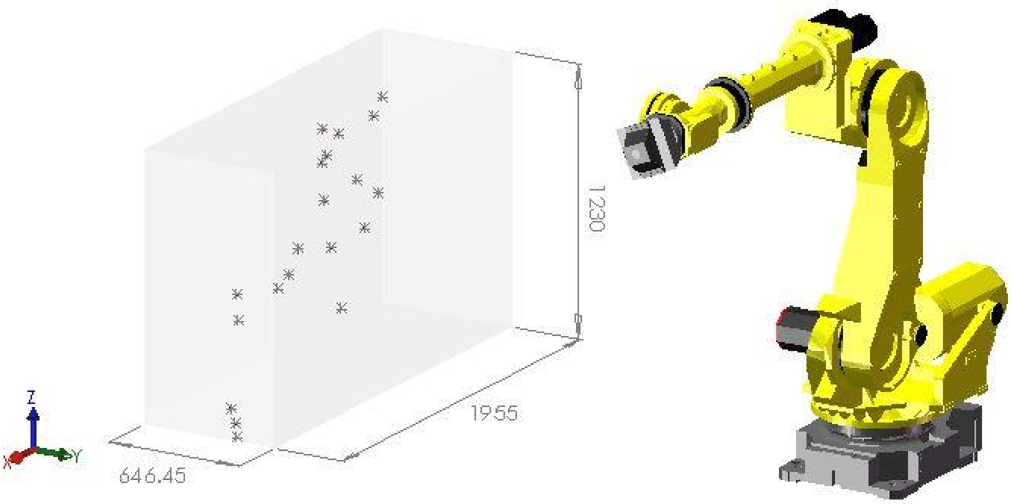


Figure 6.2: FANUC Robot R-2000iB/210F and Measurements in 3D

(CAD Model Courtesy of FANUC Robotics Corp., Dimensions are in mm)

The overall system performs measurements with accuracy of ±10 µm/m for a coordinate and 0.01° for angular components. Repeatability of a coordinate is reported as ±5 µm/m (for all specifications see Table 6.3) [45].

Table 6.3: Technical Specifications of the Measurement System [45]

Accuracy of a coordinate	±10 µm/m
Accuracy of an angle	0.01°
Distance resolution	1.26 µm
Angle resolution	0.14′′
Reproducibility of a coordinate	± 5 µm/m

Robot's base frame is attended as the reference frame during the measurements. Thus parameters of the eight frames are included into the error model including tool and base errors. The robot is moved to different configurations by inputs for all axes and laser tracker measures the tool frame locations. The mean error at a certain location is calculated by taking mean value of the modulus of the difference between actual and nominal values, individually in position and orientation components.

Three reference holes are projected on the ground surface; consequently the robot's base is registered as the reference. Detailed drawings are given in Appendix A.

6.3.2. Results

In Case B, full pose measurements are taken from FANUC Robot R-2000*i*B/210F and measurement system. Since workspace occupies a large volume, region where the 20 measurements are selected covers the robot's trajectory for the production (Figure 6.2). Estimated errors are given in Table 6.4.

Table 6.4: Estimated Errors of the Case Robot

Link	a_i(«m)	~$_i$($mrad$)	d_i(«m)	¯$_i$($mrad$)	$_i$($mrad$)
R	0	0	0	0	0
1	-86.7330	1.7125	93.7855	0.1055	0
2	-38.8046	-0.8294	47.4197	-3.5695	0
3	9.3449	1.5056	0	1.4041	-0.3277
4	60.4734	-2.0319	62.3239	-2.9721	0
5	162.4505	1.6874	52.1116	3.8052	0
6	-36.2985	-3.4485	-95.3920	-3.3286	0
T	-13.1092	1.3194	55.1706	2.1314	0

By using position and orientation coordinates of measurements, mean positional error over measurements is reduced from 3.2970 mm to 0.0182 mm and mean orientation error is reduced from 0.5503° to 0.0039°. Results show that identification using full pose measurements concluded in 99.45% improvement in position and 99.29% improvement in rotation of components of the robot model (Figure 6.3).

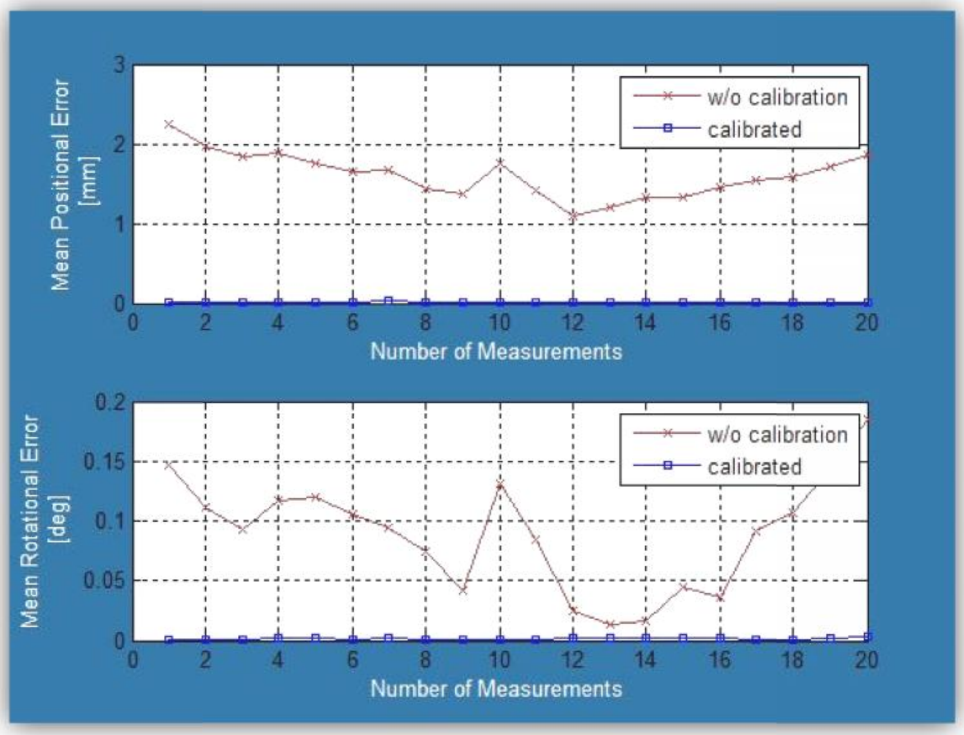


Figure 6.3: Results for Position and Orientation Measurements

In Case C, given sequential configurations for FANUC Robot R-2000*i*B/210F and utilizing relative position coordinates of two sequential configurations, 15 relative position measurements are used to find the actual robot geometric parameters. Using the identified kinematic parameters, a set of validation data is used for confirmation of improvement. By knowing relative position coordinates, results are compared in the form of Euclidean distances.

Table 6.5: Relative Distance Errors in Validation Set (Dimensions are in mm)

Meas. Number	Before Identification	After Identification
1	2.0807	0.1085
2	1.0339	0.1241
3	2.5221	0.1784
4	3.7178	0.0299
5	5.2482	0.0221
6	1.0029	0.0278
7	4.5608	0.1870
8	2.4933	0.1073
9	1.1598	0.0632
10	3.2043	0.0701
RMS	3.0457	0.1079

By using relative position measurements, root mean square of error in relative positioning is reduced from 3.0457 mm to 0.1079 mm. Results show that identification using relative position measurements between pulses concluded in 96.46% improvement in positioning performance. Improvement can be seen in validation set measurements in Table 6.5.

6.4. CASE D: Neuro-Parametric Calibration

6.4.1. Simulation Procedure

In order to examine the introduced theory on Neural Network compensator, simulation study is processed with using full pose components. For parametric calibration, 30 virtual full pose measurements are generated by the developed software (see Figure 6.4). Nominal kinematic parameters of FANUC Robot R-2000*i*B/210F, and generated joints angles within the upper and lower bounds of the joint axes are used for the forward model. Simultaneously, predefined deviations on each parameter and noise are utilized in order to obtain the virtual full pose measurements; same procedure is followed as in Case A. An additional set of 150 random configurations in the focused region of the workspace (see Figure 6.4) is generated by following the same procedure.

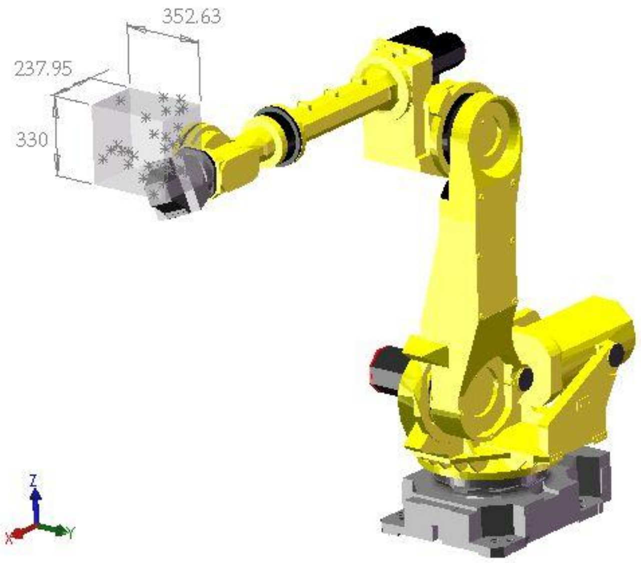


Figure 6.4: FANUC Robot R-2000iB/210F and Virtual Measurements in 3D (CAD Model Courtesy of FANUC Robotics Corp., Dimensions are in mm)

6.4.2. Results

In parametric calibration; zero joint offset calibration, only joint readings, is performed. Mean positional error of measurements is reduced from 6.7929 mm to 0.2183 mm, and mean rotational error of measurements is reduced from 0.8386° to 0.2585°. Since parameter pre-defined deviations include joint offset errors as well as the remaining geometric errors (e.g. effective link length), it is clearly expected that compensation will reduce residual error in both position and orientation.

After performing training stage with different set of configurations, neural network model is obtained for use in cooperation with nominal forward model. Calibration scheme in Section 3.2 is applied to the case robot, and a further improvement is achieved from 0.2183 mm to 0.0336 mm in position and 0.2585° to 0.0160° (see Figure 6.5).

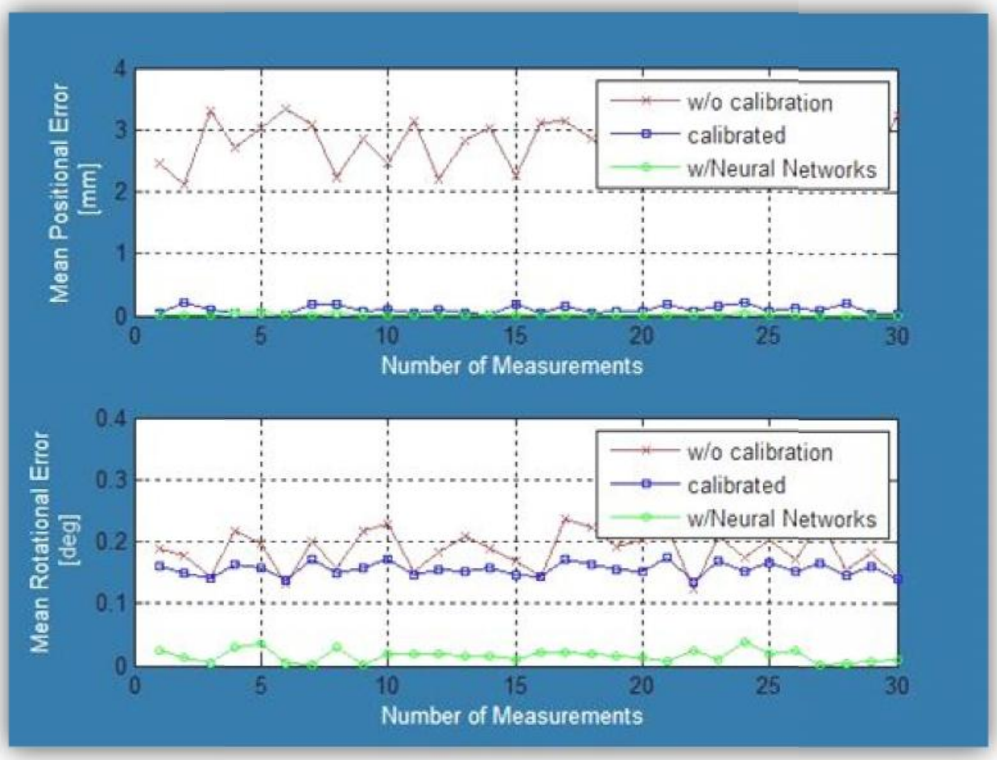


Figure 6.5: Results for Parametric and Neural Network Approaches (in comparison)

These results prove that neural network compensation is useful when the parametric calibration rates are low. Simulation results show that enhancement obtained by NN is the residual error of zero joint offset calibration, which is originated from errors in geometric parameters other than joint offset.

In real cases, end-effector errors of an actual robot are affected by geometrical and non-geometric errors. Geometric errors could be compensated via parameter identification while non-geometric errors could be compensated by on-line neural network approach.

Table 6.6: Results (Dimensions are in mm and degrees)

Methodology	Positional Error	Rotational Error	Improvement Percentage (%)	
Uncalibrated	6.7929 mm	0.8386°	Position	Rotation
Zero joint offset	0.2183 mm	0.2585°	96.79	69.18
Compensator	0.0336 mm	0.0160°	99.51	98.09

CHAPTER 7

DISCUSSION AND CONCLUSION

The aim of this study is to propose an approach in order to increase the accuracy of an articulated robot arm. Proposed approach is applied with relative position measurements, only position measurements and full pose (position and orientation) measurements. To do this, end-effector locations are measured via high performance laser tracker system on the production line. Full pose and relative distance measurements, from experimental robot, are successfully adapted to the theory; and 99.45% in position and 99.29% in orientation, and 96.46% relative position errors between model and measurements, respectively in Case Study B and C, are reduced.

After theory is introduced, simulation software reflecting proposed theory has been developed by using object-oriented strategy. Developed software provides reduction in accuracy errors with relative position measurements, position measurements, orientation measurements and full pose (position and orientation) measurements. Forward and inverse models are obtained for the approach using related modules. In the identification step, gradient methods and direct search algorithms are proposed. Results visualization is a useful feature of the software for those who are inexperienced with the theory. Simulation software enables to perform identification of kinematic parameters via generating virtual full pose, partial pose and relative position measurements.

By using virtual position measurements, 99.56% improvement in the accuracy is achieved in Case Study A. Subsequently, neuro-accuracy compensator module of the software is applied to virtual full pose measurements and results are given in comparision with zero joint offset calibration results. When improvement rates using parametric calibration were 96.79% in position error and 69.18% orientation error, neural network compensator enhance both 99.51% improvement in position error and 98.09% improvement in orientation error are achieved.

As a future study, intelligent techniques (such as genetic algorithm) for parameter identification step are to be implemented. These techniques repeal the gradient matrix and will ease the process as in the application of Nelder-Mead simplex algorithm that has been introduced. Moreover, improving accuracy with trending pure modeless calibration methods would be valuable contribution for the literature.

REFERENCES

[1] M. H. Chang, "The flexible manufacturing, uncertain consumer tastes, and strategic entry deterrence", The Journal of Industrial Economics, Volume XLI, No:1, pp.77-90, 1993

[2] P. Kostal, K. Velisek, "Flexible manufacturing system", World Academy of Science, Engineering and Technology 77, pp.825-829, 2011

[3] P. S. Shiakolas, K. L. Conrad and T. C. Yih, "On the accuracy, repeatability, and degree of influence of kinematics parameters for industrial robots", International Journal of Modelling and Simulation, Vol. 22, No.2, pp.1-10, 2002

[4] TSE, "TS EN ISO Standard 9283", 2000

[5] M. Abderrahim, A. Khamis, S. Garrido and L. Moreno, "Accuracy and Calibration issues of industrial manipulators", Industrial Robotics: Programming, Simulation and Applications, pp.131-146, 2007

[6] J. M. S. T. Motta, "Robot calibration: Modeling measurement and applications" Industrial Robotics: Programming, Simulation and Applications, pp.107-131, 2006

[7] Z. S. Roth, B. W. Mooring and B. Ravani, "An overview of robot calibration", IEEE Journal of Robotics and Automation, Vol. 3, No. 5, pp. 377-385, 1987

[8] D. E. Whitney, C. A. Lozinski and J. M. Rourke, "Industrial robot forward calibration method and results", Journal of Dynamic Systems, Measurement, and Control, 108, pp. 1-8, 1986

[9] C. Gong, J. Yuan and J. Ni, "Nongeometric error identification and compensation for robotic system by inverse calibration", International Journal of Machine Tools and Manufacture 40, pp.2119-2137, 2000

[10] R. P. Judd and A. B. Knasinski, "A technique to calibrate industrial robots with experimental verification", In Proceedings of 1987 IEEE International Conference on Robotics and Automation, pp.351-357, 1987

[11] K. L. Conrad, P. S. Shiakolas and T. C. Yih, "Robotic calibration issues: accuracy, repeatability and calibration" Proceedings of the 8th Mediterranean Conference on Control and Automation (MED 2000), 2000

[12] W. K. Veitschegger and C. H. Wu, "Robot calibration and compensation" IEEE Journal of Robotics and Automation, Vol. 4, No.6, pp.643-656, 1988

[13] J. L. Caenen and J. C. Angue, "Identification of geometric and non geometric parameters of robots", Proceedings of the IEEE International Conference on Robotics and Automation, pp.1032-1037, 1990

[14] B. W. Mooring and S. S. Padavala, "The effect of kinematic model complexity on manipulator accuracy", In Proceedings of the IEEE International Conference on Robotics and Automation, pp.593-598, 1989

[15] J. Chen and L. M. Chao, "Positioning error analysis for robot manipulators with all rotary joints", IEEE Journal of Robotics and Automation, Vol. RA-3, No. 6, 1987

[16] J. H. Jang, S. H. Kim and Y. K. Kwak, “Calibration of geometric and non-geometric errors of an industrial robot”, Cambridge University Press Robotica (2001), Vol. 19, pp.311-321, 2001

[17] J. S. Shamma and D. E. Whitney, “A method for inverse robot calibration”, Journal of Dynamic Systems, Measurement, and Control, Vol. 109, pp.36-43, 1987

[18] L. Everett, M. Driels and B. Mooring, “Kinematic modelling for robot calibration”, In Proceedings of the 1987 IEEE International Conference on Robotics and Automation, Vol. 1, pp. 183-189, 1987

[19] L. Everett and T. Hsu, “The theory of kinematic parameter identification for industrial robots” Journal of Dynamic Systems, Measurement, and Control, Vol. 110, pp-96-100, 1988

[20] H. Stone and A. Sanderson, “A prototype arm signature identification system”, In Proceedings of IEEE Conference of Robotics and Automation, pp. 175-182, 1987

[21] M. Abderrahim and A. R. Whittaker, “Kinematic model identification of industrial manipulators”, Robotics and Computer Integrated Manufacturing , Vol. 16, pp. 1-8, 2000

[22] R. He, Y. Zhao, S. Yang and S. Yang, “Kinematic-parameter identification for serial-robot calibration based on POE formula”, IEEE Transactions on Robotics, Vol. 26, No. 3, 2010

[23] J. Denavit and R. S. Hartenberg, “A kinematic notation for lower-pair mechanisms based on matrices”, ASME Journal of Applied Mechanics, Vol. 77, pp.215-221, 1955

[24] S. A. Hayati and M. Mirmirani, "Improving the absolute positioning accuracy of robot manipulators", Journal of Robotics Systems, Vol. 2, pp.397-413, 1985

[25] Y. Bai and H. Zhuang, "Modeless robots calibration in 3D workspace with an on-line fuzzy interpolation technique", IEEE International Conference on Systems, Man and Cybernetics, pp. 5233-5239, 2004

[26] J. U. Dolinsky, I. D. Jenkinson and G. J. Colquhoun, "Application of genetic programming to the calibration of industrial robots", Elsevier Computers in Industry, Vol. 58, pp.255-264, 2007

[27] J. M. Hollerbach and C. W. Wampler, "The calibration index and taxonomy of kinematic calibration methods", Introduction Journal of Robotics Research, Vol. 15, pp.573-591, 1996

[28] M. Vincze, J. P. Prenninger and H. Gander, "A laser tracking system to measure position and orientation of robot end effectors under motion", International Journal of Robotics Research, Vol. 13, pp.305-314, 1994

[29] A. Goswami, A. Quaid and M. Peshkin, "Identifying robot parameters using partial pose information", IEEE International Conference on Systems, Man and Cybernatics, pp.6-14, 1992

[30] M. R. Driels and W. E. Swayze, "Automated partial pose measurement system for manipulator calibration experiments", IEEE Transaction of Robotics and Automation, Vol. 10, pp.430-440, 1994

[31] M. Lee, D. Kang, Y, Cho, Y. Park and J. Kim, "The effective kinematic calibration method of industrial manipulators using IGPS", International Joint Conference, pp.5050-5062, 2009

[32] W. Khalil and S. Besnard, “Geometric calibration of robots with flexible joints and links”, Journal of Intelligent and Robotics Systems, Vol. 34, pp.357-379, 2002

[33] K. Levenberg, “A method for the solution of certain problems in least squares”, Quarterly of Applied Mathematics, Vol. 2, pp.164-168, 1944

[34] D. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters”, SIAM Journal of Applied Mathematics, Vol. 11, pp.431-441, 1963

[35] T. Monica, L. Giovanni, M. Pierluigi and T. Diego, “A closed-loop neuro-parametric methodology for the calibration of a 5 DOF measuring robot”, Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp.1482-1487, 2003

[36] J. J. Craig, “Introduction to robotics: Mechanics and control”, Pearson Education International, 2005

[37] R. P. Paul, ”Robot Manipulators: Mathematics, programming and control”, MIT Press, 1984

[38] W. Khalil and E. Dombre, “Modeling, identification and control of robots”, Taylor Francis, 2002

[39] FANUC, “FANUC Robot R-2000*i*B mechanical unit: Operator’s manual”, B-82234EN/08, 2007

[40] W. Khalil, J. L. Caenen and C. Endueheard, “Identification and calibration of the geometric parameters of robots”, First International Symposium on Experimental Robotics, pp.528-538, 1989

[41] J. Zhang, "A new computational method of the identification jacobian for robot manipulator calibration", International Conference on Control, Automation and Systems Engineering, pp.1-3, 2011

[42] J. C. Lagarias, J. A. Reeds, M. H. Wright and P. Wright, "Convergence properties of the Nelder-Mead simplex algorithm in low dimensions" SIAM Journal of Optimization, Vol. 9, pp.112-147, 1998

[43] F. Y. Chen and V. L. Chan, "Dimensional synthesis of mechanisms for function generation using Marquardt's compromise", ASME J. Eng. Ind., Vol. 96, No. 1, pp.1312-1317, 1974

[44] H.G. Tull and D.W. Lewis, "Three dimensional kinematic synthesis", ASME J. Eng. Ind., Vol. 90, No. 3, pp.481-484, 1968

[45] Leica, "LTD500 laser tracker system: Specifications", Leica Geosystems Switzerland, 1999

[46] FANUC, "FANUC Robot R-2000*i*B mechanical unit: Parts manual", B-82236EN/01, 2007

[47] I. C. Ha, "Kinematic parameter calibration method for industrial robot manipulator using relative position", Journal of Mechanical Science and Technology, Vol. 22, pp.1084-1090, 2008

[48] R. Kang et al., "Learning the forward kinematics behavior of a hybrid robot employing artificial neural networks", Journal of Cambridge University Press: Robotica, pp.1-9, 2011

[49] B. Rooks, "Vision systems feature at TEAM", Emerald Sensor Review, Vol. 23, No. 2, pp.123-127, 2003

[50] B. Yurttagül, “Kinematic calibration of industrial robots using full pose measurements and optimal pose selection” METU Thesis Library, 2010

[51] JAVA, “Java BluePrints: Model-View-Controller”, Sun Microsystems, 2003

[52] R. Singh and H. S. Sarjoughian, “Software architecture for object-oriented simulation and simulation environments: Case study and approach”, Arizona State University Computer Science and Engineering Report, 2003

[53] M. J. D. Hayes and P. L. O’Leary, “Kinematic calibration procedure for serial robot with six axes”, Institute for Automation Montauniversitaet Leoben, 2001

[54] D. S. Wang, X. G. Liu and X. H. Xu, “Calibration of the arc-welding robot by neural network”, Proceedings of the IEEE Fourth International Conference on Machine Learning and Cybernetics, pp.4064-4069, 2005

APPENDIX A

REFERENCE FRAME ASSIGNMENT PROCEDURE

Before starting the experiments, base frame origin is registered as the reference. To do this, three reference holes are used to locate the origin; by measuring real coordinates of the centerline of holes, a transformation is applied to the measurement system's reference. Locations of the measured points are shown in the Figure A.1.

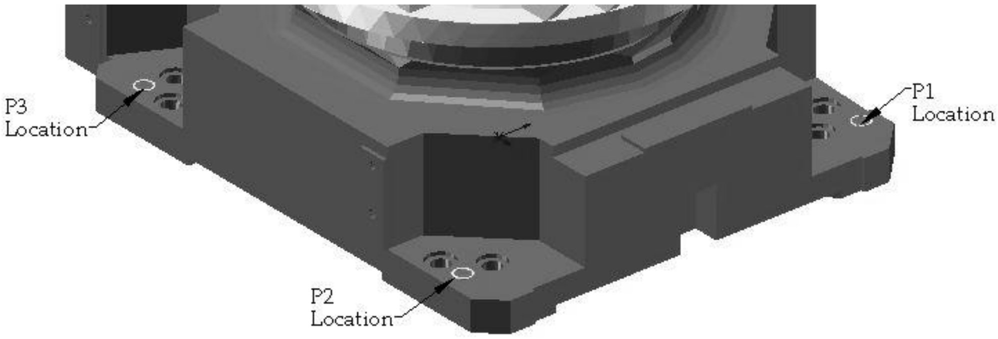


Figure A.1: Location of the Reference Points P1, P2 and P3
(CAD Model Courtesy of FANUC Robotics Corp.)

Using relative distances from the robot's zero point (see Figure A.2), theoretical zero is found. Real location of the robot's zero point, is found as the intersection of the three points on the ground. Subsequently, transformation from measured location to theoretical location carries the reference point of the measurements from measurement system to robot's zero point.

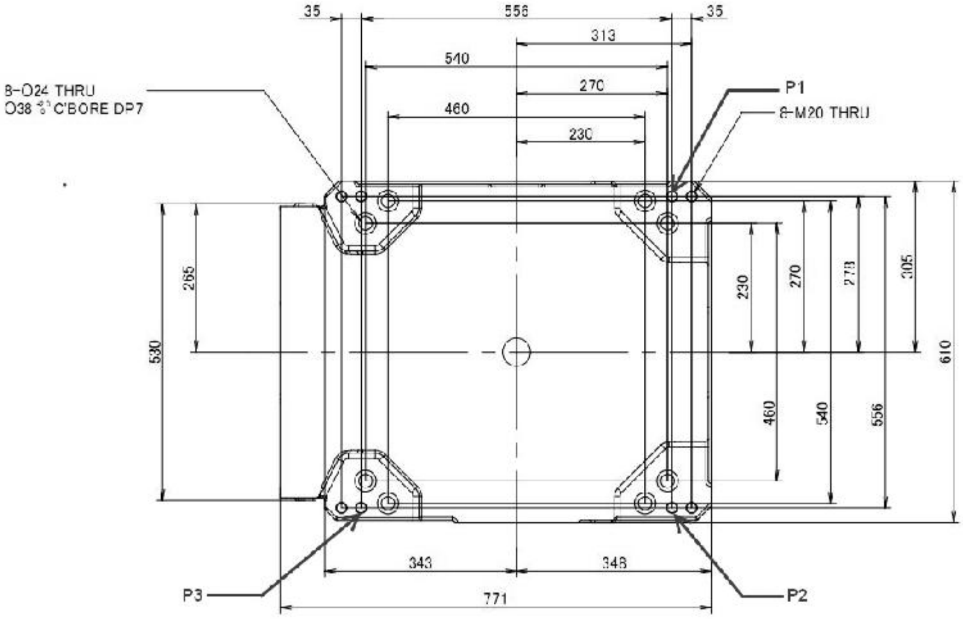


Figure A.2: Reference Points P1, P2 and P3 in CAD Drawings [46]

APPENDIX B

INVERSE KINEMATICS

Using advantageous property from of the spherical wrist, position of the end-effector is determined by first three joints.

$$ {}_0T^4 = {}_0T^1\,{}_1T^2\,{}_2T^3\,{}_3T^4 = \begin{bmatrix} {}_0R^4 & {}_0P^4_{ORIGIN} \\ 0\quad 0\quad 0 & 1 \end{bmatrix} \tag{B.1} $$

$$ {}_0P^4_{ORIGIN} = {}_0T^1\,{}_1T^2\,{}_2T^3\,{}_3T^4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{B.2} $$

Equation B.1 and Equation B.2 give the position of origin of the 4th frame. Multiplying Equation B.2 with ${}_1T^0$ (":" shows the terms are not concerned);

$$ \begin{bmatrix} P_x \cos\bar{\ }_1 + P_y \sin\bar{\ }_1 \\ P_y \cos\bar{\ }_1 - P_y \sin\bar{\ }_1 \\ P_z - d_1 \\ 1 \end{bmatrix} = \begin{bmatrix} : \\ 0 \\ : \\ 1 \end{bmatrix} \tag{B.3} $$

By using $P_y \cos\bar{\ }_1 - P_y \sin\bar{\ }_1 = 0$ from second row of the equation

$$ \begin{aligned} \theta_1 &= atan2(-P_y\,, -P_x\,) \\ \theta_1^{'} &= \theta_1 + \pi \end{aligned} \tag{B.4} $$

$\bar{\ }_1$ is obtained from Equation B.4.

Multiplying Equation B.2 with ${}_2T^0$ (":" shows the terms are not concerned);

$$\begin{bmatrix} cos\theta_2(-a_2+(P_x cos\theta_1)+(P_y sin\theta_1))+sin\theta_2(d_1-P_z) \\ cos\theta_2(d_1-P_z)-sin\theta_2(-a_2+(P_x cos\theta_1)+(P_y sin\theta_1)) \\ : \\ 1 \end{bmatrix} = \begin{bmatrix} a_3+a_4cos\theta_3+d_4sin\theta_3 \\ d_4cos\theta_3-a_4sin\theta_3 \\ : \\ 1 \end{bmatrix} \quad (B.5)$$

Using Equation B.5, two equations are extracted, shown below;

$$\begin{aligned} &cos\theta_2(-a_2+(P_x cos\theta_1)+(P_y sin\theta_1))+sin\theta_2(d_1-P_z)=a_3+a_4cos\theta_3+d_4sin\theta_3 \\ &cos\theta_2(d_1-P_z)-sin\theta_2(-a_2+(P_x cos\theta_1)+(P_y sin\theta_1))=d_4cos\theta_3-a_4sin\theta_3 \end{aligned} \quad (B.6)$$

and following terms are assigned;

$$\begin{aligned} W_1 &= d_4 \\ W_2 &= -a_4 \\ X &= (d_1-P_z) \\ Y &= (a_2-(P_x cos\theta_1)+(P_y sin\theta_1)) \\ Z_1 &= 0 \\ Z_2 &= -a_3 \end{aligned} \quad (B.7)$$

Consequently; B_1, B_2 and B_3 are calculated;

$$\begin{aligned} B_1 &= 2(Z_1Y+Z_2X) \\ B_2 &= 2(Z_1X-Z_2Y) \\ B_3 &= W_1^2+W_2^2-X^2-Y^2-Z_1^2-Z_2^2 \end{aligned} \quad (B.8)$$

$B_1sin\theta_2+B_2cos\theta_2=B_3$ provides $sin\theta_2$ and $cos\theta_2$, and finally θ_2 is found.

$$\theta_2 = atan2(sin\theta_2, cos\theta_2) \quad (B.9)$$

After finding θ_2, Equation B.6 simplifies into;

$$\begin{aligned} Z_1 &= Y_1 cos\theta_3 - X_1 sin\theta_3 \\ Z_2 &= X_2 sin\theta_3 + Y_2 cos\theta_3 \end{aligned} \quad (B.10)$$

In these equations variables are described as;

$$
\begin{aligned}
&X_1 = -Y_2 = -a_4 \\
&X_2 = Y_1 = d_4 \\
&Z_1 = (d_1 - P_z)cos\theta_2 + (a_2 - (P_x cos\theta_1) + (P_y sin\theta_1))sin\theta_2 \\
&Z_2 = (d_1 - P_z)sin\theta_2 - (a_2 - (P_x cos\theta_1) + (P_y sin\theta_1))cos\theta_2 - a_3
\end{aligned}
\tag{B.11}
$$

Subsequently by using Equation B.10, ¯$_3$ is found as,

$$sin\theta_3 = \frac{Z_1 Y_2 - Z_2 Y_1}{X_1 Y_2 - X_2 Y_1} \tag{B.12}$$

$$cos\theta_3 = \frac{Z_2 X_1 - Z_1 X_2}{X_1 Y_2 - X_2 Y_1} \tag{B.13}$$

$$\theta_3 = atan2(sin\theta_3, cos\theta_3) \tag{B.14}$$

In manipulations so far, $\left[q_1, q_2, q_3\right]^T$ are found. From now on, solution is extended to find $\left[q_4, q_5, q_6\right]^T$ by using rotation matrix of the end-effector. Since ¯$_1$, ¯$_2$ and ¯$_3$ is known ${}_0R^3$ is also known. In multiplication of ${}_6R^3\,{}_3R^4 = {}_6R^4$, ${}_6R^3$ defined as;

$${}_6R^3 = \begin{bmatrix} F_x & G_x & H_x \\ F_y & G_y & H_y \\ F_z & G_z & H_z \end{bmatrix} \tag{B.15}$$

Subsequently,

$$\begin{bmatrix} F_x cos\theta_4 - F_z sin\theta_4 & G_x cos\theta_4 - G_z sin\theta_4 & H_x cos\theta_4 - H_z sin\theta_4 \\ -F_z cos\theta_4 - F_x sin\theta_4 & -G_z cos\theta_4 - G_x sin\theta_4 & H_z cos\theta_4 - H_x sin\theta_4 \\ F_y & G_y & H_y \end{bmatrix} \tag{B.16}$$

$$= \begin{bmatrix} : & : & sin\theta_5 \\ sin\theta_6 & cos\theta_6 & 0 \\ : & : & cos\theta_5 \end{bmatrix}$$

From Equation B.16 by selecting $H_z cos\theta_4 - H_x sin\theta_4 = 0$;

$$\begin{aligned} \theta_4 &= atan2(H_z, -H_x) \\ \theta_4^{'} &= \theta_4 + \pi \end{aligned} \tag{B.17}$$

¯$_4$ is found. Alternatively from Equation B.16, following equations could be selected for ¯$_5$;

$$\begin{aligned} sin\theta_5 &= H_x cos\theta_4 - H_z sin\theta_4 \\ cos\theta_5 &= H_y \end{aligned} \tag{B.18}$$

$$\theta_5 = atan2(sin\theta_5, cos\theta_5) \tag{B.19}$$

And finally for ¯$_6$;

$$\begin{aligned} sin\theta_6 &= -cos\theta_4 F_z - sin\theta_4 F_x \\ cos\theta_6 &= -cos\theta_4 G_z - sin\theta_4 G_x \end{aligned} \tag{B.20}$$

$$\theta_6 = atan2(sin\theta_6, cos\theta_6) \tag{B.21}$$

To sum up, ¯$_1$, ¯$_2$, ¯$_3$, ¯$_4$, ¯$_5$ and ¯$_6$ found analytically since proposed approach may need inverse model for implementation stage when the robot controller do not allow parameter updating. When this will be the case, in order to update the pose components for fake targets, inverse model will be used with resultant identified parameters. For that reason, inverse model is coded into developed simulation software in the form of symbolic equations rather numeric values. By changing the parameters related with the robot parts, simulation software calculates the desired joint angles for articulated configurations.