Doctor of Philosophy

DEVELOPMENT OF ROBOT MANIPULATOR CALIBRATION TECHNIQUES USING MODEL BASED IDENTIFICATION AND UNMODELED COMPENSATION METHODS

Graduate School of the University of Ulsan
Department of Electrical Engineering
PHU-NGUYEN LE

공학박사 학위논문

모델 및 비모델 보상을 이용한 로봇 매니퓰레이터 캘리브레이션 방법 개발

DEVELOPMENT OF ROBOT MANIPULATOR CALIBRATION TECHNIQUES USING MODEL BASED IDENTIFICATION AND UNMODELED COMPENSATION METHODS

지도교수 강희준

이 논문을 공학박사 학위논문으로 제출함
2020년 11월

울산대학교 대학원
전기전자보시스템공학부

Phu-Nguyen Le

Phu-Nguyen Le 의 공학박사 학위논문을 인준함

심사위원장	**조강현**	(인)
심사위원	**김한실**	(인)
심사위원	**서영수**	(인)
심사위원	**강현덕**	(인)
심사위원	**강희준**	(인)

울산대학교 대학원

2020 년 11 월

DEVELOPMENT OF ROBOT MANIPULATOR CALIBRATION TECHNIQUES USING MODEL BASED IDENTIFICATION AND UNMODELED COMPENSATION METHODS

Supervisor: Professor Hee-Jung Kang

A Dissertation

Summited to

The Graduate School of the University of Ulsan
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

by

Phu-Nguyen Le

Department of Electrical Engineering
University of Ulsan, Korea
November 2020

DEVELOPMENT OF ROBOT MANIPULATOR CALIBRATION TECHNIQUES USING MODEL BASED IDENTIFICATION AND UNMODELED COMPENSATION METHODS

This certifies that the dissertation of
Phu-Nguyen Le is approved

Professor Kang-Hyun Jo
Committee Chair

Professor Young-Soo Suh
Committee Member

Professor Han-Sil Kim
Committee Member

Professor Hyun-Duk Kang
Committee Member

Professor Hee-Jung Kang
Committee Member

Department of Electrical Engineering
University of Ulsan, Korea
November 2020

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Professor Kang-Hee Jun. There are no words to show my appreciation and respect for him. Surely this thesis would never have been fruition without the help, valuable guidance, strong belief, constant support, continuous encouragement, technical instructions, and great patience of him through my research work.

Many thanks are also due to Professor Han-Sil Kim, whose works demonstrated to me the guidance in the areas of probability and linear systems. I sincerely thank Professor Kang-Hyun Jo for his valuable lectures about artificial intelligence. I am very thankful to Professor Yong-Soo Suh for his instructions in the estimation and control theory. I would like to show my gratitude for other faculty professors and members of the Electrical Engineering Department, Computer Engineering Department, and Mechanical Engineering Department at the University of Ulsan for their help and what they dedicated to students like me.

I would also be thankful to my senior lab members Dr. Hoai-Nhan Nguyen, and Dr. Minh-Duc Tran. I learned a lot from them since I was a novice in the robotic calibration research field. My thanks also go to Professor Tien-Dung Le, Professor Mien-Van, Dr. Xuan-Toa Tran, and Dr. Ngoc-Bach Hoang, for their help, kindness, and enthusiasm since I first set my foot in Korea.

I wish to thank my laboratory member Mr. Duy-Tang Hoang, who introduced me to the fascinating world of heuristic optimization algorithms and helped me in writing the MATLAB code for my papers. I am grateful to Mr. Anh-Tuan Vo for his comments, suggestions, and qualified time over my writing. I want to thank Mr. Van-Cuong Nguyen for his encouragement and valuable technical advices. I am also beholden to Mr. Quang-Dan Le and Mr. Thanh-Nguyen Truong for their supports to me through my research work.

I could not forget to thank my close friends, Mr. Quang-Huy Nguyen, Mr. Xuan-Diep Ho-Van, Mr. Cong-Nguyen Hua, Ba-Vinh Nguyen, for their caring and sympathy for me and my *family over a hard time. And without Dr. Thanh-Dam Pham, Dr. Anh-Dung Vo, Dr. Duy-Hung Dam, Mr. Duc-Khanh Hoang, Mr. Trang-Tien Nguyen, and Mr. Huu-Cong Vu, my days in Ulsan were not that full of joyfulness and happiness.*

I cannot end without demonstrating my gratefulness to my family. Thank you, Mrs. Tuyet-Hang Nguyen Thi, for her psychological support, shouldering burdens, teaching me the meaning of compassion, sharing my triumphs and my tears. You are the best mom all over the world. I would like to show my best grateful to my sister, Ms. Thu-Giang Le, for her watching and caring out for me when things do not seem fine. My many thanks also to Mr. Gia-Thuan Le for taking care of me throughout many years. Having a brother like you makes me the happiest. My grateful is also to my youngest brother, Mr. Viet-Khue Le Ho for everything he does for me.

I wish to avail myself for this opportunity to express my gratitude and love to all of my relatives, and my friends, especially my grandmother Mrs. Thi-Tu Nguyen, my grandfather Mr. Van-Huong Nguyen, my grandmother Thi-Bo Nguyễn, Mr. Van-Khai Lê, Mrs. Thi-Hoa Nguyen ,Mrs. Thi-Khan Nguyen, Mrs. Thi-Chien Nguyen, Mr. Van-Dinh Le and Mrs. Thi-Hue Pham, Ms. Tuyet-Nga Nguyen Thi, Mr. Duy-Thinh Le and Mrs. Thuy-Uyen Vu , Mr. Duy-Thinh Le and Mrs. Thi-Hanh Tran, Mr. Duc-Vinh Nguyen, Ms. Xuan-Dieu Ho Thi, Mr. Phu-Quy Nguyen, Mr. The-Vi Pham, Mr. The-Cuong Pham, my parents in law Mr. Van-Hung Nguyen and Mrs. Xuan-Hoa Le Thi, Mr Van-Can Pham, Mr Viet-Nhan Le and Dr. Kim-Nga Truong for the assistance and encouraged they have provided me all the way to get the Ph.D.

Last but not least, I would like to express my gratitude, appreciation, and respect for my father, Mr. Duy-Trinh Le. Thank you for no reason and thank you for every reason. I do not know how to express my love for you. I am just speechless by remembering your contributions to my life. I also want to thank my grandfather Mr. Van-Buu Le. He is the first teacher *of my life, who patiently taught me reading and writing. I love both of you a lot. You will be missed, and you will never be forgotten. May your souls rest in peace.*

To my dear wife, Thi-Huong Nguyen, I am so blessed of having you who always stand by me through ups and downs. Thank you for your love, care, support, and every little effort you are doing for me is simply amazing. Loving you is the sweetest thing to do. Because of you, my life is full of happiness. I love you.

Phu-Nguyen Le

December 2020, University of Ulsan, Korea

To my grandfather : Văn-Bửu Lê

My father : Duy-Trinh Lê

Abstract

In recent years, interest in the application of robot manipulator to automated manufacturing soared. The advent of highly capable computer-controlled manipulators indicate d that truly flexible automation was feasible, and many manufacturers rushed to take advantage of this technology. Robots are used in a wide range of tasks in industrial applications such as material handling, milling, painting welding, and roughing. Although the modeled-based robotic calibration methods have been widely researched for decades. It is difficult to create models that consider all the causes engendering the end effecter error. Therefore, to archive further accuracy, a good deal of attention has been paid to the area of un-modeled calibration for the sources of errors that could not be taken into account by model-based calibration.

In this study, new robotic calibration methods are introduced. By combining the joint deflection model with the conventional kinematic model of a manipulator, the geometric errors and joint deflection errors can be considered together to increase its positional accuracy. A new method includes the kinematic calibration and non-geometric compensation with a RBF compensator that compensates for compliance errors based on the effective torques. To improve the effectiveness of the calibration process, a neural network is designed to additionally compensate the unmodeled errors, specially, non-geometric errors. Then, the weights and biases of the neural network is determined by conventional back-propagation method. For increasing the ability of the neural network, heuristic optimization methods such as teaching learning optimization and invading weed optimization methods are hired for better convergence capability than the back propagation neural network in this calibration process.

This work also presents a new method includes the kinematic calibration and teaching learning-based optimization for directly determining joint compliance parameters. The advantages of the suggested method are easy for implementing, removing the need for torque sensors, high ability to enhance the precision of the manipulator.

In order to demonstrate the effectiveness of the proposed method, experimental studies are carried out on manipulators. The enhanced position accuracy of the manipulator after the calibration confirms the feasibility and more positional accuracy over the other calibration methods.

Keywords: Industrial robot manipulator, Constant joint stiffness, Non-linear joint stiffness, Teaching-learning optimization, Invasive weed optimization, Neural network, Radial basis function.

Contents

List of figures

List of tables

Chapter 1: INTRODUCTION

In recent years, interest in the application of robot manipulator to automated manufacturing soared. The advent of highly capable computer-controlled manipulators indicate d that truly flexible automation was feasible, and many manufacturers rushed to take advantage of this technology. Robots are used in a wide range of tasks in industrial applications such as material handling, milling, painting welding, and roughing. Although the use of robot manipulators in flexible manufacturing systems still presents problems, such as the high initial capital costs of automation along with other economic and technical problems caused many managers to avoid the use of robots, the goal remains a highly desirable one. However, experience has shown that industrial manipulators have much repeatability than accuracy. Therefore, the utility of robot manipulators would be significantly enhanced if they were made to be as accurate as they are repeatable. To attempt to solve this problem, a good deal of attention has been paid to the area of manipulator calibration.

1.1 Generally

An industrial robot is comprised of a robot manipulator, power supply, and comptrollers. From a mechanical viewpoint, designers describe the robot by means of parameters representing geometrical (link lengths, link orientations) and mechanical characteristics (link mass, link stiffness) . The source and significance of manipulator errors are topics that have attracted the attention of numerous researchers[1]. Whitney et al. [1] used a PUMA 560 robot to determine that the most significant error sources for the robot were several nongeometric errors. Judd and Kasinski [2] studied an AID-900 robots and found that the geometric errors were responsible for approximately 95% of the measured error and the nongeometric errors appeared to be almost negligible. Jean et al. [3]and Becque [4] reported that flexibility in joints and in links is responsible for 8- 10% of the total position and orientation errors. Link flexibility is than joint flexibility error-blow 5% [4]. Therefore, more recently researchers devoted to identification of compliance errors as well as kinematic errors in industrial robots. Caenen and Angue improved the robot accuracy through identifying both kinematic parameters and angular deflections in x, y and z axes [5]. Similarly, Jang et al.[6] considered the joint angular errors as the sum of geometric error and joint deformation error. Gong et al.[7] compensate joint deflections in z axis by identifying defined dimensionless parameters as well as kinematic errors. Despite the fact that the error due to temperature is responsible for only 0.1% of the total error [3], they also established and applied empirical thermal error models to estimate thermal errors by monitoring the temperature field at different operation conditions [7].

To enhance the accuracy of the robot manipulator , the reason for the difference between accuracy and repeatability should be understood. Repeatability is defined as the ability of the manipulator to return to a pose that has been stored in joint space. Accuracy, however, is the ability of the robot to move to a pose defined in task space. To achieve a pose that is defined in task space, the robot controller must convert the task space definition of the pose into joint space. The individual joints are then moved so that the desired configuration is obtained. The conversion from task space to joint space is accomplished by using a mathematical model of the manipulator. This mathematical model relates the joint displacements to the end effector pose and vice versa. When the mathematical model used by the robot controller to describe the robot motion differs from the actual geometry of the manipulator, the joint space definition of a pose defined in task space will not be accurate.

Since deviations between the mathematical model used in the controller and the actual arm geometry are a source of inaccuracy , it is clear that there are two basic ways of enhancing accuracy. The first would be to build every robot so that all of the various parameters match the “design” or “nominal” values as closely as possible. In other words, the manufacturing tolerances on every part would be extremely tight. Clearly, this approach is not feasible because of the excessive costs that would be involved. If the robot cannot be made to match the model, then the second alternative is to make the mathematical model match the robot. This is the essence of manipulator calibration. Simply put, manipulator calibration is the process of defining an appropriate mathematical model and then determining the various model parameters that make the model match the robot as closely as possible. Bernhardt and Albright defined robot calibration as ” the procedure used in determining the actual parameters which described the geometrical dimensions and mechanical characteristic of manufactured robot structure”[8].

Since the significant source of error can vary from one robot design to the next, calibration procedures can vary widely in their scope and complexity. For example, some robot calibration procedures consider only the joint transducer information while others may involve changes in the kinematic or dynamic model of the robot. To classify most of the current approaches to robot calibration, robot calibration could be classified into three levels.

Level 1 calibration shall be defined as “joint level” calibration. The goal is to determine the correct relationship between the signal produced by the joint displacement transducer and the actual joint displacement. This usually involves calibration of the kinematics of the drive and the joint sensor mechanisms.

Level 2 calibration is defined as the entire robot kinematic model calibration. At this level the purpose of the calibration is to determine the basic kinematic geometry of the robot manipulator as well as the correct joint angle relationship.

Level 3 calibration is defined as “non-geometric” calibration. Nonkinematic errors in positioning of the end effector of a robot are due to effects such as joint compliance, friction, and clearance, as well as link compliance, backlash, thermal variation.

The process of robot calibration falls conveniently into four sequential operations[9][10]:

- Modeling: This is the first step in the calibration process is the determination of a suitable functional relationship between robot parameters and the resulting transformation (position and orientation) of robot end-effector with respect to a reference coordinate.
- Measurement: The second step in the calibration process is measurement. The goal of the measurement process is to accurately determine either the end effector pose, or some subset of the pose, for a set of robot joint displacements. The system which is used in measurement process need to be highly precision.
- Identification: The identification process is the mathematical process of using model developed and the collected data to identify the desired parameters.
- Implementation: The implementation means using calibration information to improve manipulator performance. So far, the calibration process should have given an accurate model, with known parameters, that accurate relationship between the joint variables and tool pose. Since the objective of the previous three phases, modeling, measurement, and identification, has been determine the best model of the manipulator, conceptually the implementation phase simply involves the modification of the nominal model embedded in the robot controller. Clearly any manipulator controller that is able to accept an

externally specified pose and convert it to a set of joint variables is performing an inverse solution on some model. However, the detailed architecture of a particular robot controller is not usually available to the user, making it difficult to implement calibration data in this way. In situations in which a controller does not contain a model of the manipulator, as in case in which tool to joint transforms are not provided and the arm is only able to be "taught" pose, implementation in the exiting controller cannot be archived. Solving the problems is usually accomplished by implementing the calibration data in an off-line preprocessor. Since virtually all manipulators are capable of being driven by the specification of a set of joint angles, this preprocessor performs the conversion of each world specified pose into the corresponding joint set through an inverse kinematic solution of the accurate manipulator model. The sets of joint variables are then sent to the existing joint con troller and executed by the manipulator.

However, a nominal geometrically model according to a product specification sheet does not include the errors arising in manufacturing or assembly. In addition, it also does not include non-geometric errors, such as gear transmission errors, gear backlashes, arm compliance, etc., which are difficult to geometrically consider in a kinematic model. Therefore, some methods of calibrating precisely kinematic and non-kinematic parameters are required to improve the accuracy of robot manipulators. For that reason, this dissertation is intended to address issues related to the level 3 calibration, specially to develop nongeometric compensator for robot calibration. The decision to limit the discussion of these topic was made primally because that many significant issues involving level 1 and 2 calibration procedures have been resolved and that implementation of these techniques is feasible. Level 3 calibration, on the other hands, is still very much a research issue.

In this work, techniques for the calibration of industrial robots are proposed by combining the advantages of the model-based calibration and artificial neural network compensation methods. The optimization methods are also employed to optimize weights and bias of the neural network. In order to demonstrate the effectiveness of the proposed methods, experimental studies are carried out on manipulators. The enhanced position accuracy of the manipulator after the calibration confirms the feasibility and more positional accuracy over the other calibration methods.

1.2 Organized of the Thesis

The following of this thesis is organized as follows

Chapter 2 presents model-based robot calibrations. This chapter is served as basic for the rest of this dissertation. In this part. The kinematic calibration algorithm is introduced first. Then, the simultaneous identification of constant joint compliance and kinematic parameters calibration method is shown. At the end of this section, the calibration results of experiment study using the two methods are presented.

In chapter 3, a new robotic calibration method is proposed for reducing the positional errors of the robot manipulator. First, geometric errors of a robot are identified by using a conventional kinematic calibration model of the robot. Then, a radial basis function is constructed for compensating the compliance errors based on the effective torques for further increasing the positional precision of the robot. The enhanced positional accuracy of the robot manipulator in experimental studies that are carried on a YS100 robot illustrates the advantages of the suggested algorithm than the other techniques.

To archive better precision result, Chapter 4 proposes a new method for enhancing robot position accuracy. In order to increase robot accuracy, the proposed method models and identifies determinable error sources, for in-stance, geometric errors, and joint deflection errors. Because non-geometric error sources such as link compliance, gear backlash, and others are difficult to model correctly and completely, an artificial neural network (ANN) is used for compensating for the robot position errors, which are caused by these non-geometric error sources. The proposed method is used for experimental calibration of an industrial

Hyundai HH800 robot designed for carrying heavy loads. The robot position accuracy after calibration demonstrates the effectiveness and correctness of the method.

Chapter 5 demonstrates a new calibration method for enhancing robot positional accuracy of the industrial manipulators. By combining the joint deflection model with the conventional kinematic model of a manipulator, the geometric errors and joint deflection errors can be considered together to increase its positional accuracy. Then, a neural network is designed to additionally compensate the unmodeled errors, specially, non-geometric errors. The invasive weed optimization method is employed to optimize weights and bias of the neural network. In order to demonstrate the effectiveness of the proposed method, real experimental studies are carried out on HH 800 manipulator. The enhanced position accuracy of the manipulator after the calibration confirms the feasibility and more positional accuracy over the other calibration methods.

Following the chapter 5, chapter 6 proposed a robotic calibration algorithm for improving robot manipulator position precision. At first, the kinematic parameters as well as the compliance parameters of the robot can be identified together to improve its accuracy using the joint deflection model and the conventional kinematic model calibration technique. Then, an artificial neural network is constructed for further compensating the unmodeled errors. The teaching learning-based optimization is used to determine the parameters of the neural network with less computing and better convergence without relying on choosing initial parameters. To show the advantages of the suggested technique, a HH800 robot is employed for the experimental study of the proposed algorithm. The improved position precision of the robot after the experiment firmly proves the practicability and positional precision of the proposed method over the other comparing algorithms.

In chapter 7, a new calibration method for the industrial robot manipulator is proposed, which directly identifies joint compliance parameters by combining kinematic calibration technique and teaching-learning-based optimization method. The proposed method combined the geometric and non-geometric calibration to consider the geometric errors and the shifting of joints together to enhance the positional precision of the industrial robotic manipulator. Due to the fusion of the two fundamental methods, the proposed calibration technique has many advantages such as fast convergence, less computing, accurate knowledge of error sources and significantly increase the accuracy of the robot. The proposed modeled-based calibration technique is also performed on YS100 robot to demonstrate its effectiveness and feasibility.

Finally, the conclusion of this dissertation and suggestions for further development are given in chapter 8.

Chapter 2: MODEL BASED ROBOT CALIBRATIONS

2.1 Introduction

In the past several decades, robot manipulator has played an important role in the industry to carry many works such as welding, spraying paint, carrying load, etc. One of the significant technical problem to addressed is the inability of most robots to be programmed offline or to share program with other robots. This happened due to the nominal geometrically does not contain the errors arising in producing and assembly. Therefore, the robot manipulator has been characterized by high repeatability and low accuracy. To solve this problem, there are huge amount of studies concentrated to the area of manipulator calibration. Generally, the robotic calibration is classified into two main categories: the geometric calibration and non-geometric calibration [11–30].

For decades, one of the most popular ideas in geometric calibration literature is to create a properly error model to suitably describe geometric errors of the robot manipulator based on kinematic parameters. Numerous methods have been suggested to inspect an appropriate model for robotic calibration [11–17, 23–30]. The D-H model is one of the most fundamental famous calibration methods suggested by Denavit- Hartenberg [11, 12, 23–25]. The method considers a link of robot only as a rigid body which defines the relationship between two neighboring joint axes of a manipulator. This technique is still extensively adopted by many studies with some minute modification [26–28]. Besides, there are some population calibration methods. The complete and parametrically continuous (CPC) modeling technique is one of the more versatile alternative modeling conventions designated to fit the needs of manipulator calibration [29, 30]. The zero-reference position method is based upon an arbitrary choice of the zero-reference position, the principle of displacement similarity, and the hand-to-the base sequence of changing the joint variables in a serial manipulator [13–15]. The product of exponentials (POE) formulation approach is also widely investigated. This method is relied on recursive forward dyad kinematics [16–19]. There have been some attempts to use the Kalman filter , particle filter, etc. to estimate the kinematic parameters [31–33]. But the effectiveness of those attempt does not seem to be much.

The geometric calibration methods are fast, easy to convert. However, these methods are limited by studying only the geometric error. The industrial manipulator errors may also come from many non-geometric such as joint and link compliance, temperature variation, gear transmission, etc. Among these non-geometric errors , the error is caused by the link and joint elastic is dominant.

In order to address the nonlinear errors and further raising the robotic positional precision, some studies make effort to additionally model the nonlinear errors [6, 20–22, 34–38]. The flexibilities in links and joints are took into account by compliance model [20–22]. Some studies use the optimization criterion to select optimal poses which ensure the best nongeometric errors compensator[35–37]. Other researchers adopt the artificial neural network (ANN) to estimate the compliance parameters[6, 38]. There are also some works using the ANN to compensate the un-modeled errors by using the joints configurations[31, 39]. Despite the fact that these methods are effective in increasing the positional accuracy of the robot manipulator, these ANN-based methods take the computational effort, time and does not quarantine a convergence. Furthermore, most of ANN-based technique is applied after the kinematic calibration.

In this chapter, the kinematic calibration algorithm[23] is introduced first. Then, the simultaneous identification of constant joint compliance and kinematic parameters calibration method[22] is shown. Finally, the two other calibration methods are implemented for the simulated and experimental studies on a YS100 robot. At the end of this section, the calibration results of experiment study using the two methods and some conclusions are presented.

2.2 Kinematic calibration

Assuming that the robot's end-effector position P_{real} is calculated by the following equation:

$$P_{real} = P_{kin} + \Delta P_{kin} + \Delta P_c + \Delta P_{extra} \tag{2.1}$$

where P_{kin} is the position of the end effector calculated by the kinematic parameters, ΔP_{kin} is the position error caused by the geometric error, ΔP_c is the position error due to the joint compliance, andΔP_{extra} is the positional residual error that is not modeled. Assuming that geometric errors and joint deflection errors are the main parts in causing the position errors ΔP_{kin} ($\Delta P_c + \Delta P_{extra} = 0$). The error model can be expressed as:

$$\Delta P = P_{real} - P_{kin} = \Delta P_{kin} \tag{2.2}$$

The position errors caused by geometric errors ΔP_{kin} in the Eq. 2.1 could be identified by the conventional kinematic calibration[23][24][25]. ΔP_{kin}can be expressed as

$$\Delta P_{kin} = J_{kin} \Delta \phi \tag{2.3}$$

where J_{kin} ($3 \times n$) is a kinematic Jacobian matrix[10][22]. $\Delta \phi$ is a n×1 kinematic parameter error vector. n is the number of the calibrated kinematic parameters. The total number of kinematic parameters is equal to 32. However, the 6 DOF revolute robot has several dependencies between some parameters. These dependency parameters are $\{\Delta\theta_1, \Delta\theta_0\}$, $\{\Delta d_1, \Delta d_0\}$, $\{\Delta d_2, \Delta d_3\}$, $\{\Delta z_T, \Delta d_6\}$, $\{(\Delta x_T, \Delta y_T), \Delta\theta_6\}$ In each pair, the parameter errors cannot be identified together. Therefore, the dependency parameters that are chosen to calibrate are $\{\Delta\theta_1, \Delta d_1, \Delta d_3, \Delta\theta_0, \Delta x_T, \Delta y_T, \Delta z_T\}$ while the other error parameter in each pair is set to the nominal parameter value. So, the number of calibrated kinematic is reduced to 27.

The Eq. 2.3 can be solved by the least-square method to overcome the effect of noise and uncertainty:

$$\Delta \phi = [(J^T J)^{-1} J^T] \Delta P \tag{2.4}$$

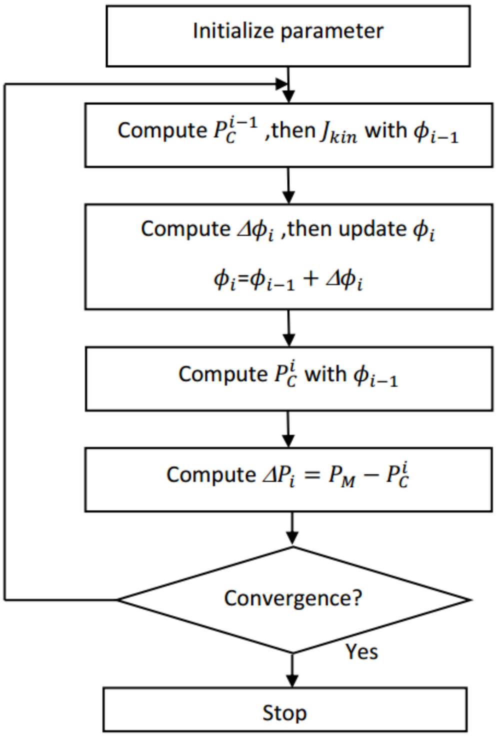

Initialize parameter
Compute P_C^{i-1} ,then J_{kin} with ϕ_{i-1}
Compute $\Delta\phi_i$,then update ϕ_i
ϕ_i=$\phi_{i-1}+\Delta\phi_i$
Compute P_C^i with ϕ_{i-1}
Compute $\Delta P_i = P_M - P_C^i$
Convergence?
Yes
Stop

Fig. 2.1 Flowchart of the conventional kinematic calibration method

2.3 Simultaneous identification of constant joint compliance and kinematic parameters of industrial robots

Accepting the manipulator joint can be demonstrated as a linear torsional spring, the joint stiffness value of the i_{th} joint is symbolized by a constant value k_i. Considering the manipulator joint is much bender than the relative link . Hence, the most elastic errors are result of the elasticity of manipulator joints under the effect of the links' weight themselves and external payload. The shifting of the i_{th} joint can be expressed by the effective torques [23]:

$$\Delta\theta_{c_i} = \frac{\tau_i}{k_i} = \tau_i c_i \tag{2.5}$$

The end effector position errors due to the elastic of the joints can be calculated as:

$$\Delta P_c = J_\theta \Delta\theta_c = (J_\theta \tau)C \tag{2.6}$$

where $C = [\, c_1 \quad c_2 \quad \ldots \quad c_n]^T$ is the joint compliance vector, $\Delta\theta_c = [\,\Delta\theta_{c_1} \quad \Delta\theta_{c_2} \quad \ldots\Delta\theta_{c_n}]^T$ is the joint deflection vector, and $\tau = diag(\tau_1, \tau_2, \ldots, \tau_n)$ is the effective torque in the robot joints at the balance position. J_θ the sub-matrices computed by the method presented in [34]. The robot's tip position vector can be formed:

$$P_{real} = P_{kin} + \Delta P_{kin} + \Delta P_c + \Delta P_{extra} \tag{2.7}$$

where P_{kin} is the result of forward kinematics based on the current kinematic parameters. ΔP_{kin}, ΔP_c, ΔP_{extra} are the position errors due to the kinematic parameter error, joint elastic, and the residual errors due to the unmodeled sources, respectively. The combined error model can be showed as:

$$\begin{aligned} \Delta X &= \Delta P_{kin} + \Delta P_c \\ &= J\varDelta\phi + (J_\theta \tau)C \\ &= [J \quad J_\theta \tau]\begin{bmatrix} \varDelta\phi \\ C \end{bmatrix} \\ &= J_\Phi \varDelta\Phi \end{aligned} \tag{2.8}$$

where ΔX is a (3 × 1) vector of three position errors of the robot end-effector. J is a (3 × p) matrix that relates the column vectors ΔX and ΔP_{kin}.(p = 27 is the total number of kinematic parameters). By using the least-square method, the kinematic and joint compliance parameter can be computed at the same time. This iterative process is conceptually explained in Fig. 2.2.

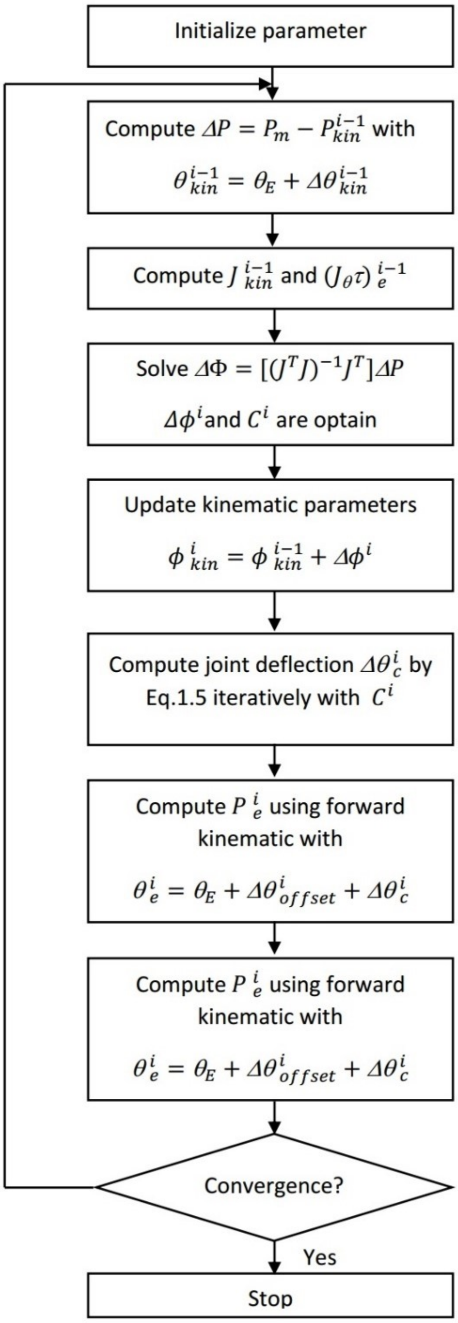

Initialize parameter
Compute $\Delta P = P_m - P_{kin}^{i-1}$ with
$\theta_{kin}^{i-1} = \theta_E + \Delta\theta_{kin}^{i-1}$
Compute J_{kin}^{i-1} and $(J_\theta\tau)_e^{i-1}$
Solve $\Delta\Phi = [(J^TJ)^{-1}J^T]\Delta P$
$\Delta\phi^i$and C^i are optain
Update kinematic parameters
$\phi_{kin}^{i} = \phi_{kin}^{i-1} + \Delta\phi^i$
Compute joint deflection $\Delta\theta_c^i$ by
Eq.1.5 iteratively with C^i
Compute P_e^i using forward
kinematic with
$\theta_e^i = \theta_E + \Delta\theta_{offset}^i + \Delta\theta_c^i$
Compute P_e^i using forward
kinematic with
$\theta_e^i = \theta_E + \Delta\theta_{offset}^i + \Delta\theta_c^i$
Convergence?
Yes
Stop

Fig. 1.2 Flowchart of the simultaneous identification of constant joint compliance and kinematic parameters method

2.4. Experiment Study and Results

2.4.1 Kinematic Structure of The YS100 Robot

YS100 is a 6 DOF serial robot[22]. The kinematic structure of it is briefly describe in the Fig.2.3 and Table. 2.1 . The homogenous transformation matrix between the base frame and the end effector of the robot is express as the following equation:

$$
{}_{E}^{0}T = {}_{1}^{0}T(\theta_1).\,{}_{2}^{1}T(\theta_2).\,{}_{3}^{2}T(\theta_{3p}).\,{}_{4}^{3}T(\theta_4).\,{}_{5}^{4}T(\theta_5).\,{}_{6}^{5}T(\theta_6).\,{}_{E}^{6}T \tag{2.9}
$$

The transformation matrix between the end effector frame and frame 6:

$$
{}_{E}^{6}T = Tr(x_6, a_6).Tr(y_6, b_6).Tr(z_7, d_7) \tag{2.10}
$$

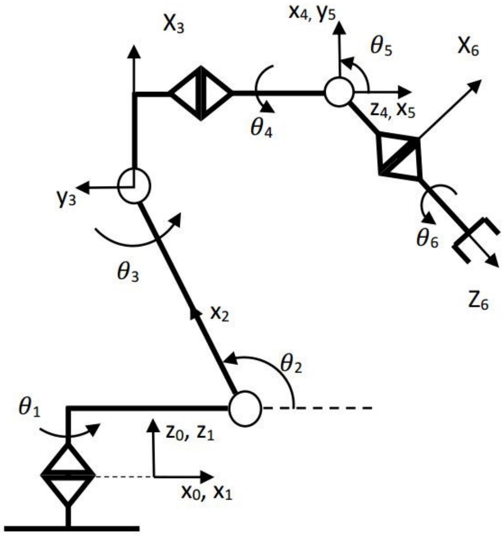


Fig. 2.3. Sketch of the YS100 robot.

Table 2.1 Nominal D-H Parameters of The Hyundai Robot YS100.

i	α_{i-1}(deg)	a_{i-1}(m)	β_{i-1}(deg)	b_{i-1}(m)	d_i(deg)	θ_i(deg)
1	0	0	0	0	0.48	θ_1
2	90	0.32	-	-	0	θ_2
3	0	0.87	0	-	0	θ_3
4	90	0.2	-	-	1.03	θ_4
5	-90	0	-	-	0	θ_5
6	90	0	-	-	0.185	θ_6
T	-	0.2	-	0.05	0.5	-

2.4.2 Experiment results of the calibration methods

2.4.2.1. Experimental calibration results

The calibration process is carried on a YS100 robot with 110 kg payload, and API laser tracker and an accompanying laser tracker . The API laser tracker has the accuracy of 0.01 mm/m, repeatability of ±0.006 mm/m. The reflector is fixed at a particular location of the robot end-effector. In order to acquire suitable measurement data for robot parameter identification, the robot moves its end-effector to positions that entirely cover the workspace. The 3D coordinates of the end effector are measured by the laser tracker and saved in a computer. At the same time, the associated robot joint readings also are recorded. These measurements will be randomly grouped in two sets . A set of 25 robot configurations (Q1) is used in parameter identification. By utilizing the conventional kinematic calibration method, the kinematic and joint compliance parameters are identified by Table 2.2

Table 2.2. D-H parameters of the Hyundai robot YS100 identified by KM. ("-": unavailable, "X": un-identifiable)

i	α_{i-1}(deg)	a_{i-1}(m)	β_{i-1}(deg)	b_{i-1}(m)	d_i(deg)	θ_i(deg)
1	-0.1677	-0.0154	0.4785	0.0496	0.4853	-0.2418
2	90.0312	0.3195	-	-	0(X)	1.1777
3	-0.0249	0.871	0.0165	-	-0.0025	-1.6358
4	89.9537	0.1997	-	-	1.0282	-1.3097
5	-90.0298	-0.0004	-	-	0.0005	-2.1707
6	90.0642	0.0016	-	-	0.1850(X)	0(X)
T	-	-0.2816	-	0.0485	0.4234	-

By utilizing the Simultaneous identification of joint compliance and kinematic parameters (SKCM) that is described in the section 2.3, the kinematic and joint compliance parameters are identified. These parameters are presented in Table 2.3 and Table 2.4 including 4 joint compliance parameters and 25 kinematic parameters.

Table 2.3. D-H parameters of the Hyundai robot YS100 identified by SKCM .
("-": unavailable, "X": un-identifiable)

i	α_{i-1}(deg)	a_{i-1}(m)	β_{i-1}(deg)	b_{i-1}(m)	d_i(deg)	θ_i(deg)
1	-0.1746	-0.0167	0.5210	0.0487	0.4869	-0.228
2	90.0129	0.3205	0	0	0(X)	1.4752
3	-0.0099	0.8699	0.0082	0	-0.0014	-1.6862
4	89.989	0.2	0	0	1.0298	-1.3747
5	-90.0052	0.0002	0	0	0.0005	-2.1613
6	90.0344	0.0013	0	0	0.185(X)	0(X)
T	-	-0.2813	0	0.0484	0.4238	-

Table 2.4. Stiffness identification by SKCM

	K_2	K_3	K_4	K_5
Stiffness	1.5123*10^6	1.1614*10^6	0.2245*10^6	0.3539*10^6

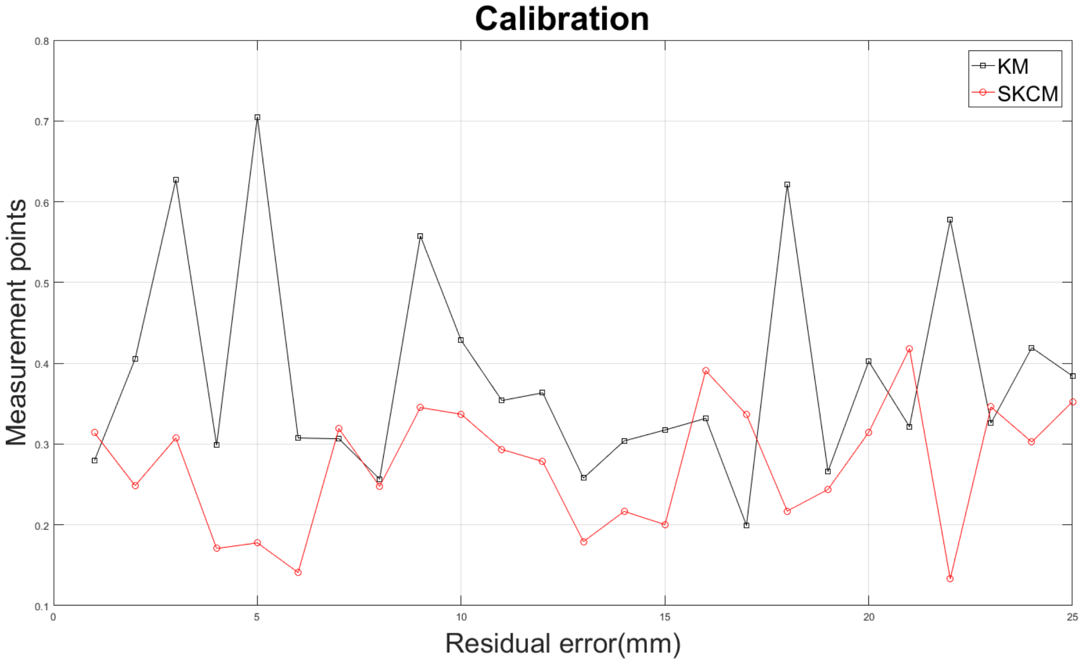


Fig. 2.4. Absolute position error of the YS100 robot after calibration.

Table 2.5. Absolute position accuracy of the YS100 robot (Calibration).

	Mean (mm)	Maximum (mm)	Std. (mm)
Nominal robot model	12.607	27.1121	5.4769
KM	0.3847	0.7049	0.1327
SKCM	0.2733	0.4180	0.0782

2.4.2.2. Experimental validation results

In order to show the general capability over the entire robot workspace, the method should be investigated by another robot configuration. The other set of 45 robot configurations (Q2) is randomly selected overall the workspace to show the general capability over the entire robot workspace of the methods.

The calibration results with Q2 data set are shown in Table 2.7 and Fig.2.6. It shows that the position errors generated by the SKCM are better than the conventional kinematic calibration method. The mean of position errors generated by the SKCM method is more precise by 98 % than the errors before calibration (from 12.2686 mm to 0.2191 mm), by 41% than the errors by KM (from 0.3842 mm to 0. 2191mm).

Table 2.6. Absolute position accuracy of the YS100 robot (Validation).

	Mean (mm)	Maximum (mm)	Std. (mm)
Nominal robot model	12.2686	23.288	5.3057
KM	0.3842	0.7930	0.1864
SKCM	0.2191	0.3799	0.0752

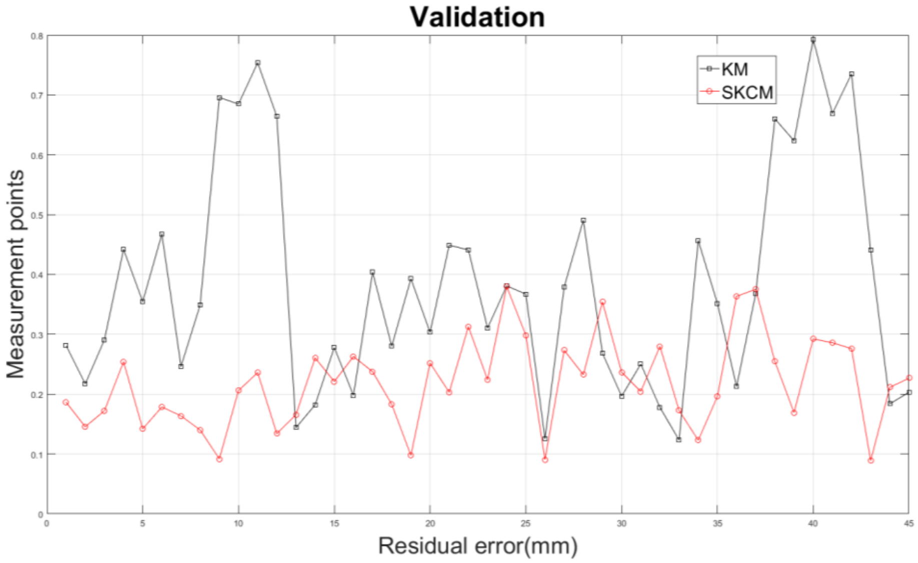


Fig. 2.5. Absolute position error of the YS100 robot after validation.

2.5 Conclusion

This chapter presented the model-based calibration methods for the industrial robot manipulator including KM, SKCM.

The conventional kinematic method is fast converging, helpful in reducing position errors due to the deviations between the nominal and actual values of kinematic parameters. However, because of neglecting the compliance errors, the residual errors after using this method is still high.

The SKCM has many advantages such as less computing time, fast convergence, and accurate knowledge of error sources. It is also precision and robustness. These calibration techniques are also performed on YS100 robot to demonstrate their effectiveness and feasibility.

Chapter 3: A NEW ROBOTIC MANIPULATOR CALIBRATION METHOD OF IDENTIFICATION KINEMATIC AND COMPLIANCE ERRORS[*]

In this chapter, a new robotic calibration method is proposed for reducing the positional errors of the robot manipulator. First, geometric errors of a robot are identified by using a conventional kinematic calibration model of the robot. Then, a radial basis function is constructed for compensating the compliance errors based on the effective torques for further increasing the positional precision of the robot. The enhanced positional accuracy of the robot manipulator in experimental studies that are carried on a YS100 robot illustrates the advantages of the suggested algorithm than the other techniques.

[*] *Le, Phu-Nguyen, and Hee-Jung Kang. "A New Robotic Manipulator Calibration Method of Identification Kinematic and Compliance Errors." International Conference on Intelligent Computing. Springer, Cham, 2020.*

3.1 Introduction

The robot manipulators are widely used in the industry. Although the robots are high repeatability, they are well-known by their low accuracy[10][1]. The errors of the robot end-effector mostly come from geometric errors and non-geometric errors. The geometric errors are the results of misalignments, incorrect in manufacturing, and assembly robot. The non-geometric errors may come from many non-geometric sources, such as joint and link compliance, temperature variation, gear transmission, etc. Among the non-geometric errors, the compliance errors are dominant. These errors are caused by the flexibility of joints and links under the link self-gravity and external payload.

Geometric calibration methods are widely examined and become mature. The most famous kinematic calibration method, the D-H model is suggested by Denavit-Hartenberg[23–25]. This method is widely used in kinematic calibration by many researchers recently[26–28]. Moreover, the other geometric calibration methods are CPC model[29][30], POE model[16][17], and the zero-reference position method[13][14]. However, these calibration methods do not consider the non-geometric errors. On the other hand, some studies used another approach to investigate joint compliance errors[20][21]. However, these methods neglected effect of the geometric errors.

Some works have been proposed to deal with kinematic and compliance calibration. For instance, a calibration method to calibrate the geometric errors and compensate the joint by radial basis function (RBF)[6] is proposed by Jang et al. However, the work [6] focused on calibrating the geometric parameters and compensating the compliance errors by compensating the joint ("joint level" calibration[10]). The work also needs to divide the robot working space into many subspaces and required many measurements and consuming a lot of time. Meggiolaro et al. proposed a method to approximate the compliance errors by a polynomial function of joint parameters and wrench using torque sensors[40]. Zhou and Hee-Jung proposed a method to simultaneously calibrate the geometric and joint stiffness parameters of the robot[22]. However, this method linearized the relationship between effective torques and joint compliance errors. Recently, some studies have been performed on joint stiffness calibration[41–43] with the need of the torques sensors.

This study proposed a new calibration algorithm for robotic manipulators. The method includes the kinematic calibration and non-geometric compensation with a RBF compensator that compensates for compliance errors based on the effective torques. It is assumed that the gravity compensation torques are nonlinearity related to the compliance errors. These relationships can be constructed by a RBF. The advantages of the suggested method are easy for implementing, removing the need for torque sensors, high ability to enhance the precision of the manipulator. These advantages are firmly confirmed by the experimental studies in contrasting with 2 other methods such as the conventional kinematic calibration and the method for simultaneously calibrate the geometric and joint stiffness parameters of the robot.

Following the introduction. section 2 presents the kinematic model of the YS 100 robot. Which is the kinematic structure of it is briefly described in Fig. 5.1 and Table 5.1. In Section 2, the geometrical and the gravity compensator using a Radial basis function that is based on the effective torques are presented. Sections 3 is devoted to the experimental calibration result of the proposed method in contrasting with other methods. Section 4 summarizes the abilities and advantages of the proposed method.

3.2. Identification kinematic parameters and compliance compensation based on the effective torques using a radial basis function.

Assuming that the robot's end-effector position P_{real} is calculated by the following equation:

$$P_{real} = P_{kin} + \Delta P_{kin} + \Delta P_{c} + \Delta P_{extra} \tag{3.1}$$

where P_{kin} is the position of the end effector calculated by the kinematic parameter, ΔP_{kin} is the position error caused by the geometric error, ΔP_c is the position error due to the joint compliance, and ΔP_{extra} is the positional residual error that is not modeled. Assuming that geometric errors and joint deflection errors are the main parts in causing the position errors P_{real} ($\Delta P_{extra} = 0$). The error model can be expressed as:

$$\Delta P = \Delta P_{kin} + \Delta P_c = P_{real} - P_{kin} \tag{3.2}$$

The position errors caused by geometric errors ΔP_{kin} in the Eq. 3.1 could be identified by the conventional kinematic calibration[23][24][25]. ΔP_{kin} can be expressed as

$$\Delta P_{kin} = J_{kin} \Delta \phi \tag{3.3}$$

where J_{kin} ($3 \times n$) is a kinematic Jacobian matrix[10][22]. $\Delta \phi$ is a $n \times 1$ kinematic parameter error vector. n is the number of the calibrated kinematic parameters. The total number of kinematic parameters is equal to 32. However, the 6 DOF revolute robot has several dependencies between some parameters. These dependency parameters are { $\Delta \theta_1, \Delta \theta_0$ },{ $\Delta d_1, \Delta d_0$ },{ $\Delta d_3, \Delta d_2$ },{ $\Delta z_T, \Delta d_6$ },{ $(\Delta x_T, \Delta y_T), \Delta \theta_6$ }. In each pair, the parameter errors cannot be identified together. Therefore, the dependency parameters that are chosen to calibrate are { $\Delta \theta_1$, Δd_1, Δd_3, Δx_T , Δy_T , Δz_T } while the other error parameter in each pair is set to the nominal parameter value. So, the number of calibrated kinematic is reduced to 27.

The Eq. 3.3 can be solved by the least-square method to overcome the effect of noise and uncertainty:

$$\Delta \phi = [(\mathrm{J}^{\mathrm{T}}\mathrm{J})^{-1}\mathrm{J}^{\mathrm{T}}] \Delta P \tag{3.4}$$

The positional error ΔP is calculated by

$$\Delta P = P_m - P_{kin} \tag{3.5}$$

where P_m is the measured position vector and P_{kin} is the computed position vector by the recent kinematic parameters. The Eq. (3.4) is employed repetitive until the geometric parameters converge. Through the kinematic calibration process, the P_{kin} converges to the P^c_{kin} value. The position errors of the robot end-effector after kinematic calibration process are calculated by:

$$\Delta P_{res} = P_m - P^c_{kin} \tag{3.6}$$

Assuming that the position errors due to joint deflection errors are the main parts in causing these residual position errors ($\Delta P_{res} = \Delta P_c$). The joint deflections under link self-gravity and external payload are also assumed to be dominant in causing compliance errors. Therefore, the joint deflection errors can be calculated from the related effective torque of joints. It should be noted that previous literatures[20],[22] constructed the compliance errors by linearizing the relationship of the effective torques and the joint compliances. However, there are some residual errors that could not be neglected caused by the nonlinear relation between joint torques and joint deflections. For further enhanced the robot precision, the relationship of the effective torque and the residual errors is constructed by a RBF in this chapter. The RBF has 6 inputs that represent the total effective torque in 6 robot joints, 40 nodes in the hidden layer, and 3 nodes in the output layer that represent three elements of the position error vector.

The total effective torques in the robot j_{th} joint under related gravity forces are given as:

$$\tau_i = \sum_{j=i}^{N+1} \tau_{i,j} = \sum_{j=i}^{N+1} J_{\theta_{i,j}}{}^T F_j \tag{3.7}$$

where N=6 is the number of DOF of the robot and F_{N+1} is the gravity force due to payload Here, the gravity force accompanying to j_{th} link is calculated by

$$F_j = \begin{bmatrix} 0 & 0 & -M_j g \end{bmatrix} \tag{3.8}$$

where M_j is the mass of the j_{th} link and g is the gravity coefficient. The transpose of the Jacobian matrix is used as a force transformation to find the effective joint torques $\tau_{i,j}$ in the i_{th} joint due to the gravity force in the j_{th} link. The Jacobian matrix is defined as

$$J_{\theta_{i,j}} = z_i \times l_{i,j} \tag{3.9}$$

where $l_{i,j}$ is the 3×1 vector between the origin of the i_{th} frame and the mass center of the j_{th} link. The total effective torques are set to be the input of the RBF. Fig. 3.1 shows the structure of the RBF. The output of the hidden node *i* in the RBF layer is calculated as follow:

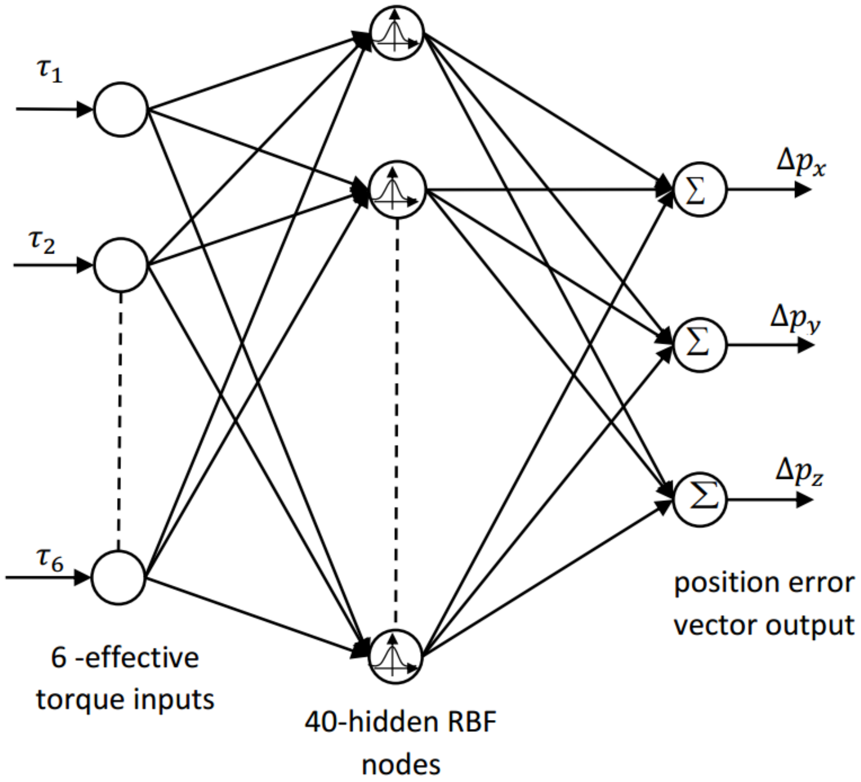


Fig. 3.1. Structure of the RBF.

$$o_j = e^{-n^2} \tag{3.10}$$

where *n* is a transfer function that describes the vector distance between the weight vector *w_i* and the input vector *p*, multiplied by the bias *b_i*.

$$n = \|\mathrm{w}_i - p\| b_i \tag{3.11}$$

The output layer is a linear function with 3 nodes in the output layer that represent three elements of the position error vector.

The output of the RBF is used to compensate for the compliance error(which is assumed to be the residual error $\Delta P_{res} = \Delta P_c$). Therefore, the residual error after compensated by the RBF is calculated by:

$$e = \Delta P_{res} - P_{nn} \tag{3.12}$$

In this work, the weights and bias of the RBF are trained by the MATLAB toolbox that creates a two-layer network. The hidden layer is the RBF layer (Eq. 3.10,11). The output layer is a linear layer. At the beginning, there are no neurons in the hidden layer. The learning process is carried out following the steps below:

- Run the network and find the input vector with the greatest error.
- A RBF neuron is added with weights equal to that vector.
- The linear layer weights are redesigned to minimize error.
- Repeat until convergence

In order to keep the RBF layer from increasing too much, the number of nodes in this layer are limited at 40 nodes. Overall, the suggested method could be described in the following flowchart.

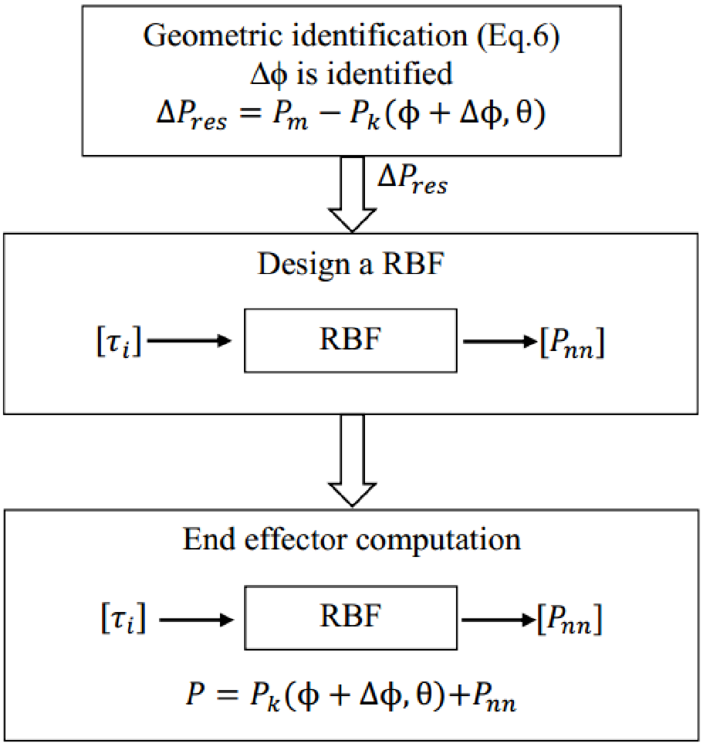

Geometric identification (Eq.6)
$\Delta\phi$ is identified
$\Delta P_{res} = P_m - P_k(\phi + \Delta\phi, \theta)$
ΔP_{res}
Design a RBF
$[\tau_i]$
RBF
$[P_{nn}]$
End effector computation
$[\tau_i]$
RBF
$[P_{nn}]$
$P = P_k(\phi + \Delta\phi, \theta)+P_{nn}$

Fig. 3.2. Flowchart of the proposed method.

3.3. Experiment and results

The experimental system is shown in Fig. 3.3. The 6 DOF robot manipulator (YS100). In this work, the mass of link jth M_j (Eq. 10) is provided by the robot's manufacturer. The external payload weight is 110 kg. Therefore, the weight matrix is descried as follow:

$$M = [196.7 \quad 79.25 \quad 170.27 \quad 10.58 \quad 22.33 \quad 2.0 \quad 110] \tag{3.13}$$

An API laser tracker (accuracy of 0.01 mm/m, repeatability of ±0.006 mm/m) and an accompanying laser reflector are used to perform the calibration process. The proposed method (KM-RBF) is used to calibrate the YS100 robot to show the advantage of the method in comparing with 2 others methods including the kinematic calibration method (KM) [23][24][25], the simultaneous identification of joint compliance and kinematic parameters methods (SKCM) [22] in the experimental study.

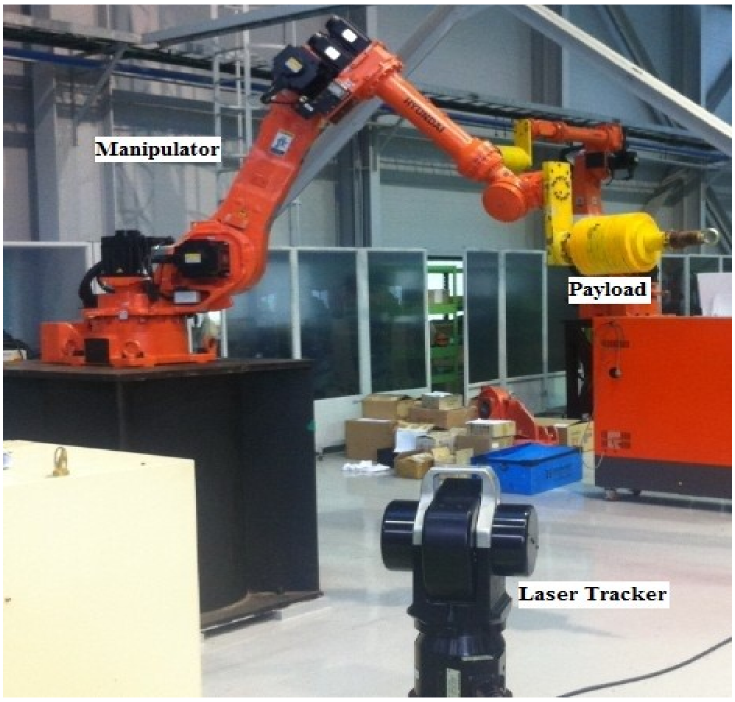


Fig. 3.3. Experimental setup.

3.3.1 Experimental calibration

The robot configuration data are randomly collected in the working space and classified into 2 sets. Set Q1 including 50 robot configurations is employed in the calibration process and the other set of 50 robot configurations (Q2) is used in the validation process. By using the conventional calibration method (Eq. 6), 27 geometric parameters are identified. The results are demonstrated in Table 3.1. The residual errors and the computed torques are used for training the RBF to determine the weights and bias of the RBF. It should be noted here the reason why the RBF is used in this working rather than the conventional fed forward neural network. In the conventional feedforward neural network, the sigmoid neurons can have outputs over a large region of the input space, while radial basis neurons only respond to relatively small regions of the input space[44]. Therefore, the RBF could be said to be more stable in responding to noises and uncertainties inputs. However, the drawback of this method is that the larger the input space the more radial basis neurons are required[45]. The experimental calibration processes are carried out by 3 different calibration methods such as conventional kinematic calibration, SKCM, and KM-RBF. The results of these calibration methods are shown in Fig. 3.4 and Table 3.2.

Table 3.1. Identified D-H parameters of the Hyundai robot YS100.

i	α_{i-1}(deg)	a_{i-1}(m)	$\beta_{i-1(}$deg)	b_{i-1}(m)	d_i(deg)	θ_i(deg)
1	-0.1646	-0.016	0.4748	0.0493	0.4851	-0.3316
2	90.0578	0.3199	-	-	0	1.1517
3	0.0004	0.8704	0.0681	-	-0.0036	-1.6686
4	89.9919	0.2001	-	-	1.0272	-1.2079
5	-90.121	0.0003	-	-	-0.0017	-0.0017
6	89.9656	-0.0035	-	-	0.185	-1.8226
T	-	-0.28	-	0.0469	0.4219	-

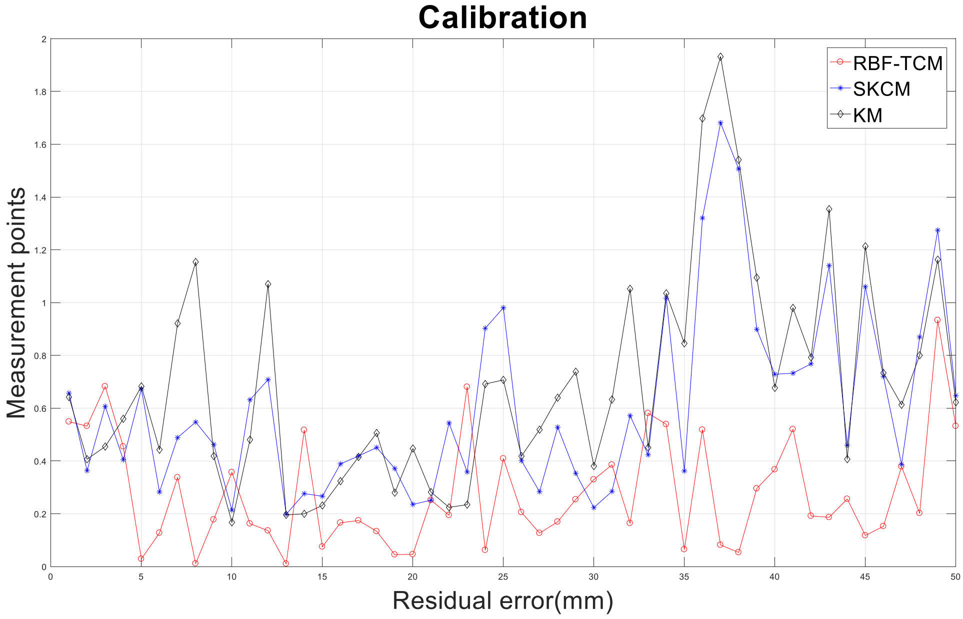


Fig. 3.4. Residual errors of the YS100 robot after calibration.

Table 3.2 The absolute position accuracy of the YS100 robot (Calibration).

	Mean (mm)	Maximum (mm)	Std. (mm)

Nominal robot model	13.5527	30.5911	6.0528
KM	0.6894	1.9318	0.4015
SKCM	0.6065	1.6811	0.3488
Proposed method	0.2785	0.9332	0.2095

The calibration results show that the precision of the robot after calibrated by the proposed method is dramatically reduced. By employing the KM-RBF method, the position errors are lower than the results by other methods. In comparing to the conventional kinematic calibration method, the proposed method reduces the mean of position errors from 0.6894 mm to 0.2785 mm (precise increasing by 59.6%). It also increases the accuracy by 54.08% in comparison to the results generated by the SKCM method (from 0. 6065 mm to 0.2785 mm). The suggested algorithm also generates the lowest maximum position error (0.9332 mm), and the lowest standard deviation (0.2095mm).

3.3.2 Experimental validation results

The proposed method should be validated by another robot configuration to demonstrate the ability of it over the working space. The robot configuration set Q2 that is totally different from Q1 is hired for the validation process with 3 different methods.

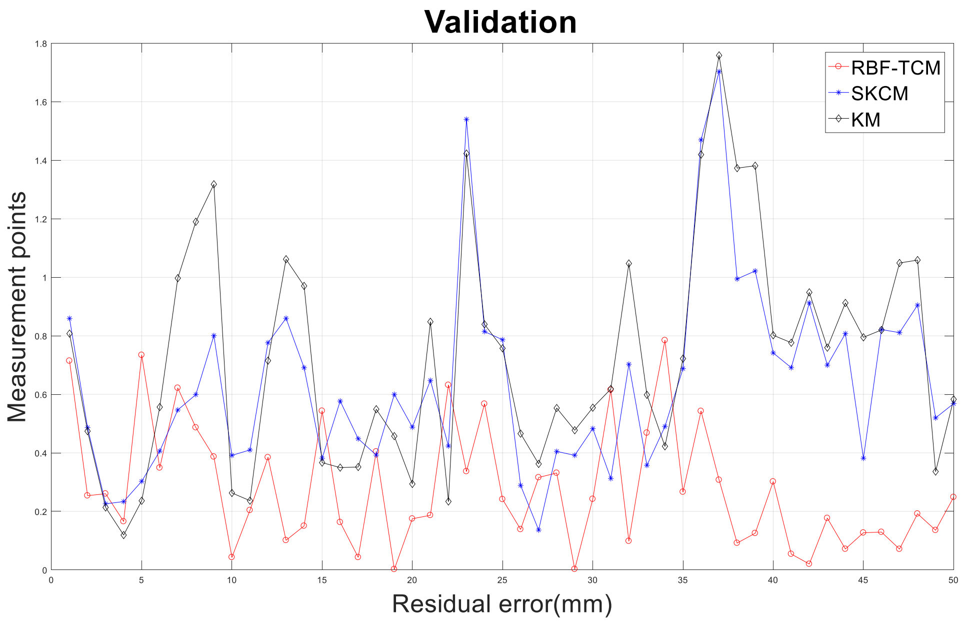


Fig. 3.5. Residual errors of the YS100 robot after validation.

Table 3.3. the absolute position accuracy of the YS100 robot (Validation).

	Mean (mm)	Maximum (mm)	Std. (mm)
Nominal robot model	14.1106	32.3303	5.9835
KM	0.7245	1.7584	0.3814
SKCM	0.6398	1.7031	0.3214
Proposed method	0.2802	0.7846	0.2084

By employing the method, the position errors are lower than the results by other methods in the validation process (Table 3.3 and Fig. 3.5). In comparing to the conventional kinematic calibration method, the proposed method reduces the mean of position errors from 0.7245 mm to 0.2802 mm (precise increasing by 61.33%). It also increases the accuracy by 56.21% in comparison to the results generated by the SKCM method (0.6398 mm to 0.2802 mm). The suggested algorithm also generates the lowest maximum position error (0.7846 mm), and the lowest standard deviation (0.2084mm).

3.3.3 Discussion

In previous literatures [20],[22], the relationship of joint deflections of robot and the effective torques are linearized:

$$\Delta\theta_c = \tau C \tag{3.14}$$

where $\Delta\theta_c$ is the $N \times 1$ joint deflection vector, τ is the diagonal effective torque matrix, C is the $N \times 1$ joint compliance vector. Then, the Cartesian position errors due to the joint compliances can be modeled as:

$$\Delta P_c = J_e \Delta\theta_c = J_e \tau C \tag{3.15}$$

where J_e is vector of joint compliance parameters that is computed by the following published work [34]. $J_e\tau$ is the transformation matrix relating the joint compliance parameters and the deflections of robot end-effector. It should be noted here that the effective torques in the i_{th} joint is not only due to the gravity force related to the i_{th} link but also due to the gravity forces related to both the link after and the external load. In this work, the relationship of the effective torque τ and the positional errors due to the compliance errors ΔP_c could be constructed by a RBF for higher increasing the precision of the robot.

3.4 Conclusion

In this work, a new robotic calibration method is proposed for reducing the positional errors of the robot manipulator. First, geometric errors of a robot are identified by using a conventional kinematic calibration model of the robot. Then, a radial basis function is constructed for compensating the compliance errors based on the effective torques for further increasing the positional precision of the robot. By using a RBF, the relationship of the effective torque and the compliance errors is constructed for higher increasing the precision of the robot. The advantages of the suggested method are easy for implementing, removing the need for torque sensors, high ability to enhance the precision of the manipulator. These advantages are firmly confirmed by the experimental studies on a YS100 robot in contrasting with 2 other methods such as the conventional kinematic calibration and the method for simultaneously calibrate the geometric and joint stiffness parameters of the robot .

Chapter 4: A NEW CALIBRATION METHOD FOR ENHANCING ROBOT POSITION ACCURACY BY COMBINING OF A ROBOT MODEL BASED IDENTIFICATION APPROACH AND AN ANN-BASED ERROR COMPENSATION TECHNIQUE[*]

Robot position accuracy plays a very important role in advanced industrial applications. This chapter proposes a new method for enhancing robot position accuracy. In order to increase robot accuracy, the proposed method models and identifies determinable error sources, for instance, geometric errors and joint deflection errors. Because non-geometric error sources such as link compliance, gear backlash, and others are difficult to model correctly and completely, an artificial neural network (ANN) is used for compensating for the robot position errors, which are caused by these non-geometric error sources. The proposed method is used for experimental calibration of an industrial Hyundai HH800 robot designed for carrying heavy loads. The robot position accuracy after calibration demonstrates the effectiveness and correctness of the method.

[*] *Nguyen, Hoai-Nhan, Phu-Nguyen Le, and Hee-Jun Kang. "A new calibration method for enhancing robot position accuracy by combining a robot model–based identification approach and an artificial neural network–based error compensation technique." Advances in Mechanical Engineering 11.1 (2019): 1687814018822935.*

4.1 Introduction

In the offline programming (OLP), the joint angles are desired to achieve a given Cartesian position of the arm's tip by using a kinematic model of the robot arm. However, a nominal geometrically model according to a product specification sheet does not include the errors arising in manufacturing or assembly. In addition, it also does not include non-geometric errors, such as gear transmission errors, gear backlashes, arm compliance, etc., which are difficult to geometrically consider in a kinematic model. Therefore, some methods of calibrating precisely kinematic and non-kinematic parameters are required to improve the accuracy of robot manipulators.

Previously published studies have focused on modeling robot error sources for purposes of calibration [1, 7, 9, 10, 22, 25, 46–51]. These errors are classified into two types: geometric errors and non-geometric errors. Geometric errors include link twist angle errors, link length errors, link offset and joint angle offsets. Non-geometric errors include gear backlash, joint deflection, link compliance, etc. Some researchers have focused on kinematic model of robots but took into account only geometric errors [10, 25, 48–50]. Other researchers[51][7] developed robot kinematic models that included both link geometric and non-geometric errors (only joint deflection)[22]. However, there are other non-geometric error sources that significantly affect the robot accuracy; therefore, we cannot leave out any non-geometric error parameters. Some research has suggested that robot joint deflection and link compliance affect position accuracy[46][47]. The deformation of robot joints and links is due to robot link weights and carried payloads. Previously, Dulen et al.[46] applied a theory of flexible beams to investigate the effects of robot link compliance. These effects were demonstrated by changing six differential elements (three for translation changes and three for rotation changes). Hudgen et al.[47] utilized a methodology for identifying general robot deflection characteristics under applied torques and forces. These methods resorted to special tools and sensors for finding compliance error elements. Both works[46][47] should include other non-geometric errors for more accurate robot calibration.

Recently, the extended Kalman filter (EKF) algorithm[52], genetic algorithm[53], maximum likelihood estimation[3][54], particle filter (PF) algorithms[32] and Fourier polynomials[55] are also applied for identified the nonlinear kinematic parameters. Some researchers have been developing the hybrid calibration algorithms, Zhihong et al.[32] suggested a combination algorithm including the EFK and PF. This research based on the EKF algorithm providing priori value, the PF algorithm was used to calibrate the robot kinematic parameters. Although the method has advantages of working for high dimensional systems, it still has some remains drawbacks. It does not include the non-geometric error sources such as link compliance, gear backlash, and other. Acilli et al.[55] applied Fourier polynomial for compensating the non-geometric error. Although, these methods were successful used in robot calibration, they do not supply knowledge of error sources in robot structure, has slow convergence, and the calibration space is limited. Artificial Neural Networks (ANN) have also been widely using for approximating non-linear relationships in systems or industrial robots because of their advantageous properties such as learning ability, adaptation, and flexibility. Some researchers have applied the ANN to compensate for robot position errors[6][38][56][57][58]. Jang et al.[6] used a Radial Basis Function Network (RBFN) to form a functional relation to describe the robot joint offset errors in terms of their joint positions. The works[6][56] have utilized an ANN in order to describe the functional relationship of end-effector positions and corresponding position errors. However, the position errors essentially depend on robot configurations. It has been proven in previous works [59][60] that a robot can approach a given Cartesian coordinate with multiple configurations and each configuration will produce a different position error. The work[57] has applied a multiple layer ANN to approximate the relationship between robot joint positions and associated joint offset errors. The application of ANN for robot error compensation still has some shortcomings because a user cannot know the robot error sources. However, some errors can be modeled easily and correctly, for examples, the robot link geometric errors and the robot joint compliance are easily modeled.

The model-based calibration method, mentioned above, has many advantages such as less computing time, fast convergence, and accurate knowledge of error sources. Some error sources (especially non-geometric errors) cannot, however, be determined and modeled correctly. Therefore, the robot position error, which is caused by these error sources, should be compensated for by using an ANN. The combination of both model-based calibration and ANN compensation methods can be an effective solution for enhancing robot position accuracy.

In this chapter, we present a technique for the calibration of industrial robots by combining the advantages of the two above methods. The first one is a model-based calibration method and the second is position error compensation using an ANN. First, we model and identify robot kinematic error parameters, including geometric errors and joint compliance errors. Second, the robot residual position errors, which are caused by other non-geometric errors such as link deflection, gear backlash, etc., are compensated for by using an ANN. By comparing with the work[6], the proposed method can also include the robot joint compliance errors in the robot model. The ANN, which was built with the proposed method, also described a more appropriate relationship between the robot joint readings and its end-effector positions instead of the unsuitable one used in the work[6]. Experimental calibration for the Hyundai HH800 robot was carried out to demonstrate the effectiveness and correctness of the proposed method.

The remainder of the chapter is organized as follows: Section 4.2 develops a robot kinematic model for use in calibration. Section 4.3 derives formulas for the identification of robot kinematic parameters including geometric parameters and joint compliance. Section 4.4 constructs an ANN and shows how to compensate for the robot position errors. Section 4.5

describes experimental calibration and results. Finally, some conclusions are presented in Section 4.6.

4.2. Kinematic Model of the HH800 Robot

This section describes kinematic model for the Hyundai HH800 robot (Fig. 4.1). Sub-section 4.2.1 describes building the robot kinematic model and sub-section 4.2.2 describes modeling the robot joint compliance.

4.2.1 Kinematic model of HH800 robot

The 6 degree of freedom (dof) HH800 robot, which is described in Fig. 4.1, consists of a main open kinematic chain (6 dof) and a closed loop mechanism (2 dof). The open chain is connected by the revolute joints 1, 2, 3p, 4, 5, and 6. The closed loop PQRS is connected by the revolute joints 2, 3, Q, R and S (both joints' axes 2 and 3 are coincident at point P). The frames are fixed at the open chain's links based on the Danevit-Hartenberg (D-H) convention[11]. The nominal D-H parameters of the open chain and the closed loop are shown in Table 4.1.

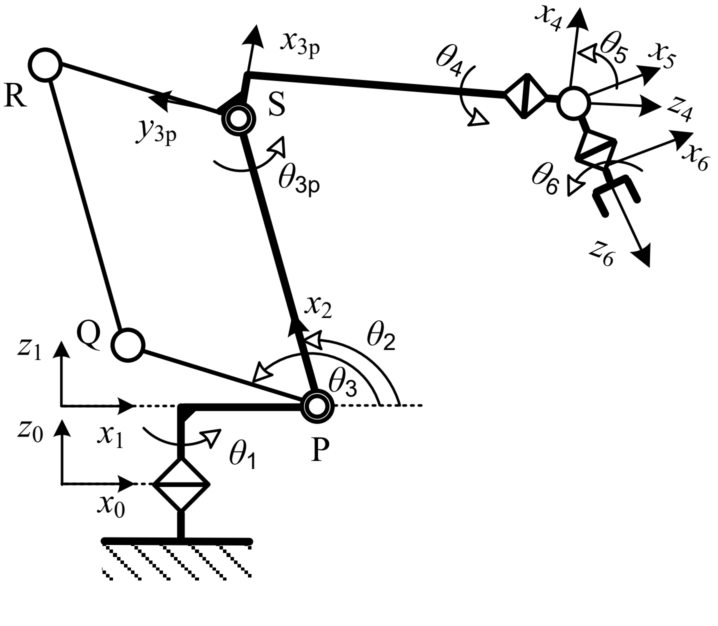


Fig. 4.1. A sketch of the HH800 robot and its link frames

A transformation matrix of two consecutive link frames {i-1} and {i} is computed by:

$$ {}^{i-1}_{i}T = Rot(x_{i-1}, \alpha_{i-1}).Tr(x_{i-1}, a_{i-1}).Tr(z_i, d_i).Rot(z_i, \theta_i) \tag{4.1} $$

where robot link parameters include the twist angle α_{i-1}, link length a_{i-1}, link offset d_i, and joint variable θ_i; $Rot\ (\cdot)$ and $Tr\ (\cdot)$ are pure rotation and translation matrices about and along an axis, respectively[23].

Table 4.1. D-H parameters of HH800 robot (units: length [m], angle [º]; "-" unavailable)

DH parameters of the main open chain

i	α_{i-1}	a_{i-1}	β_{i-1}	b_{i-1}	d_i	θ_i
1	0	0	0	0	1.2	θ_1
2	90	0.515	-	-	0	θ_2
3	0	1.6	0	-	0	θ_3
4	90	0.35	-	-	1.9	θ_4
5	-90	0	-	-	0	θ_5
6	90	0	-	-	0.445	θ_6
Tool	-	0.2	-	0.2	0.2	-

Link lengths of the closed loop

L_5	0.8	L_4	1.6	L_3	0.8

A homogenous transformation from the robot base frame to the end-effector frame, which describes the pose of the end-effector with respect to the base frame, is computed as follows:

$$ {}^{0}_{E}T = {}^{0}_{1}T(\theta_1).{}^{1}_{2}T(\theta_2).{}^{2}_{3p}T(\theta_{3p}).{}^{3p}_{4}T(\theta_4).{}^{4}_{5}T(\theta_5).{}^{5}_{6}T(\theta_6).{}^{6}_{E}T \tag{4.2} $$

where ${}^{i-1}_{i}T$ is computed by Eq. (4.1), i = 2, …, 6. The matrix ${}^{0}_{1}T$ is computed by

$$ {}^{0}_{1}T = Tr(x_0, a_0).Tr(y_0, b_0).Tr(z_1, d_1).Rot(x_0, \alpha_0).Rot(y_0, \beta_0).Rot(z_1, \theta_1) \tag{4.3} $$

the matrix ${}^{6}_{E}T$ is computed by:

$$ {}^{6}_{E}T = Tr(x_6, a_6).Tr(y_6, b_6).Tr(z_7, d_7) \tag{4.4} $$

and the matrix ${}_{3p}^{2}T(\theta_{3p})$ in Eq.(4.2) is computed with the passive joint angle θ_{3p}. The angle θ_{3p} is found by solving a system of constraint equations of the closed loop PQRS (Fig. 4.2) with given values of active joint angles θ_2 and θ_3.

For simplicity, the closed loop actuating mechanism PQRS can be considered as a planar mechanism. The position constraints for the closed loop are formed with respect to the plane O_{1x1z1} as follows[61][62]

$$\begin{aligned} a_2 c\theta_2 + L_3 c\theta_{2,3p'} - L_5 c\theta_3 - L_4 c\theta_{3,4p} = 0 \\ a_2 s\theta_2 + L_3 s\theta_{2,3p'} - L_5 s\theta_3 - L_4 s\theta_{3,4p} = 0 \end{aligned} \tag{4.5}$$

where $c\theta_i$and $s\theta_i$ are short forms of $cos(\theta_i)$ and $sin(\theta_i)$, respectively, and $\theta_{h,k} = \theta_h + \theta_k$, $L_5 = PQ, L_4 = QR, L_3 = RS, a_2 = SP.$

The closed loop PQRS is a parallelogram, therefore it has properties such as $L_4 = a_2$ and $L_5 = L_3$ (Fig. 4.1 and 4.2). The passive joint position θ_{3p} can be solved from Eq. (4.5) with given joint variables θ_2 and θ_3 as follows: $\theta_{3\mathrm{p}} = \theta_{3\mathrm{p}'} - \xi$, where $\theta_{3\mathrm{p}'} = \theta_3 - \theta_2$and ξ = 90º is a constant angle.

It is worth to mentioning that when the closed mechanism degenerates (dashed line in Fig. 4.2), its actual link lengths differ from its nominal values. The passive joint position can be computed by $\theta_{3\mathrm{p}} = \theta_{3\mathrm{p}'} - \xi$, where $\theta_{3p'}$ is determined by solving Eq. (4.5) with the actual link lengths of the closed loop PQRS.

Because the robot kinematic model in Eq. (4.2) is used for calibration, a modification of a link transformation is required if two joint axes of this link are parallel (or nearly parallel). This transformation is modified as in the research [25] to satisfy the properties of the calibration robot model: complete, proportional, and continuous [58]. Particularly, the transformation matrix ${}_{3p}^{2}T$ from frame {2} to frame {3} (the axes z_2 and z_3 are parallel) is modified as follows:

$${}_{3p}^{2}T = Rot(x_2, \alpha_2).Tr(x_2, a_2).Rot(y_2, \beta_2).Rot(z_{3p}, \theta_{3p}) \tag{4.6}$$

where β_2 is the link twist angle about the axis y_2.

4.2.2 Robot joint compliance model

In a robot static configuration, a robot joint torque causes a twist deformation about a rotation shaft (which represents the entire drive train from the motor to the associated robot link). We can consider the shaft as a torsion spring. In this section, we study characteristics of a torsion spring because they are related to modeling of rotational joint compliance. The characteristics of torsion springs are basically presented by non-linear functions, for example $M = k(\delta\theta)^3$, where M is the applied torque on the spring, $\delta\theta$ is the spring rotational deformation, and k is the torsion spring's stiffness parameter[23]. Because the robot joint deformation is small, we can assume that the functional relationship between the joint torque and its deformation is linear.

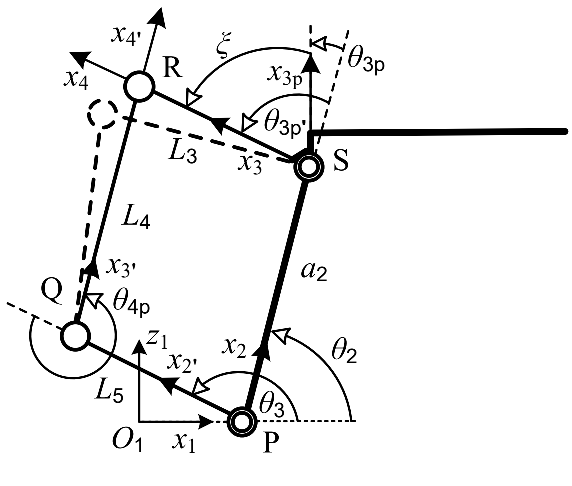


Fig. 4.2. The closed mechanism PQRS of the HH800 robot (dashed lines represent a degenerated mechanism)

The model of robot joint deformation, which is caused by the joint torque, can be represented by:

$$\delta\theta_i = \frac{1}{k_i} M_i = s_i M_i \tag{4.7}$$

where Mi [Nm] is the joint torque applied at joint axis i, k_i [Nm/rad] is the stiffness parameter of joint i, s_i [rad/Nm] is the compliance parameter (it is the inverse ratio of the joint stiffness parameter) of joint i, $\delta\theta_i$ [°] is the deformation angle of joint i under applied torque M_i, i = 1, …, 6 (for 6 dof robots).

For serial robots with one closed loop mechanism such as the Hyundai HH800 robot, the active joint torques are computed by following the methods presented in the studies[62][34]. In order to compute the robot joint torques of active joints, we applied a virtual work principle and a technique for analyzing the original robot to form an open tree-structure robot. By assuming that the joint R of the closed loop PQRS (Figs 4.1 and 4.2) is cut open, we can obtain an open tree-structure mechanism that consists of two open chains. The first chain is connected by the joints θ_1, θ_2, θ_{3p}, θ_4, θ_5, and θ_6. The second chain is connected by the joints θ_1, θ_3, Q, and R. As a result, the active robot joint torques are M_1, M_2, M_3, M_4, M_5, and M_6. Computation of the static torques due to link gravities and payload requires knowledge of the robot dynamic parameters, such as link weights, link mass centers' positions, and payload. Hyundai supplied us with these dynamic parameters, so we can use them directly without need of a robot dynamic parameter identification procedure.

In this section, we built the robot kinematic model consisting of geometric and joint compliance parameters. The next section presents a formulation for the identification of these parameters.

4.3. Formulation for Parameter Identification

The robot calibration process identifies robot kinematic model parameters that describe exactly the kinematics of a physical robot, and consequently enhance the robot position accuracy. This section describes robot kinematic parameter identification. A robot kinematic model consists of link geometric and joint compliance parameters.

Because the robot consists of a main open chain and a closed loop mechanism (Fig. 4.1), in order to identify the kinematic parameters of the entire robot kinematic model, robot identification equations are formulated by integrating the differential equations of the open chain and the closed loop via the passive joint angle θ_{3p}.

Differential transformation of the open kinematic chain can be obtained by differentiating Eq. (4.2) in term of its kinematic parameters as follows[48]:

$$\Delta X = J.\Delta\phi \tag{4.8}$$

where ΔX is a (3×1) vector of 3 position errors of the robot end-effector. $\Delta\phi = [\Delta\alpha\Delta a\Delta\beta\Delta b\Delta d\Delta\theta]^T$ is a (p×1) vector of kinematic errors belonging to the open chain (p is the number of open chain parameters), where $\Delta\alpha$ is a (N_α×1) vector of link twist errors $\Delta\alpha_i$ about the x axis, Δa is a (N_a×1) vector of link length errors Δa_i, $\Delta\beta$ is an (N_β×1) vector of link twist errors $\Delta\beta_i$ about the y axis, Δb is a (N_b×1) vector of link length errors Δb_i, Δd is a (N_d×1) vector of link offset errors Δd_i, and $\Delta\theta$ is a (N_θ×1) vector of joint offsets $\Delta\theta_i$. The total number of parameters is $p = N_\alpha + N_a + N_\beta + N_b + N_d + N_\theta$ (N_α, $\mathrm{N_a}$, N_β, N_b, N_d, and N_θ are numbers of parameters α, a, β, b, d, and θ, respectively). $J = [J_\alpha\, J_a\, J_\beta\, J_b\, J_d\, J_\theta]$ is a (3×p) matrix that relates the column vectors ΔX and ΔP. $J_\alpha = [J_{\alpha_i}]$, $J_a = [J_{a_i}]$, $J_\beta = [J_{\beta_i}]$, $J_b = [J_{b_i}]$, $J_d = [J_{d_i}]$, and $J_\theta = [J_{\theta_i}]$ are the sub-matrices whose columns are computed by following the published work[34]:

$$\begin{gathered} J_{\alpha_{i-1}} = x_{i-1} \times p_{i-1}, J_{a_{i-1}} = x_{i-1}, J_{b_{i-1}} = y_{i-1} \\ J_{\beta_{i-1}} = y_{i-1} \times p_{i-1}, J_{d_i} = z_i, J_{\theta_i} = z_i \times p_i \end{gathered} \tag{4.9}$$

where x_{i-1}, y_{i-1} and z_i are directional vectors of the link frames $\{i$-1$\}$ and $\{i\}$. p_i is a (3×1) vector of the end-effector position with respect to the link frame $\{i\}$.

The position constraints in Eq. (4.5) of the closed loop PQRS (Fig. 4.2) can be expressed in the following form:

$$\theta_{3p} = f(\theta_2, \theta_3, L_5, L_4) \tag{4.10}$$

By differentiating the variable θ_{3p} in terms of the parameters $\theta_2, \theta_3, L_5, L_4$, we got the following equation:

$$\Delta\theta_{3p} = \frac{\partial f}{\partial\theta_2}\Delta\theta_2 + \frac{\partial f}{\partial\theta_3}\Delta\theta_3 + \frac{\partial f}{\partial L_5}\Delta L_5 + \frac{\partial f}{\partial L_4}\Delta L_4 \tag{4.11}$$

By the expansion of Eq. (4.8) we have the following equation:

$$\Delta X = J_{\alpha_0}\Delta\alpha_0 + \ldots + J_{a_2}\Delta a_2 + \ldots + J_{\theta_2}\Delta\theta_2 + J_{\theta_{3p}}\Delta\theta_{3p} + J_{\theta_4}\Delta\theta_4 + \ldots, \tag{4.12}$$

The identification equation of kinematic parameter errors (of the open chain and the closed loop) can be obtained by integrating Eq. (4.11) into Eq. (4.12) via the passive joint parameter θ_{3p}. Substituting Eq. (4.11) into Eq. (4.12), we have the following equation:

$$\Delta X = J_{\alpha_0}\Delta\alpha_0 + \cdots + J_{a_2}\Delta a_2 + \cdots + J'_{\theta_2}\Delta\theta_2 + J'_{\theta_3}\Delta\theta_3 + J_{\theta_4}\Delta\theta_4 + \cdots$$

$$+J'_{L_5}\Delta L_5 + J'_{L_4}\Delta L_4, = J'.\Delta P', \tag{4.13}$$

where column vectors J'_{θ_2}andJ'_{θ_3} correspond to the parameters θ_2 and θ_3, which are belong to both the open chain and the closed loop, respectively. Column vectors J'_{L_5} and J'_{L_4} correspond to the parameters L_5 and L_4, respectively. These column vectors are computed as follows:

$$J'_{\theta_2} = J_{\theta_{3p}}\frac{\partial f}{\partial\theta_2} + J_{\theta_2}, J'_{\theta_3} = J_{\theta_{3p}}\frac{\partial f}{\partial\theta_3}, J'_{L_5} = J_{\theta_{3p}}\frac{\partial f}{\partial L_5}, J'_{L_4} = J_{\theta_{3p}}\frac{\partial f}{\partial L_4} \tag{4.14}$$

Equation (4.13) describes the differential relationship between robot geometric errors and its end-effector position errors. In order to take into account the effects of its joint compliance errors, the identification equation (4.13) has to be modified as follows:

$$\Delta X = J'.\Delta\phi' + J'_{\theta}.\delta\theta \tag{4.15}$$

where a Jacobian matrix $J'_{\theta} = [J'_{\theta_i}]$ is computed by Eq. (4.9) and i = 1, …, 6. $\delta\theta = [\delta\theta_i]^T$ is a vector of joint compliance parameters:

$$[\delta\theta_i] = diag[\tau_i].[c_i]^T = \tau.C \tag{4.16}$$

$\tau = diag[\tau_i]$, τ_i is the i^{th} joint torque; $c = [c_i]^T$ is the column vector of joint compliance parameters c_i, $i = 1, \ldots, 6$.

Substituting Eq. (4.16) into Eq. (4.15), after some manipulations we obtain the identification equation of the robot kinematic errors, which comprises the geometric errors and joint compliance parameters:

$$\Delta X = [J' \quad J'_\theta . diag[\tau_i]] \begin{bmatrix} \Delta\phi' \\ C \end{bmatrix} \tag{4.17}$$

Table 4.2. Identified D-H parameters of HH800 robot (units: length [m], angle [º]; "-": unavailable, "×": unidentifiable)

DH parameters of the main open chain

i	α_{i-1}	a_{i-1}	β_{i-1}	b_{i-1}	d_i	θ_i
1	0.8714	0.0005	0.3554	0.0001	0.0972	0.3569
2	89.942	0.5155	-	-	-0.0016	-0.8934
3	-0.0019	1.6022	0.079	-	0 (×)	-0.9231
4	90.162	0.3539	-	-	1.8830	3.2731
5	-90.103	0.0007	-	-	-0.0002	2.8017
6	90.078	-0.0004	-	-	0.445 (×)	0 (×)
Tool	-	-0.4514	-	0.0117	0.9321	-
Link lengths of the closed loop						
L_5	0.7996	L_4	1.602	L_3	0.8 (×)	

The robot model that is used for the calibration includes many kinematic parameters. Only some of them, however, are identifiable. In order to determine the identifiability of the parameters, a technique from the previous study[63] is applied. This technique uses two measures, mag(p_i) and var(p_i), which are based on the singular value decomposition of the identification matrix ***J***, where p_i is the i^th robot parameter. Among the 30 parameters of the robot open kinematic chain that are listed in Table 4.2, d_2 and d_6 are the coupling parameters of d_3 and d_7, respectively; a_6 and b_7 are the coupling parameters of θ_6. For the robot closed-loop mechanism, it is necessary to fix one length parameter (L_3 is selected to be fixed) to scale the size of the mechanism[34]. The coupling parameters cannot be determined together. The sign "×" in Table 4.2 indicates the indeterminable parameters. Thus, only 29 kinematic parameters of the HH800 robot are observable ($n = 29$).

The robot end-effector position errors are caused by many kinematic error sources which can be classified into geometric and non-geometric errors. Eq. (4.17) is utilized for identifying the robot geometric errors and partial non-geometric error (joint compliance errors due to the effect of gravity of the robot link and payload). However, other non-geometric error sources such as gear backlash, link deflection, etc. should be taken into account because they also affect the robot position accuracy. It is very difficult to model all the non-geometric errors completely and exactly. Therefore, the residual position errors after geometric and joint compliance correction should be compensated for by using an ANN. Application of the ANN for further enhancing robot accuracy is presented in the next section.

4.4. ANN and Its Application for Robot Position Error Compensation

The robot position errors after correction using the identified link geometric and joint compliance errors are still large. The residual position errors are supposed to be caused by undetermined non-geometric error sources, which have extreme non-linearity[64]. In order to increase the robot accuracy, an ANN will be used to compensate for the residual position errors.

In this study, an ANN that has three layers is constructed as in Fig. 4.3. The input layer consists of 6 nodes. These nodes represent 6 robot joint positions $[\theta_i]$, $i = 1, \ldots, 6$. The hidden layer consists of 20 nodes that have a "tan-sigmoid" activating function. The output layer consists of 3 nodes which have “linear” activating function. These nodes represent 3 elements of a robot position error vector.

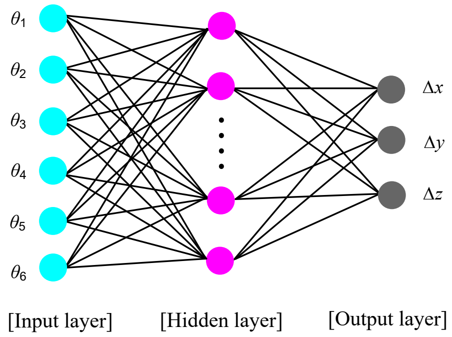


Fig. 4.3. Construction of a three layers ANN

The ANN needs to be trained prior to applying for non-geometric error compensation. A back-propagation algorithm is used to train the ANN[65][66]. The training data includes pairs of input and output values. The inputs are robot joint encoder readings. The outputs are the residual position errors after correction of the robot geometric and joint angle deflection errors.

Equation (4.15) describes the relation between the robot end-effector position errors and its kinematic errors (geometric errors and joint compliance parameters). This equation should be manipulated suitably into Eq. (4.17) for identifying of the vector of the joint compliance parameter $s=[s_i]^T$. The solution of equation Eq. (4.17) in the sense of least squares is expressed as follows:

$$\begin{bmatrix} \Delta\phi' \\ s \end{bmatrix} = (H^T H)^{-1} H^T \Delta P$$

$$H = [J' \quad J'_{\theta}.diag[\tau_i], \quad \Delta P = P_m - P] \tag{4.18}$$

where $\boldsymbol{P}_m$ is a (3×1) vector of the end-effector position measurement. $\boldsymbol{P}$ is a (3×1) vector of the end-effector position that is computed with the current robot kinematic parameters. ΔP is a (3×1) total position error vector, which is caused by the robot geometric and non-geometric errors, and is expressed as follows:

$$\Delta P = \Delta P_k + \Delta P_c + \Delta P_{extra} \tag{4.19}$$

where ΔP_k is the robot end-effector position error caused by its link geometric errors, ΔP_c is the robot end-effector position error caused by its joint angle deflection, and ΔP_{extra} is the robot end-effector position error caused by other non-geometric error sources (for examples link deformation, gear backlash, etc.).

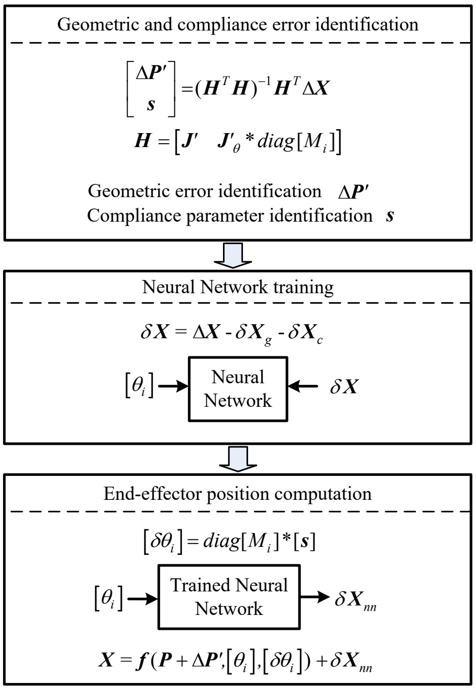

Geometric and compliance error identification
$\begin{bmatrix} \Delta \boldsymbol{P}' \\ \boldsymbol{s} \end{bmatrix} = (\boldsymbol{H}^T \boldsymbol{H})^{-1} \boldsymbol{H}^T \Delta \boldsymbol{X}$
$\boldsymbol{H} = [\boldsymbol{J}' \quad \boldsymbol{J}'_\theta * diag[M_i]]$
Geometric error identification $\Delta \boldsymbol{P}'$
Compliance parameter identification $\boldsymbol{s}$
Neural Network training
$\delta \boldsymbol{X} = \Delta \boldsymbol{X} - \delta \boldsymbol{X}_g - \delta \boldsymbol{X}_c$
$[\theta_i]$
Neural Network
$\delta \boldsymbol{X}$
End-effector position computation
$[\delta\theta_i] = diag[M_i] * [\boldsymbol{s}]$
$[\theta_i]$
Trained Neural Network
$\delta \boldsymbol{X}_{nn}$
$\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{P} + \Delta \boldsymbol{P}', [\theta_i], [\delta\theta_i]) + \delta \boldsymbol{X}_{nn}$

Fig. 4.4. A methodology for robot end-effector positioning

The correction process for the robot position is depicted by a flowchart as in Fig. 4. 4. First the robot link geometry and joint compliance errors are identified simultaneously by Eq. (4.18). After that the residual robot position errors after compensation for link geometry and joint compliance errors are used for training the ANN. The residual position errors are supposedly caused by other unmodeled non-geometric error sources such as robot link deformation, gear backlash, etc. The modified robot model and the trained ANN are used to compute the robot end-effector positions for a given robot joint position reading.

There are many methods of compensation for robot position error. Among them is a so-called “false target” technique[3], which is used widely for robot off-line programming. The off-line program contains a set of modified robot joint commands, which is used to guide the robot to its desired position in the workspace with a specific accuracy.

4.5. Experiment and Results

The proposed method is applied for the experimental calibration of a Hyundai HH800 robotic manipulator. The enhanced position accuracy of the robot after calibration proves the effectiveness and correctness of the proposed method. This section presents orderly steps of experimental calibration such as arrangement of the robot calibration system, measurement process, calibration results, and validation of robot accuracy after calibration.

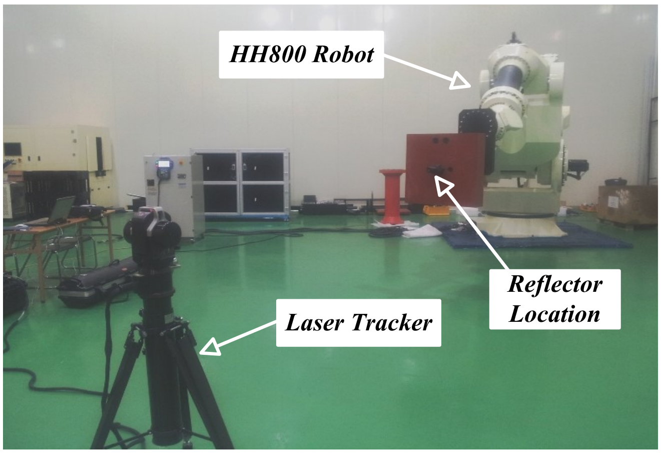


Fig. 4.5. Calibration setup of the Hyundai HH800 robot

A robot calibration system consists of a Hyundai HH800 robot (6 dof) which has one closed-loop actuating mechanism, a 3D point sensing device (API Laser Tracker, measurement accuracy of 0.01 mm/m, repeatability of +/-0.006 mm/m), and an accompanying laser reflector. The reflector is fixed at a particular location of the robot end-effector. The system is arranged as shown in Fig. 4.5.

In order to acquire suitable measurement data for robot parameter identification and ANN training, the robot workspace is divided into 10 sub-workspaces which are depicted in Fig. 4.6. In each sub-workspace (the central angle is 18º), the robot moves its end-effector to 50 positions such that they entirely cover the sub-workspace. The three-dimensional coordinates of the end points are measured by the Laser Tracker and saved in a computer. At the same time, the associated robot joint readings also are recorded. For the other sub-workspaces, the measurement procedure is performed similarly. As a result, a set of 500 end-effector positions and a set of 500 robot joint angle readings are acquired. These measurements will be grouped as follows: a set of 4×10 = 40 end-effector positions (referred to as Q_1) for parameter identification is collected from the sub-workspaces (4 endpoints from each of 10 sub-workspaces).

A set of 20×10 = 200 (referred to as Q2) robot endpoints for training the ANN is collected from the sub-workspaces (20 endpoints from each of 10 sub-workspaces). A set of 20×10 = 200 (referred to as Q3) arbitrary endpoints for robot accuracy validation is collected from the sub-workspaces (20 end-points from each of 10 sub-workspaces).

Elements of the sets Q1 and Q2 should be totally different. It is worth to noting that the sets Q1 and Q2 of end-points are selected such that these points have a uniform distribution within each sub-workspace.

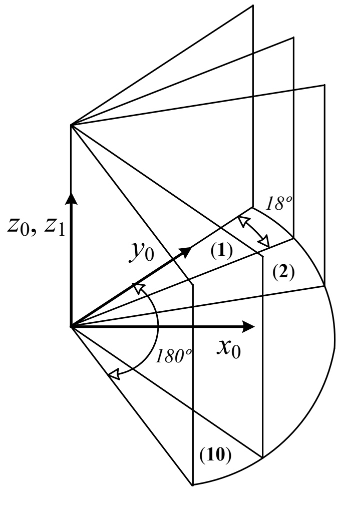


Fig. 4.6. An arrangement for collecting robot end-effector measurements in each sub-workspace

Table 4.3. Absolute position accuracy of the HH800 robot (calibration)

	Mean [mm]	Std. [mm]	Max. [mm]
Before calibration (nominal robot model)	4.0654	0.8803	6.3291
After robot link geometry and joint compliance compensation	0.6919	0.3244	1.9312
After robot non-geometric error compensation by ANN	0.3264	0.1275	0.8430

Table 4.4 Absolute position accuracy of the HH800 robot (validation)

	Mean [mm]	Std. [mm]	Max. [mm]
Before calibration (nominal robot model)	4.0629	0.8451	6.1681
After robot link geometry and joint compliance compensation	0.7035	0.3154	2.1372
After robot non-geometric error compensation by ANN	0.3959	0.1799	0.8928

In the identification process, we first identify the robot link geometric errors and joint compliance parameters by using the measurement set Q_1. By using the whole set Q_1, we can define an over-determined system of 120 ($3 \times 40 = 120$) differential equations based on Eq. (4.17). The solution of Eq. (4.18) of this system of equations in the sense of least squares is identified link geometry and joint compliance parameters[67]. The total number of identifiable parameters is 29 (25 geometric parameters and four joint compliance parameters, s_2, s_3, s_4, s_5). With this type of robot, the first joint axis is vertical, so the s_1 parameter is not taken into account because the torsional deformation about the first axis is so small compared with other joint axes. The joint deformation about the sixth axis is not identifiable because of dependence characteristics associated with other parameters[63]. The identified values of these geometric parameters are listed in Table 4.2. The compliance parameters of joints 2, 3, 4, 5 are $s_2 = (1/2.348) \times 10^{-7}$ [rad/Nm], $s_3 = (1/5.702) \times 10^{-6}$ [rad/Nm], $s_4 = (1/1.728) \times 10^{-6}$ [rad/Nm], and $s_5 = (1/2.074) \times 10^{-6}$ [rad/Nm], respectively.

After compensating for the robot link geometry and joint compliance errors, the residual end-effector position errors are still fairly large. The residual errors are supposedly caused by the other non-geometric error sources (for example, the robot link deformation, gear back-lash, and so on). In order to reduce the residual position errors, the ANN is applied to compensate for these robot non-geometric error sources. Because the set Q_2 is used for training the ANN, and in order to prepare the training data, the residual position errors are computed for a total of 200 positions of the set Q_2. The position errors are displayed in Fig. 4.7. This figure shows the residual robot end-effector position errors after geometry and joint compliance error compensation. The ANN should be trained before it is used for error compensation.

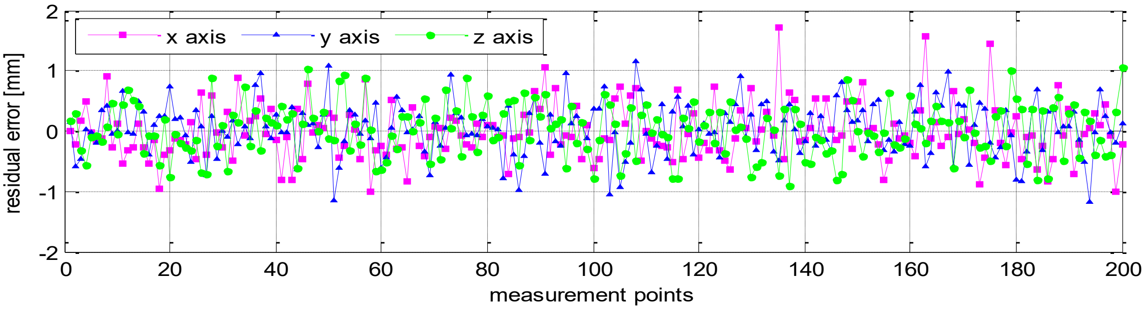


Fig. 4.7. The residual position errors after correction of geometric and joint compliance error (being used as ANN training output data)

The ANN is trained by the data that consists of the robot joint readings Q_2 (the ANN training input) and the residual position errors (the ANN training output as shown in Fig. 4.7). As a result, after non-geometric error compensation using the trained ANN, the average robot position accuracy is enhanced drastically up to 0.3264 [mm] from 4.0654 [mm] (before calibration). The more details are shown in Table 4.3. The residual position errors at measurement points are displayed in Fig. 4.8. A portion of Fig. 4.8 is magnified and is then displayed in Fig. 4.9 for better presentation.

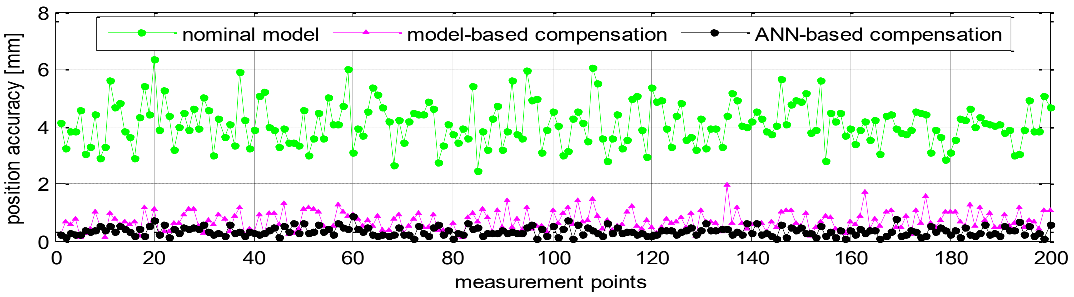


Fig. 4.8. Absolute position accuracy of the HH800 robot (before and after calibration)

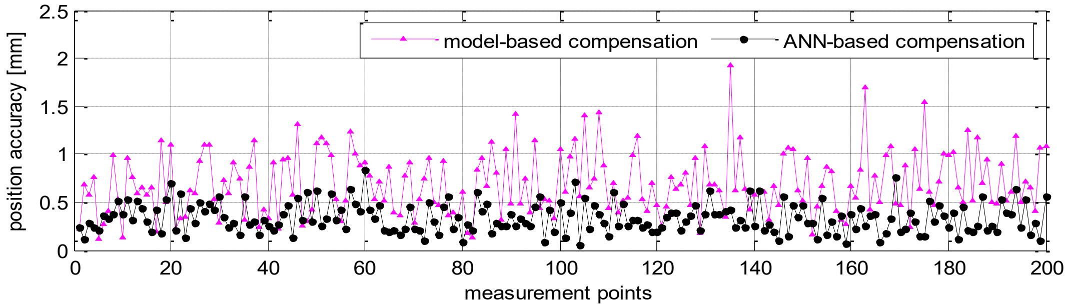


Fig. 4.9. Magnification of a section of Fig. 4.8

The robot position accuracy at the calibration measurement points is normally higher than the accuracy at other arbitrary measurement points. In this section, a validation of the robot performance after calibration is performed with the measurement set Q_3 (mentioned above).The validation results show that the robot average position accuracy after link geometry and joint compliance error correction is 0.7035 [mm] maximum position error is 2.1372 [mm]), and after non-geometric error compensation by the ANN is 0.3959 [mm] (maximum position error is 0.8928 [mm]).

The position errors at measurement points are also shown in Fig. 4.10. The position errors for the validation data set Q_3 (in Fig. 4.10) have the same order of magnitude as those of the calibration data set Q_2 (in Fig. 4.9). Therefore, the robot after calibration has the same level of position accuracy across its

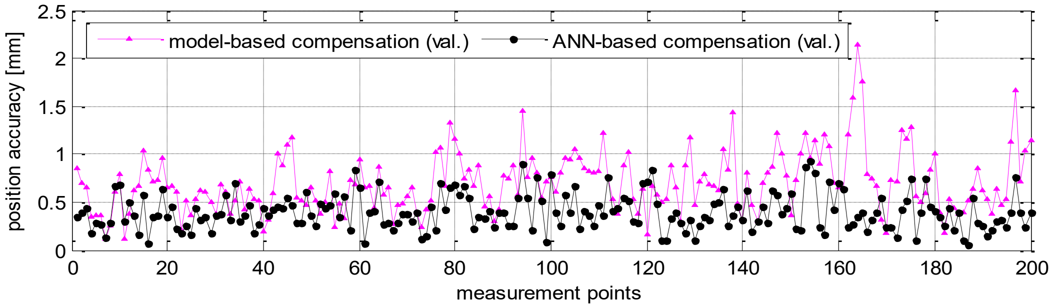


Fig. 4.10. Absolute position accuracy of HH800 robot (validation)

4.6. Conclusions

This chapter proposed a new calibration method for enhancing robot position accuracy. The method is a combination of a model-based calibration procedure and an ANN-based error compensation technique. The model-based calibration method has many advantages such as less computing time, fast convergence, and accurate knowledge of error sources. Some error sources (especially non-geometric errors) cannot, however, be determined and modeled correctly. Therefore, in order to further increase the robot position accuracy, the un-modeled error sources should be compensated for by using an ANN.

The experimental calibration was performed on the Hyundai HH800 robot. After calibration, the robot average accuracy is increased significantly to 0.3264 [mm] (from 4.0654 [mm] before calibration). The calibration results demonstrate the effectiveness and correctness of the proposed method. Additionally, the validation results for robot accuracy at arbitrary points show that the robot after calibration has the same level of position accuracy in its entire workspace.

Chapter 5: A ROBOTIC CALIBRATION METHOD USING A MODEL–BASED IDENTIFICATION TECHNIQUE AND AN INVASIVE WEED OPTIMIZATION NEURAL NETWORK COMPENSATOR *

This chapter proposed a robotic calibration algorithm for improving robot manipulator position precision. At first, the kinematic parameters as well as the compliance parameters of the robot can be identified together to improve its accuracy using the joint deflection model and the conventional kinematic model calibration technique. Then, an artificial neural network is constructed for further compensating the unmodeled errors. The invasive weed optimization is used to determine the parameters of the neural network. To show the advantages of the suggested technique, a HH800 robot is employed for the experimental study of the proposed algorithm. The improved position precision of the robot after the experiment firmly proves the practicability and positional precision of the proposed method over the other comparing algorithms.

**Le, Phu-Nguyen, and Hee-Jun Kang. "A Robotic Calibration Method Using a Model-Based Identification Technique and an Invasive Weed Optimization Neural Network Compensator." Applied Sciences 10.20 (2020): 7320.*

5.1 Introduction

Robot manipulators are widely used in industry to attain many duties such as welding, painting, pick and place task, etc. The construction of robot manipulators is characterized using their kinematic model parameters. However, experience has shown that industrial manipulators have much repeatability than accuracy[10]. Therefore, the utility of robot manipulators would be significantly enhanced if they were made to be as accurate as they are repeatable. The reason for the difference between accuracy and repeatability could be described as follow. Repeatability is defined as the ability of the manipulator to return to a pose that has been stored in joint space. Accuracy, however, is the ability of the robot to move to a pose defined in task space. To achieve a pose that is defined in task space, the robot controller must convert the task space definition of the pose into joint space. The individual joints are then moved so that the desired configuration is obtained. The conversion from task space to joint space is accomplished by using a mathematical model of the manipulator. This mathematical model relates the joint displacements to the end effector pose and vice versa. When the mathematical model used by the robot controller to describe the robot motion differs from the actual geometry of the manipulator, the joint space definition of a pose defined in task space will not be accurate.

In producing and assembly, many errors arise that could not be taken into accounts by the nominal geometric model. For that reason, there is a demand to create model-based robotic calibrations that depend on an error model that symbolizes the connection between the errors of geometric parameters and the end effector positioning errors. The kinematic calibration method has been widely researched by numerous studies [13, 16–18, 22–25, 29, 30, 68, 69] to describe the proper geometric model. Denavit, Hartenberg et al. proposed one of the most fundamental calibration methods that is widely used. The DH model based on homogenous transformation matrices. The model is a description of the kinematic relations between the links of a kinematic chain connected by 1 degree-of-freedom lower pair joints [23–25]. There are also other methods such as the zero-reference position model that is proposed by Gupta et al. has been widely used by many studies [13, 14, 70]. Others fundamental modeled based calibration method are the complete and parametrically continuous (CPC) model which relies on the singularity-free line representation by Zhuang et al. [29][30]. Park and Okamura [16] employed the product of exponentials (POE) error model to robot calibration. This model has been adopted by several researchers [17][18]. There have been some attempts to replace the least square estimation with the Kalman filter, particle filter, fuzzy theory [31, 71, 72]. However, the effectiveness of those attempts does not seem to be much.

Besides, the kinematic errors are not the only source of robot manipulator errors. Whitney et al. [1] used a PUMA 560 robot to determine that the most significant error sources for the robot were several nongeometric errors. Judd and Kasinski[64] studied an AID-900 robot and found that the geometric errors were responsible for approximately 95% of the measured error. Jean et al. [3] and Becque [4] reported that flexibility in joints and in links is responsible for 8- 10% of the total position and orientation errors. Therefore, more recently researchers devoted to identification of compliance errors as well as kinematic errors in industrial robots[7, 22, 64]. Jian Zhou and Hee-Jun Kang [22] proposed a method for simultaneous identifying the kinematic and compliance parameter of the robot. The model-based calibration has been widely applied for the reason of fast computing, and knowledge of error sources [20, 22, 73]. However, the accuracy of the model-based calibration method is dependent on the model accuracy. It is practically impossible to create a model that contains all the sources committing to the end effector errors. Some error sources such as gear backlash, temperature variation, and other errors that are difficult to model correctly and completely.

To archive further accuracy, non-geometric calibration has been widely studied for compensating for the sources of errors that could not be taken into account by geometric calibration [3, 6, 52, 53, 71, 74–78]. The conventional back-propagation neural network (BPNN)

[79] is widely adopted by researchers [80] for compensating the unmodeled errors to increasing the precision of the robot. In robotic calibration processing, BPNN is usually employed to construct the relationship between the end effector position and the corresponding joint angle configuration [39, 57, 81, 82]. However, the conventional BPNN has some drawbacks such as getting stuck in local minima and slowing convergence [83].To overcome these drawbacks, some heuristic algorithms have been used for training the network [84–88]. One among them is the Invasive Weed Optimization (IWO) algorithm.

In 2016, Mehrabian et al. presented the Invasive Weed Optimization (IWO) algorithm [89]. This method is motivated by the spreading of weeds to find a suitable place for expanding and breeding. The technique is characterized by fast convergence for global optimization. Based on the properties of IWO, this algorithm is employed for optimizing the parameters of a neural network to reduce its cons such as high dependence on input data. Furthermore, the IWO prevents the neural network from falling to local minima.

This work proposed a new calibration method that includes the model-based calibration technique and unmodeled calibration method. At first, the kinematic and compliance parameters are simultaneously determined by a model-based calibration method [22]. Then, to compensate some unmodeled errors that cannot be ignored such as friction, mechanical transmission error, thermal expansion. A suggested neural network optimizing by IWO is used to compensate the residual positional errors. It should be noted here, the proposed method is a combination of model-based and ANN methods which is used the IWO technique to determine the weight and bias. While most of ANN-based technique is applied after the kinematic calibration. This calibration method simultaneously calibrates both the kinematic errors and joint compliances. After the simultaneous calibration, the IWO-NN based compensation is accomplished for the un-modelled non-geometric errors. By using IWO neural network, the proposed calibration method seems to reach the global minima easily. Therefore, the IWO neural network can be said to have better convergence capability than the traditional backpropagation NN. Finally, a HH800 robot is employed for the experimental study of the proposed algorithm in comparing with 4 others calibration methods including conventional kinematic calibration method (KM), as well as the simultaneous identification of joint compliance and kinematic parameters method (SKCM), and the combination of NN compensator and SKCM method (SKCM-NN) . The advantages of the method had been shown, such as: the enhanced position accuracy of the manipulator after the calibration confirms the feasibility and more positional accuracy over the other calibration methods. Additionally, the adopted IWO neural network has better convergence capability than the back propagation neural network in this calibration process This advantage allows that the proposed method is more feasible in real offline programming environment.

5.2. Iwo-NN Errors Compensator Technique

In Equation (4.3), the position errors ΔP_{extra} are still large after applying the algorithm of simultaneous identification of joint compliance and kinematic parameters. This occurred because of some un-modeled sources such as: friction, thermal extension, mechanical transmitting errors which are hardly considered in geometric calibration. To decrease these unmodeled errors, a non-geometric compensator should be carried out for compensating. In this chapter, an IWO-NN errors compensator is taken to eliminate the residual positional errors. The method is fully described as follow.

First, the robot geometry error $\Delta\phi$ and joint compliance parameter C are identified together by Eq. (4.4) using the least square method. Total positional error vector is denoted by ΔP (3×1) :

$$\Delta P = P_m - P_c = \Delta P_{kin} + \Delta P_c + \Delta P_{extra} \tag{5.1}$$

whereP_m indicates the positions of the end effector by measuring, and P_c denotes the calculated position of it. $\Delta P_{kin}, \Delta P_c$ are the kinematic error and compliance error that are calculated by applying the method in Ref. [22]. The residual position error after the modeled based calibration process is ΔP_r. Here, the IWO-NN is hired to compensate the residual error ΔP_r.

$$\Delta P_r = P_m - P_{kin} - \Delta P_{kin} - \Delta P_c \tag{5.2}$$

The IWO-NN contains six inputs which is the joints configuration . There are 5 nodes in the hidden layer. The tag-sigmoid is selected as the activate function for the hidden layer as follow[90]:

$$z = \text{tansig(a)} = \frac{2}{1+e^{-2a}} \tag{5.3}$$

Three outputs of the NN has the linear activation function. The outputs of the NN is used for compensating the residual errors ΔP_r

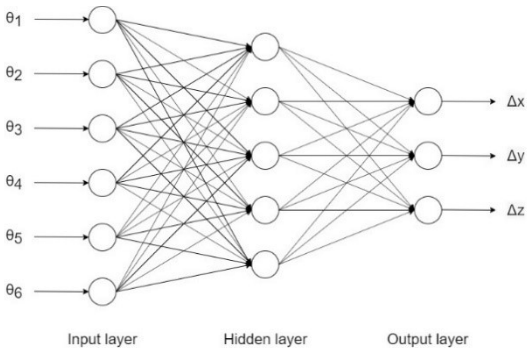


Fig. 5.1 The IWO-NN diagram

The error in the output layer:

$$e = \Delta P_r - P_{nn} \tag{5.4}$$

where P_{nn} are the output of the neural network.

The mean square error is calculated as follow:

$$\text{E} = \frac{1}{m}\sum_{k=1}^{m} e_k^2 \tag{5.5}$$

where m=3 represents the three dimensions of the end effector position. The IWO is a recently proposed population-based heuristic optimization algorithm that mimics the spreading of weeds to find a suitable place for expanding and breeding [91][92]. IWO has four main stages: i) initialization, ii) reproduction, iii) space distribution, and iv) ranking and selection.

i) Initialization: The population of solution is generated. The seeds are randomly distributed over all the space problem. The number of seeds is chosen.

ii) Reproduction: In this step, every seed reproduces the next generation. The number of every seed's heir depends on its fitness in the community. If the seed has a good fitness, it will produce more seeds than another seed that has a worst fitness in the society.

iii) Space Distribution: This is the progress that the offspring are randomly spread over all the searching space. The spreading process is made to distribute the offspring close to their parents' location. Therefore, the seeds with lower fitness in the population are deleted over the time. The distribution function is described as below:

$$\sigma_{iter} = \frac{(iter_{max} - iter)^n}{(iter_{max})^n}\left(\sigma_{initial} - \sigma_{final}\right) + \sigma_{final} \tag{5.6}$$

where σ_{iter} is the standard deviation (SD) at the present step, $\mathrm{iter}_{\mathrm{max}}$ is the maximum number of iterations before stopping the algorithm, $\sigma_{initial}$ is the initial value of the SD, σ_{final} is the final value of the SD, and n is the nonlinear modulation index.

iv) Ranking and Selection: when the maximum number of populations in the colony is hit, all the seeds are evaluated including new seeds and their producers. The unqualified fitness seeds will be eliminated .

In this work the IWO are introduced for optimizing the weights and biases of a NN. The optimizing process is described by the flowchart in the Fig. 5.2.

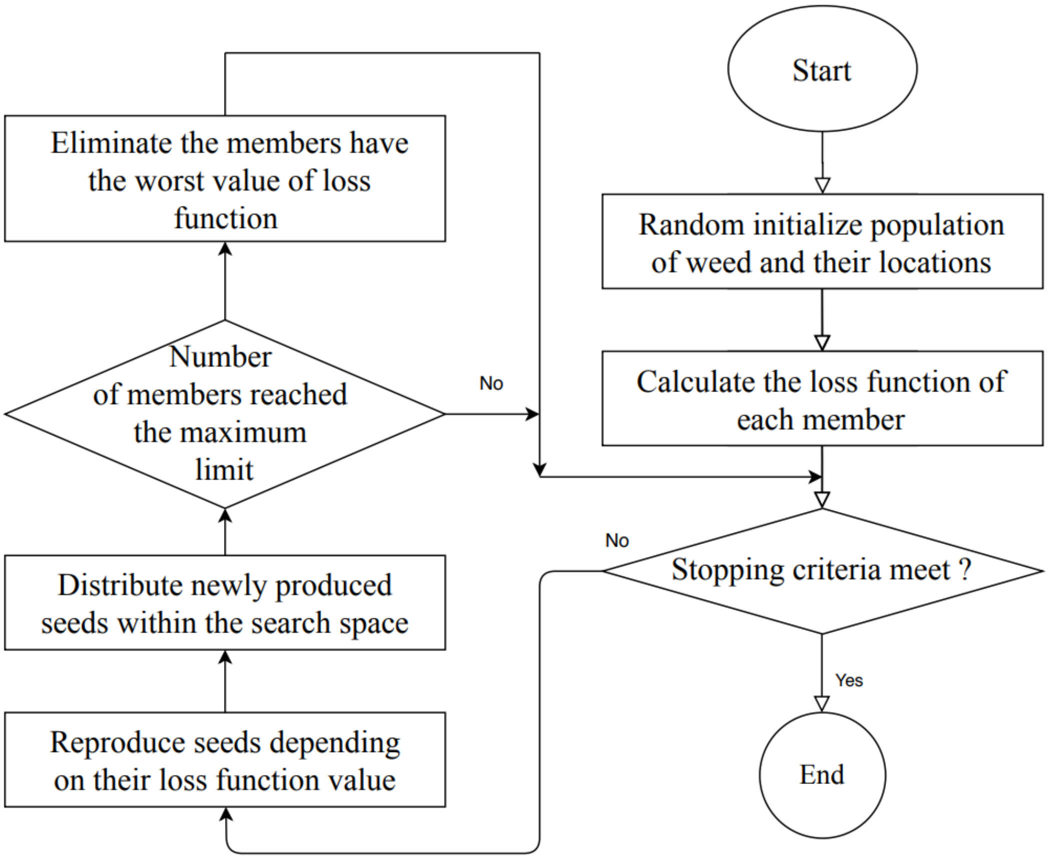

Start
Random initialize population of weed and their locations
Calculate the loss function of each member
Stopping criteria meet ?
No
Yes
End
Reproduce seeds depending on their loss function value
Distribute newly produced seeds within the search space
Number of members reached the maximum limit
No
Eliminate the members have the worst value of loss function

Fig. 5.2 Algorithm flow chart of an IWO-NN

Assume $X_i = \{x_{i_1,} x_{i_2,} \dots , x_{i_n,}\}$ is the i$_{th}$ member that contains weights and biases of the NN. An initial population including 10 members is randomly spread over the searching space. The maximum population size is 50. The minimum number of seeds is $S_{min} = 0$ and the maximum number of seed is $S_{max} = 5$. The variance reduction exponent in Eq.(5.6). is $n = 2$. The initial value of the SD is$\sigma_{initial} = 0.05$, while the final value of the SD is $\sigma_{final} = 0.0005$. The dimension of member X_i of the population is $i = 53$ in according to the number of the weights and biases of the fully connected NN, which has 6 inputs, 3 outputs, and 5 hidden nodes.

The loss function is used for evaluating the fitness of member X_i of the population:

$$LF_i = l_1 * \text{mean}(E_i) + l_2 * \max(E_i) \tag{5.7}$$

Here, E_i is the matrix of mean square errors in Eq. (5.5) of the $\mathrm{i_{th}}$ set of NN respectively to 50 inputs and outputs of the calibration data, $l_1 = 1, l_2 = 0.7$ are the weights that represent the contribution of mean, and max function in forming the cost function of each solution i.

By using Eq. (5.7), the number of seeds for member X_i of the population can be calculated:

$$\mathrm{S} = floor(S_{min} + (S_{max} - S_{min}) * r_i) \tag{5.8}$$

where $floor$ is the function that rounds to the nearest integer less than or equal to the input number. r_i is the ratio calculated by the following equation:

$$r_i = \frac{LF_i - \max{(LF)}}{\min(LF) - \max{(LF)}} \tag{5.9}$$

The seed S_j of member X_i can be calculated by

$$S_j = X_i + \sigma_{iter} * \mathrm{K} \tag{5.10}$$

using the value of σ_{iter} in Eq. (5.6). K is a matrix that has the same dimension as X_i. Every element of K is a random number between 0 and 1. For speeding the process, the values of elements of S_j are bounded. The lower and upper bounds are -1.7 and 1.7, respectively.

After this process, all the seed S_j of member X_i are merged into the population and considered members of the population. If the number of members overcomes the maximum population size, the worst members having the highest valued loss function are eliminated. Then the processing is repeated again until it reaches the maximum iteration or one of the members archives the desired loss function value.

The Figure 5.3 shows the flowchart of the proposed method.

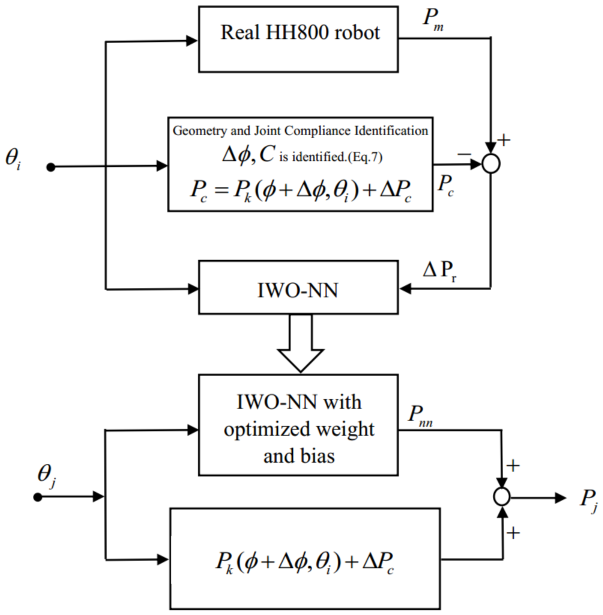

Real HH800 robot
P_m
Geometry and Joint Compliance Identification
$\Delta\phi, C$ is identified.(Eq.7)
$P_c = P_k(\phi+\Delta\phi,\theta_i)+\Delta P_c$
θ_i
P_c
−
+
IWO-NN
ΔP_r
IWO-NN with optimized weight and bias
P_{nn}
θ_j
+
+
P_j
$P_k(\phi+\Delta\phi,\theta_i)+\Delta P_c$

Fig. 5.3. Flow chart of the proposed method

5.3. Experiment and Validation Results.

A HH800 robot is employed to examine the proposed method to show the effectiveness and practicability of it. To show the advantages of the proposed method (SKCM-IWONN) in improving the robot position accuracy, four calibration methods are carried and compared in both calibration and validation process. The conventional kinematic calibration method (KM)[3], as well as the simultaneous identification of joint compliance and kinematic parameters method (SKCM)[12], and the combination of NN compensator and SKCM method (SKCM-NN) [25] are also used in this experiment calibration.

5.3.1 Experimental Calibration Results

In this experiment, a HH800 robot carrying a heavy load (745 kg) and an API laser tracker (accuracy of 0.01 mm/m, repeatability of ±0.006 mm/m) is used to perform the experimental process. The laser reflector is attached to the manipulator end-effector. To collect the measurement data, the end effector of robot is moved evenly overall the working space. The 3D position of the end effector and the related joints configuration are measured and recorded to computer simultaneously. To identify the kinematic parameters and compliance parameters of the HH 800 robot, 40 random robot configurations (set Q1) is used to calibrate the robot using the SKCM. The results of geometric parameters and stiffness parameters of HH800 robot are demonstrated in Table 5.1 and 5.2. In order to determine a neural network that has more ability to compensate over all the workspace, a set (Q2) that is includes 50 random robot configurations completely different from the set Q1 is used to find out the weights and bias of the NN. After that, the weight and bias of this network are generated by the IWO method .

In this calibration progress, the HH800 robot is used to apply 4 different calibration algorithms. Tables 5.3 and Fig. 5.4 demonstrate the results of these calibration methods. By using the IWO-NN SKCM, the mean of errors is reduced from 0.5961 mm to 0.3450 mm as much as 42.12% in comparing to KM method. This mean of errors is also decreased as much as 31.52% in comparing to the SKCM method (0.5038 mm). In comparing to the SKCM-NN method, the mean of errors result is declined 15.92% (0.4103 mm). The IWO-NN is better than SKCM-NN in reducing the maximum of absolute position errors (41.88%). The proposed method also generates the lowest maximum position error (0.6374 mm) and standard deviation (0.1624 mm).

Table 5.1. Stiffness identification.(Inverse of joint compliance)

	K2	K3	K4	K5
Stiffness (Nm/rad)	6.159*10^7	4.388*10^6	3.151*10^6	2.220*10^6

Table 5.2. Identified D-H parameters of Hyundai robot HH800.

D-H parameters of the main open chain						
i	αi-1(deg)	ai-1(m)	βi-1(deg)	bi-1(m)	di(deg)	θi(deg)
1	0.8752	0.0003	0.006	0.0001	0.0976	0.3468
2	89.9412	0.5157	-	-	0(X)	-0.8836
3	0.0133	1.5998	0.001	-	-0.0014	-1.2385

4	90.1172	0.3545	-	-	1.8862	3.3033
5	-90.038	0.0002	-	-	4.087e-05	2.5786
6	90.0371	0.0003	-	-	0.445(X)	0(X)
T	-	-0.4511	-	0.0111	0.9279	-
Link lengths of the closed loop						
L5(m)	0.7996	L4(m)	1.601	L3(m)	0.8(X)	

("-": unavailable, "X": un-identifiable)

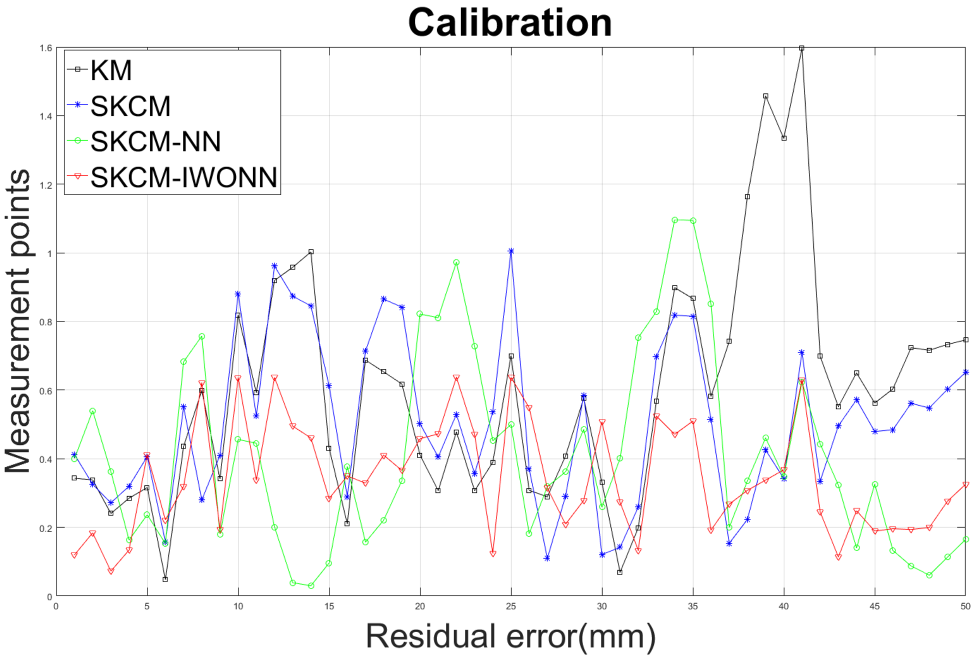


Fig. 5.4 Residual positional errors after calibration

Table 5.3. Absolute position accuracy of the HH800 robot (calibration).

	Mean (mm)	Maximum (mm)	Std. (mm)
Before calibration	4.5969	6.6664	0.8408
KC	0.5961	1.5967	0.3299
SKC	0.5038	1.005	0.2346
NN-SKC	0.4103	1.0967	0.2814

SKCM-IWONN	0.3450	0.6374	0.1624

5.3.2 Experimental Validation Results

The validation process is examined to illustrate general capability of the suggested calibration technique over the entire robot workspace. In this progress, the set Q3 includes 50 robot configurations of the manipulator that is not used in the calibration process is taken to examine the 4 different methods. The results of validation process are demonstrated in the Figure 5.5 and Table 5.4.

By using the proposed method, the mean of errors is reduced from 0.5981 mm to 0.4662 mm as much as 22.07% better in comparing to KM method. This mean of errors is also decreased as much as 26.72% in comparing to the SKCM method (0.5458 mm), and 10.14% decline from the mean of errors by using the SKCM-NN method (0.5187 mm). The proposed method also generates the lowest maximum position error (0.6538 mm) and standard deviation (0.1333 mm).

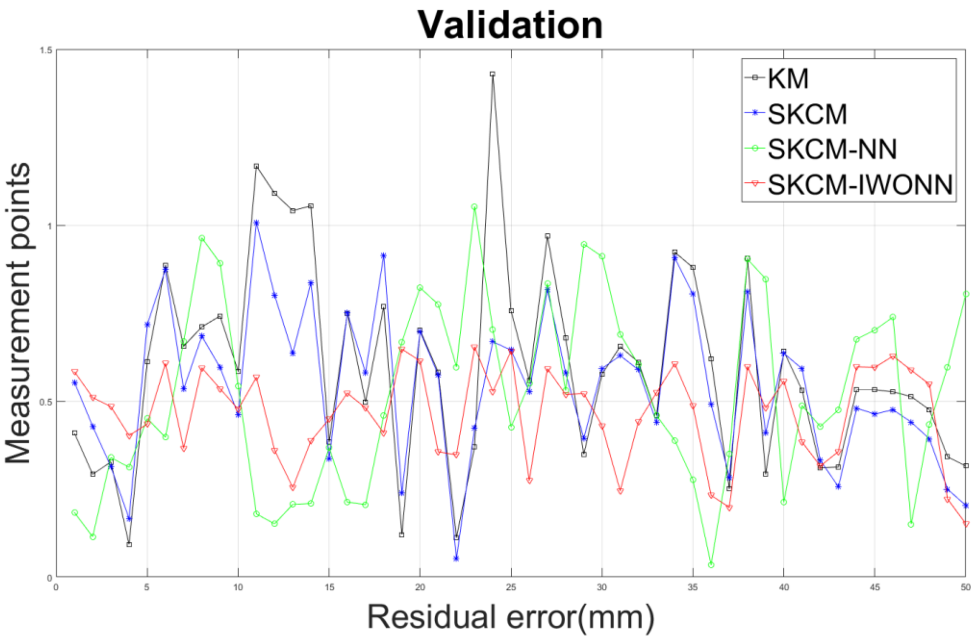


Fig. 5.5. Residual positional errors after validation

Table. 5.4. Absolute position accuracy of the HH800 robot (Validation).

	Mean (mm)	Maximum (mm)	Std. (mm)
Before calibration	4.4266	6.2749	0.6772
KC	0.5981	1.4293	0.285
SKC	0.5458	1.0078	0.2142
NN-SKC	0.5187	1.0532	0.2642
SKCM-IWONN	0.4662	0.6538	0.1333

5.3.3 Advantages of The Iwo-NN Compensator

In this work, the robot configurations are divided into several data sets that are compensated by the NN and the proposed IWO-NN. From this experimental application, the NN appear to be usually fallen into the local minima and a reinstalling of random weights and bias is required to reach the global minima. On the other hand, the IWO-NN seems to archive the global minima quite efficiently. For that reason, the proposed IWO-NN can be said to have better convergence capability. Moreover, accuracy consistency of experimental calibration(0.345 mm) and validation(0.4662 mm) confirms the abilities of the suggested calibration method.

5.4. Conclusions

The study suggested a new robot calibration method combining the geometric and the non-geometric calibration technique. The kinematic and joint compliance parameters are simultaneously calibrated by using the SKCM method. A NN is adopted to eliminate the unmodeled position errors. The IWO is used to determine the weights and bias of the NN to overcome some drawbacks of it such as heavy relying on the input and easy falling to local minima. Inheriting advantages from the model-based calibration method as well as the non-geometric calibration method, the proposed technique not only supplies the knowledge of the position errors but also has abilities to calibrate the unmodeled position errors.

In order to show the demonstrate the effectiveness of the proposed method, a HH800 robot is employed for the experimental study of the proposed algorithm. The improved position precision in the experiment process shows the effectiveness and advantages of the method over the other comparing calibration methods.

Chapter 6: ROBOT MANIPULATOR CALIBRATION USING A MODEL BASED IDENTIFICATION TECHNIQUE AND A NEURAL NETWORK WITH THE TEACHING LEARNING-BASED OPTIMIZATION*

This chapter proposes a new calibration method for enhancing robot positional accuracy of the industrial manipulators. By combining the joint deflection model with the conventional kinematic model of a manipulator, the geometric errors and joint deflection errors can be considered together to increase its positional accuracy. Then, a neural network is designed to additionally compensate the unmodeled errors, specially, non-geometric errors. The teaching-learning-based optimization method is employed to optimize weights and bias of the neural network. In order to demonstrate the effectiveness of the proposed method, real experimental studies are carried out on HH 800 manipulator. The enhanced position accuracy of the manipulator after the calibration confirms the feasibility and more positional accuracy over the other calibration methods.

**Le, Phu-Nguyen, and Hee-Jun Kang. "Robot Manipulator Calibration Using a Model Based Identification Technique and a Neural Network With the Teaching Learning-Based Optimization." IEEE Access 8 (2020): 105447-105454.*

6.1. Introduction

Robot manipulators are broadly employed in industry to attain many duties such as welding, painting, pick and place task, etc. The construction of robot manipulators is characterized using their kinematic model parameters. However, in producing and assembly, many errors arise that could not be taken into accounts by the nominal geometric model. For that reason, there is a demand to create model-based robotic calibrations that depend on an error model that symbolizes the connection between the errors of geometric parameters and the arm's tip positioning errors. There are many researchers work on this approach[68, 69]. Numerous studies have been carried out to investigate the suitable geometric model for the robot calibration. Denavit, Hartenberg at el. suggested the DH model that is one of the most fundamental geometric calibration methods. The model has been extensively adopted by researchers[23–25]. Gupta proposed another model using the zero-reference position model[13]. This model has been used by some authors [14, 70]. Besides, researchers also proposed many others approaches. Zhuang et al. proposed a complete model for completely expressing the geometric and movement of the robotic manipulator of the robot (CPC) by using the singularity-free line representation[29][30]. Another approach that has been widely examined is the product of exponentials model (POE). The method based on skew theory that allows a global description of robotic motion[16–19]. There have been some attempts to replace the least square estimation with the Kalman filter, particle filter.[31, 71, 93–96]. However, the effectiveness of those attempt does not seem to be much. The model-based access has many benefits. It is potential to find out the model parameters precisely in considering that error sources are suitability described by the error model. By using this access, geometric errors can be formed accurately. For instance, the kinematic model and the joint stiffness of the robot are modeled efficiently by many researchers [20, 22, 73]. Geometric error calibration methods have abundant abilities: reducing the calibration time, easy to covert, contributing exactly the information of error's sources. However, they also have some drawbacks.

Practically, it is difficult to create kinematic identification models that consider all the causes engendering the end effecter error.

The kinematic calibration has been researched for decades and gradually mature. For further accuracy, non-geometric calibration methods have been recently developed, [3, 6, 78, 32, 39, 52, 53, 74–77]. Using the non-geometric error approach, some authors employ optimization methods to optimize the robot parameters [3, 32, 52, 53, 74–77]. Some studies take the radial basis function [6][78] or artificial neural network (NN) [39] to generate a connection between the arm's tip position errors and the matching joint angle configurations [39, 57, 81, 82]. Among them, the back propagation neural network (BPNN) [79][36] has been widely used due to its capabilities such as learning, adaptation, and approximating any nonlinear function with arbitrary precision [80]. However, the conventional BPNN has a problem in which the performance relies heavily on the input data and hardly finding the overall minima[83]. Numerous methods are suggested to overcome these drawbacks of the BPNN. Heuristic optimization methods are carried on optimizing the structure of the NN. Some of them are particle swam optimization, genetic algorithm, teaching learning optimization, firefly optimization[84–87].

In 2011, Rao et al. proposed the teaching-learning-based optimization (TLBO) [97]. Overall, the TLBO method replicates the teaching and learning of human. It has two main phases: (1) select the nominate with the best performance to be the teacher of the class to activate the teacher learning step (2). The other nominates are students. They learn from each other and from the teacher. In every interaction, the best solution will become the teacher and the students will change itself base on the teacher and the other students. The TLBO approach is described as a feasible and strong method. It supplies the fast convergence time and easy to get the overall minima.

In this study, the kinematic parameters and joint compliance parameters of the robot are simultaneously identified first by our model-based calibration[22]. However, there are still some unmodeled errors such as friction, thermal variety, gear errors. Then, a proposed neural network with TLBO is applied to compensate the remain errors. For better compensation, the teaching-learning-based-optimization(TLBO) is employed to optimize the weight and bias of the neural network. Finally, the proposed algorithm is implemented for the experimental studies on a HH800 robot. The kinematic and structure of this robot is clearly described in the Section 6.2 of the Chapter 6. The enhanced position accuracy of the manipulator after the calibration confirms the feasibility and more positional accuracy over the other calibration methods. The proposed method is a combination of model-based and ANN methods which is used the TLBO technique to determine the weight and bias. By using TLBO neural network, the proposed calibration method seems to quite easily reach the global minima. Therefore, the TLBO neural network can be said to have better convergence capability than the traditional backpropagation NN. Furthermore, most of ANN-based technique is applied after the kinematic calibration. Our calibration method simultaneously calibrates both the kinematic errors and joint compliances. The model-based compensation part is the extension to the closed-chain manipulator of our previous work for simple serial robot[22]. After the simultaneous calibration, the TLBO-NN based compensation is accomplished for the un-modelled non-geometric errors. The combination of simultaneous hybrid calibration with a TLBO based neural network is shown to lead to the better capability of reducing the errors of the robot than other calibration methods . Furthermore, the proposed calibration method utilizes the advantages of the TLBO neural network over the conventional back propagation neural network such as better error reducing capability and better global minimum reaching capability.

6.2. Simultaneous Identification Joint Compliance and Kinematic Parameters

Accepting the manipulator joint can be demonstrated as a linear torsional spring, the joint stiffness value of the i_{th} joint is symbolized by a constant value k_i. Considering the manipulator joint is much bender than the relative link . Hence, the most elastic errors are result of the elasticity of manipulator joints under the effect of the links' weight themselves and external payload. The shifting of the i_{th} joint can be expressed by the effective torques [23]:

$$\Delta\theta_{c_i} = \frac{\tau_i}{k_i} = \tau_i c_i \tag{6.1}$$

The end effector position errors due to the elastic of the joints can be calculated as:

$$\Delta P_c = J_\theta \Delta\theta_c = (J_\theta \tau)C \tag{6.2}$$

where $C = [\, c_1 \quad c_2 \; \dots \; c_n]^T$ is the joint compliance vector, $\Delta\theta_c = [\, \Delta\theta_{c_1} \quad \Delta\theta_{c_2} \; \dots \Delta\theta_{c_n}]^T$ is the joint deflection vector, and $\tau = diag(\tau_1, \tau_2, \dots, \tau_n)$ is the effective torque in the robot joints at the balance position. J_θ the sub-matrices computed by the method presented in [34]. The robot's tip position vector can be formed:

$$P_{real} = P_{kin} + \Delta P_{kin} + \Delta P_c + \Delta P_{extra} \tag{6.3}$$

where P_{kin} is the result of forward kinematics based on the current kinematic parameters. ΔP_{kin}, ΔP_c, ΔP_{extra} are the position errors due to the kinematic parameter error, joint elastic, and the residual errors due to the unmodeled sources, respectively. The combined error model can be showed as:

$$\begin{aligned} \Delta X &= \Delta P_{kin} + \Delta P_c \\ &= J\Delta\phi + (J_\theta \tau)C \\ &= [J \quad J_\theta \tau] \begin{bmatrix} \Delta\phi \\ C \end{bmatrix} \\ &= J_\Phi \Delta\Phi \end{aligned} \tag{6.4}$$

where ΔX is a (3 × 1) vector of three position errors of the robot end-effector. J is a (3 × p) matrix that relates the column vectors ΔX and ΔP_{kin}.(p = 27 is the total number of kinematic parameters). By using the least-square method, the kinematic and joint compliance parameter can be computed at the same time.

6.3. Error Compensation with TLBO based Neural network

The calibration process that is shown in the previous chapter is effective. However, it does not have any capability to decrease the position errors ΔP_{extra} that cannot be negligible. This happens due to unmodeled errors such as friction, thermal variety, gear errors. To lower the residual errors, the additional error compensation technique should be devised such as a neural network, a special function network. In here, a TLBO based neural network is employed to reduce the residual errors. The TLBO algorithm is used to optimize the weight and bias of the neural network and shows the better performance over the conventional back propagation algorithm in this calibration process. The details of this are shown as follows. First, the robot geometry and joint compliance are simultaneously identified by the Eq. (6.4). By using the least square method, the geometric errors $\Delta\phi$ and the compliance parameters C are identified. ΔP is a (3 × 1) total position error vector, expressed as:

$$\Delta P = \Delta P_{kin} + \Delta P_c + \Delta P_{extra} \tag{6.5}$$

$\Delta P_{kin}, \Delta P_c$ is calculated by compensation for link geometry and joint compliance errors. The residual position error is ΔP_r .

$$\Delta P_r = P_m - P_{kin} - \Delta P_{kin} - \Delta P_c \tag{6.6}$$

In this study, the NN optimized by TLBO is used to compensate this residual error. It concludes six inputs representing the joint configurations $\theta_n = [\,\Delta\theta_1,\ \ \Delta\theta_2\,,\ \ldots, \Delta\theta_6]$, three outputs representing the residual position errors after the robot geometric and joint angle deflection adjustment. There are 5 nodes in the hidden layer.

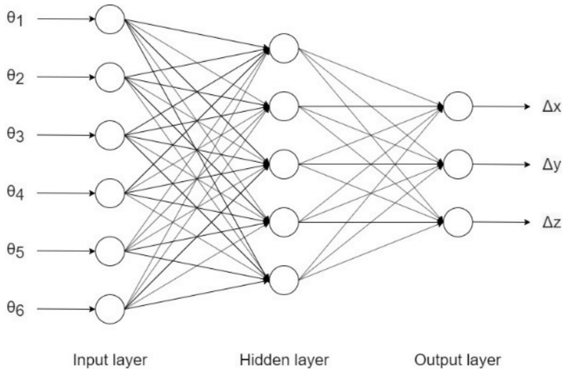


Fig. 6.1. Structure of the TLBO-BPNN

When an input pattern p is applied to the NN, the output of each unit j is described by:

$$\mathrm{o_{pj}} = \sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{f(w_{ij}x + b_j)} \tag{6.7}$$

where $\mathrm{o_{pj}}$ is the output of unit j as a result of the application of input x, $\mathrm{w_{ij}}$ is the weight of the node, and $\mathrm{b_j}$ is the bias of the unit j. The function f is the activation function, such as a sigmoid, tan-sigmoid, or linear function.

The error in the output layer is given as follows:

$$e = \Delta P_r - P_{nn} \tag{6.8}$$

where ΔP_r and P_{nn} are the desired and actual output of the neural network. The total mean square error in the output layer is given as follows:

$$E = \frac{1}{m}\sum_{k=1}^{m} e_k^2 \tag{6.9}$$

where m=3 is the number of nodes in the output layer.

The TLBO is adopted in the error compensation algorithm for the better optimized weights and biases of the neural network. The TLBO method [87][97][98] is a recently reported heuristic optimizing method that replicates the teaching and learning of human and is said to be better performance than the conventional back propagation method. The TLBO has two main phases: the teacher phase and the learner phase. In the teacher phase, the candidate with the best performance is selected to be the teacher of the class to activate the teacher learning step. The teacher shares its knowledge to all the learner and increase their performance. In the student phase, the students learn from each other and from the teacher. In every interaction, the best solution will become the teacher and the students will change itself based on the teacher and the other students. The details of this method are fully described by Rao et al. [97]. In this study, the TLBO is applied to optimize the weight and bias of the NN. The process is briefly described as following:

6.3.1Teacher Phase

In the teacher phase, the candidate with the best performance is selected to be the teacher applying the mean squared error cost function in Eq. (6.9). Assume $X_i = [\, x_{i_1},\ \ x_{i_1}, \dots, x_{i_1}]$ is the position of the i-th learner that represents the set of weights and biases of the NN. The mean position of the current class is called X_{mean} and the best position of the current class is noted as $X_{teacher}$. The teacher phase is expressed as the following equation:

$$X_{new,i} = X_{old,i} + \text{rand}(.) * (\, X_{teacher} - \text{TF} * \ X_{mean}) \qquad (6.10)$$

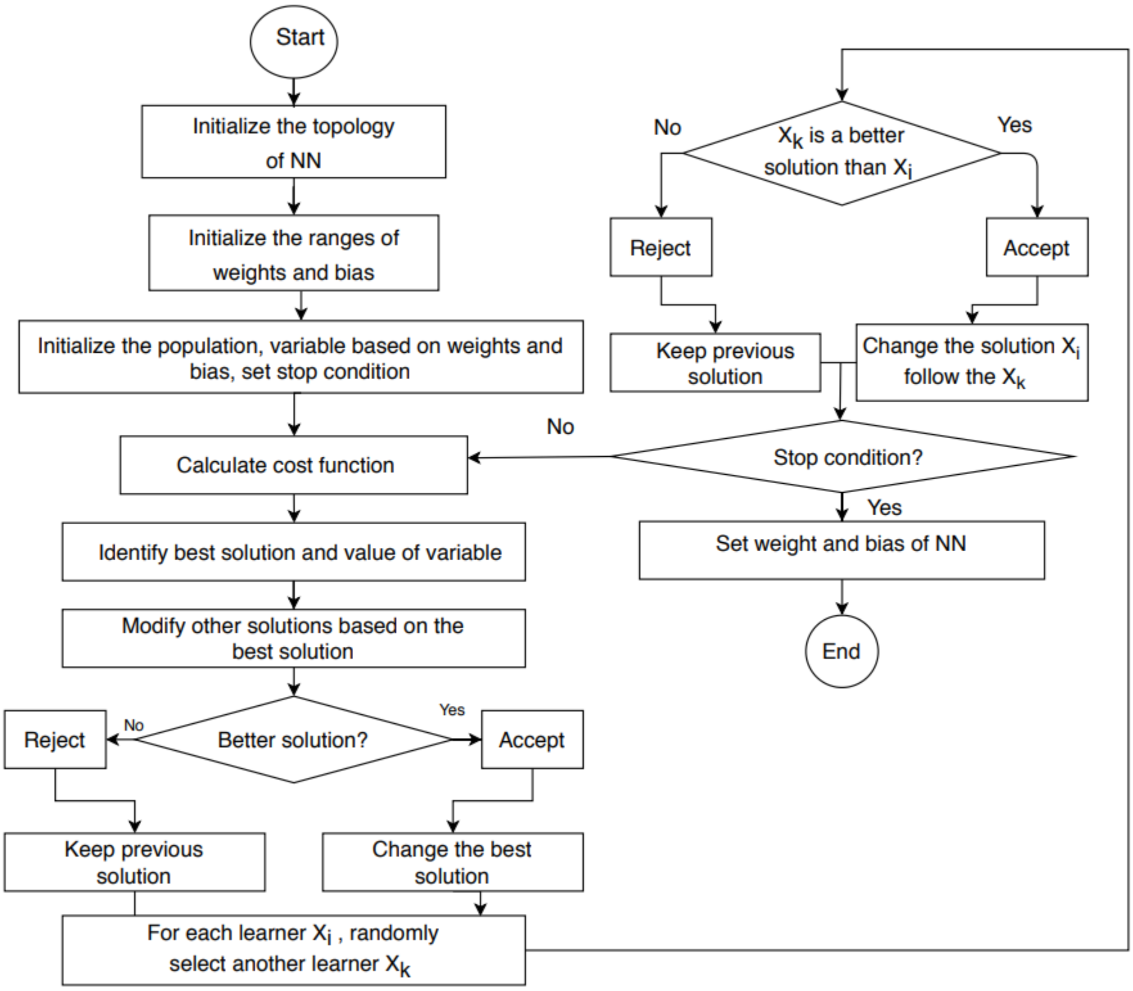
where $X_{new,i}$ and $X_{old,i}$ are the new and old positions of the *i*th learner. $X_{teacher}$ is the position of the current teacher and $\text{rand}(.)$ is a random number within the range [0, 1]. TF is the teaching factor. It is set to 1 or 2 randomly. All the learners should be re-evaluated in this phase. If $X_{new,i}$ has a higher performance than $X_{old,i}$ then $X_{new,i}$ will be chosen and will replace $X_{old,i}$, otherwise $X_{old,i}$ is not changed.

6.3.2 Learner Phase

For each *ith* learner in this phase, another learner *kth* (i ≠ k) is randomly selected from the class. The learning process is shown by the following equation:

$$X_{new,i} = \begin{cases} X_{old,i} + \text{rand}(.) * \left(X_{old,i} - X_{old,k} \right) * \\ X_{old,i} + \text{rand}(.) * \left(X_{old,k} - X_{old,i} \right) ** \end{cases} \tag{6.11}$$

$$\begin{aligned} * &\quad : If\ f\left(X_{old,i}\right) < f\left(X_{old,k}\right) \\ ** &\quad : otherwise \end{aligned}$$



Start
Initialize the topology of NN
Initialize the ranges of weights and bias
Initialize the population, variable based on weights and bias, set stop condition
Calculate cost function
Identify best solution and value of variable
Modify other solutions based on the best solution
Better solution?
No
Yes
Reject
Accept
Keep previous solution
Change the best solution
For each learner X$_i$, randomly select another learner X$_k$
X$_k$ is a better solution than X$_i$
No
Yes
Reject
Accept
Keep previous solution
Change the solution X$_i$ follow the X$_k$
Stop condition?
No
Yes
Set weight and bias of NN
End

Fig. 6.2. Algorithm flow chart of the TLBO to optimize weights and bias of the NN

If $X_{new,i}$ has a higher performance than $X_{old,i}$ then $X_{new,i}$will be selected and $X_{old,i}$ will be get rid of, otherwise $X_{old,i}$ is not changed.

In the following step, the stop conditions are checked. If they are satisfied, then the process is stopped. The weights and biases of the NN are set following the $X_{teacher}$. The process of optimizing the NN using the TLBO is shown in Fig. 6.2

Overall, the proposed method (SKCM-TLBONN) is shown by the flowchart in the Fig. 6.3:

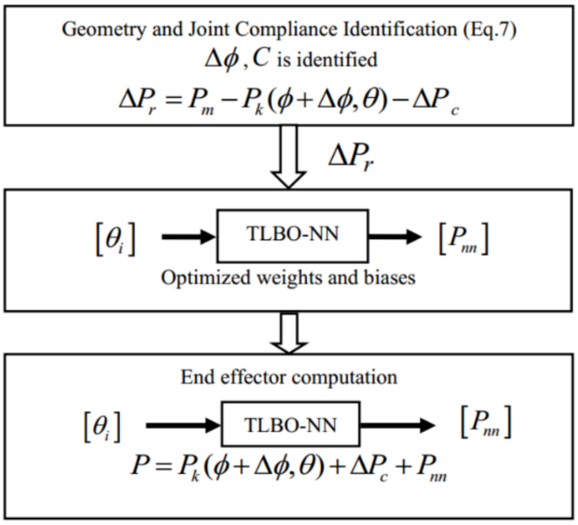

Geometry and Joint Compliance Identification (Eq.7)
$\Delta\phi, C$ is identified
$\Delta P_r = P_m - P_k(\phi+\Delta\phi,\theta) - \Delta P_c$
ΔP_r
$[\theta_i]$
TLBO-NN
$[P_{nn}]$
Optimized weights and biases
End effector computation
$[\theta_i]$
TLBO-NN
$[P_{nn}]$
$P = P_k(\phi+\Delta\phi,\theta) + \Delta P_c + P_{nn}$

Fig. 6.3. Flow chart of the SKCM-TLBONN algorithm

6.4. Experiment and Validation Results

To clarify the feasibility and efficiency of the proposed method, the proposed calibration method is employed to calibrate a HH800 robot. In addition to verify the effectiveness of the method, results of the calibration process are validated by another set of configurations that is not used for calibration working. Moreover, the proposed technique is compared with other calibration methods to show its capability. The conventional kinematic calibration method (KM)[23], the simultaneous identification of joint compliance and kinematic parameters method (SKCM)[22], and the combination of NN compensator and SKCM method (Chapter 2) (SKCM-NN) [39] are used to compare with the proposed method (SKCM-TLBONN) in both calibration and validation process.

6.4.1 Experimental Calibration Results

In the calibration process, the HH800 robot with a heavy load (745 kg), an API laser tracker and an accompanying laser reflector are used to perform the calibration process. The API laser tracker has the accuracy of 0.01 mm/m, repeatability of ±0.006 mm/m. The reflector is fixed at a particular location of the robot end-effector. In order to acquire suitable measurement data for robot parameter identification and weights and bias of the NN determining, the robot moves its end-effector to positions that entirely cover the workspace. The 3D coordinates of the end effector are measured by the Laser Tracker and saved in a computer. At the same time, the associated robot joint readings are also recorded. These measurements will be randomly grouped in three sets . A set of 40 robot configurations (Q1) is used in parameter identification.

By utilizing the SKCM method, the kinematic and joint compliance parameters are identified. These parameters are presented in Table 6.1 and Table 6.2 including 4 joint compliance parameters and 29 kinematic parameters. Another set (Q2) of 50 robot configurations is also randomly selected to determine the weights and biases of the neural network that has 5 hidden nodes, 6 inputs and 3 outputs. The reason why we use the different set (Q2) from the set(Q1) is that the neural network compensator has more general error compensation capability over the entire robot workspace. Then, the TLBO is employed to generate the optimized weights and biases of the neural network. The system is arranged as shown in Fig. 3.3.

The calibration processes of the robot are performed with the 4 different calibration methods such as KM[23], SKCM [22], and SKCM-NN [39]and the proposed SKCM-TLBONN. Their results are shown in the Table 6.3 and Fig. 6.4 show that the proposed method achieve the best performance over 4 calibration methods. It shows that the position errors generated by the proposed method is the lowest and better converging in compare with the other methods. The mean of position errors generated by the proposed method is more precise by 56.53% than the errors by KM method (from 0.9076 mm to 0.3945 mm), by 52.87% than the errors by SKCM method(from 0.8370 mm to 0.3945 mm), and by 15.16% than the errors by SKCM-NN method (from 0.4650 mm to 0.3945 mm). The proposed technique also has the lowest maximum position error as well as the best standard deviation.

Table 6.1. Stiffness Identification by SKCM method

	K2	K3	K4	K5
Stiffness	6.159*10^7	4.388*10^6	3.151*10^6	2.220*10^6

Table 6.2. Identifies D-H Parameters of The Hyundai Robot HH800("-": unavailable, "X": un-identifiable)

DH parameters of the main open chain						
i	α_{i-1}	a_{i-1}	β_{i-1}	b_{i-1}	d_i	θ_i
1	0.8752	0.0003	0.006	0.0001	0.0976	0.3468
2	89.9412	0.5157	-	-	0(X)	-0.8836
3	0.0133	1.5998	0.001	-	-0.0014	-1.2385
4	90.1172	0.3545	-	-	1.8862	3.3033
5	-90.038	0.0002	-	-	4.087e-05	2.5786
6	90.0371	0.0003	-	-	0.445(X)	0(X)
T	-	-0.4511	-	0.0111	0.9279	-
Link lengths of the closed loop						
L_5	0.7996	L_4	1.601	L_3	0.8 (×)	

Table 6.3. Absolute position accuracy of the HH800 robot (calibration) using SKCM-TLBONN .

	Mean (mm)	Max. (mm)	Std. (mm)
Nominal robot model	4.0654	6.3291	0.8803
KM	0.9076	3.7486	0.7671
SKCM	0.8370	2.7578	0.5461
NN- SKCM	0.4650	1.0123	0.1967
After proposed technique	0.3945	0.6544	0.1717

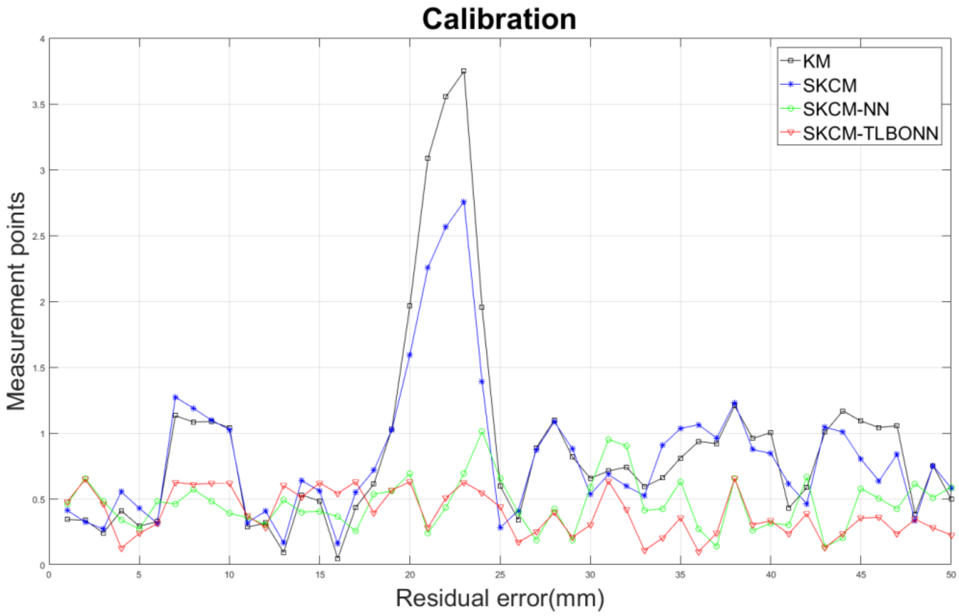


Fig. 6.4. Absolute position error of the HH800 robot after calibration using SKCM-TLBONN

6.4.2 Experimental Validation Results

From Table 6.3 and Fig. 6.4, the proposed method shows its error compensation capability with the positions used in this calibration process. In order to show the general capability over the entire robot workspace, it should be validated with the other data set. The other set of 50 robot configurations (Q3) is randomly selected and the calibration processes are carried out for 4 different methods.

The calibration results with Q3 data set are shown in Table 6.4 and Fig. 6.5. The Fig. 6.5 shows the residual error of each poses using 4 methods in the validation process. It is clearly to see from the figure that the proposed method is the best over both the positions used in the calibration process and the general position in overall workspace.

The mean of position errors generated by the proposed method is more precise by 56.03% than the errors by KM method (from 0.9227 mm to 0.4057 mm), by 49.85% than the errors by SKCM method (from 0.8009 mm to 0.4057 mm), and by 13.90% than the errors by SKCM-NN method (from 0.4712 mm to 0.4057 mm). The proposed technique also has the lowest maximum position error as well as the best standard deviation.

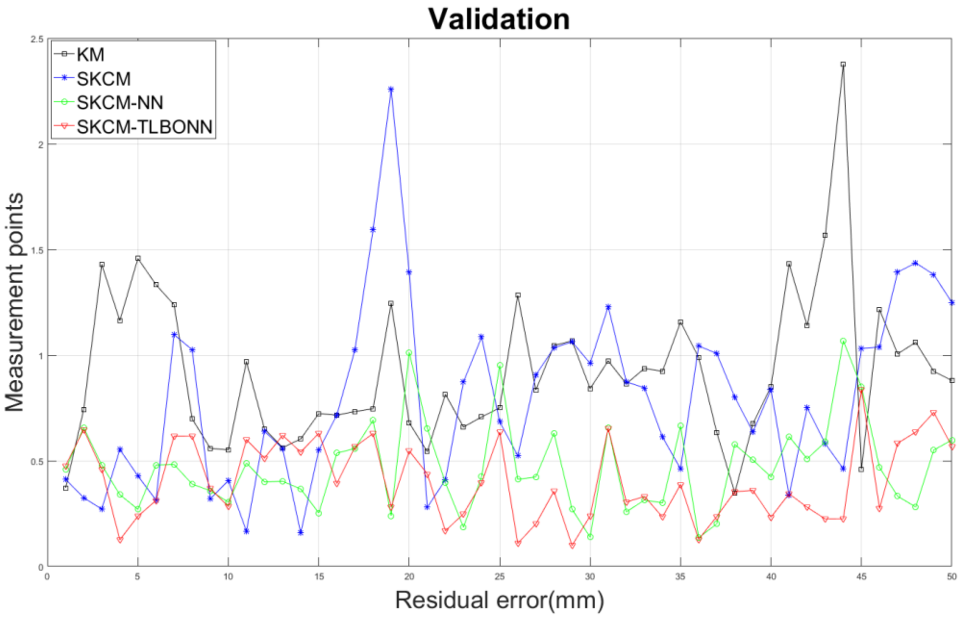


Fig. 6.5. Absolute position error of the HH800 robot after validation using SKCM-TLBONN

Table 6.4. Absolute position accuracy of the HH800 robot (validation) using SKCM-TLBONN

	Mean (mm)	Max. (mm)	Std. (mm)
Nominal robot model	4.0629	6.1681	0.8451
KM	0.9227	2.3769	0.3597
SKCM	0.8009	2.2571	0.4227
SKCM-NN	0.4712	1.0678	0.2087
After proposed technique	0.4057	0.8378	0.1864

6.4.2 Better Convergence of TLBO Neural Network

In this calibration process, several data sets of robot configurations are applied to the back propagation neural network and the TLBO neural network. From this implementation experience, while the back propagation neural network seems to be easily got into the local minima and to need reiteration by randomly resetting the weights and biases to reach the global minima, the TLBO neural network seems to quite easily reach the global minima. Therefore, the TLBO neural network can be said to have better convergence capability.

6.5. Conclusions

In this study, a new calibration method with an error compensating TLBO neural network is proposed for enhancing robot positional accuracy of the industrial manipulators. By combining the joint deflection model with the conventional kinematic model of a manipulator, the geometric errors and joint deflection errors can be simultaneously considered to increase its positional accuracy. Then, a neural network is designed to additionally compensate the unmodeled errors, specially, non-geometric errors. The teaching-learning-based optimization(TLBO) method is employed to optimize weights and biases of the neural network.

In order to demonstrate the effectiveness of the proposed method, real experimental studies are carried out on the HH 800 manipulator. The enhanced position accuracy of the manipulator after the calibration confirms the feasibility and more positional accuracy over the other calibration methods. Additionally, the adopted TLBO neural network can be said to have better convergence capability than the back propagation neural network in this calibration process, less computing and does not require initial parameter in comparing to IWO This advantage allows that the proposed method is more feasible in real offline programming environment.

Chapter 7: SKCM WITH COMPLIANCE PARAMETERS DETERMINED BY TLBO

The elastic of link and joint keep the industrial robotic from archive the specified accuracy to perform task when a payload is carried .This chapter proposed a new calibration method for the industrial robot manipulator, which simultaneously identifies joint compliance and kinematic parameters by combining kinematic calibration technique and teaching-learning-based optimization method. The proposed method combined the geometric and non-geometric calibration to consider the geometric errors and the shifting of joints together to enhance the positional precision of the industrial robotic manipulator. Due to the fusion of the two fundamental methods, the proposed calibration technique has many advantages such as fast convergence, less computing, accurate knowledge of error sources and significantly increase the accuracy of the robot. The proposed modeled-based calibration technique is also performed on YS100 robot to demonstrate its effectiveness and feasibility.

7.1 Introduction

In the past several decades, robot manipulator has played an important role in the industry to carry many works such as welding, spraying paint, carrying load, etc. One of the significant technical problem to addressed is the inability of most robots to be programmed offline or to share program with other robots. This happened due to the nominal geometrically does not contain the errors arising in producing and assembly. Therefore, the robot manipulator has been characterized by high repeatability and low accuracy. To solve this problem, there are huge number of studies concentrated to the area of manipulator calibration. Generally, the robotic calibration is classified into two main categories: the geometric calibration and non-geometric calibration [1, 7, 49–51, 9–11, 22, 25, 46–48]

For decades, one of the most popular ideas in geometric calibration literature is to create a properly error model to suitably describe geometric errors of the robot manipulator based on kinematic parameters. Numerous methods have been suggested to inspect an appropriate model for robotic calibration [13, 16, 68, 69, 17, 18, 22–25, 29, 30]. The D-H model is one of the most fundamental famous calibration methods suggested by Denavit- Hartenberg[23–25]. The method considers a link of robot only as a rigid body which defines the relationship between two neighboring joint axes of a manipulator. This technique is still extensively adopted by many studies with some minute modification [26–28] Besides, there are some population calibration methods. The complete and parametrically continuous (CPC) modeling technique is one of the more versatile alternative modeling conventions designated to fit the needs of manipulator calibration [29, 31]. The zero-reference position method is based upon an arbitrary choice of the zero-reference position, the principle of displacement similarity, and the hand-to-the base sequence of changing the joint variables in a serial manipulator [13, 14, 70]. The product of exponentials (POE) formulation approach is also widely investigated. This method is relied on recursive forward dyad kinematics [16–19]. There have been some attempts to use the Kalman filter, particle filter, etc. to estimate the kinematic parameters [31, 71, 72]. However, the effectiveness of those attempt does not seem to be much. The geometric calibration methods are fast, easy to convert. However, these methods are limited by studying only the geometric error. The industrial manipulator errors may also come from many non-geometric such as joint and link compliance, temperature variation, gear transmission, etc. Among these non-geometric errors , the error is caused by the link and joint elastic is dominant.

In order to address the nonlinear errors and further raising the robotic positional precision, some studies make effort to additionally model the nonlinear errors [6, 20–22, 34–38]. The flexibilities in links and joints are took into account by compliance model [20–22]. Some studies use the optimization criterion to select optimal poses which ensure the best nongeometric errors compensator[35–37]. Other researchers adopt the artificial neural network (ANN) to estimate the compliance parameters [31, 38].There are also some works using the ANN to compensate the un-modeled errors by using the joints configurations[31,39]. Despite the fact that these methods are effective in increasing the positional accuracy of the robot manipulator, these ANN-based methods take the computational effort, time and does not quarantine a convergence. Furthermore, most of ANN-based technique is applied after the kinematic calibration.

The TLBO has been proposed by Rao et al[97]. The method is one of the most famous heuristic method . Based on many advantaged of TLBO such as feasible, strong and fast convergence, the method is adopted by many researchers to solve the optimization problems[99][100]. Generally, the TLBO method is a replicate of human teaching- learning process. It contains two main phases: (1) In the teacher phase, the solution with the best performance becomes the teacher, the other solutions become student. (2) In the student phases, the students learn from each other and from the teacher. Every student will change itself base on this learning process. Then, the process is repeated again until conversion.

In this chapter, a new industrial robot calibration method is proposed to simultaneous identified the kinematic and stiffness parameters by using a combination of a kinematic calibration and teaching-learning-based optimization (TLBO). Owing to the fundamental kinematic calibration and the TLBO, the proposed method is feasibility, require less measurement, fast covert, provide knowledge of error sources, identify the compliance parameter correctly. Finally, the proposed method and two other calibration method are implemented for the simulated and experimental studies on a YS100 robot. The enhanced position accuracy and correctness identified geometric and stiffness parameters of the proposed method over the other calibration methods firmly confirms the effectiveness and feasibility of the methods.

7.2 Modeling of joint compliance

Assuming that the link of robot is much stiffer than its actuated joint, most compliance errors are due to the flexibility of robot joints under the link self-gravity and external payload. Therefore, the effective torque in the joint ith related to all the link after it and the external payload. Thus, the deflection of the ith joint can be express by the effective torque:

$$\tau_i = \sum_{j=i}^{N+1} \tau_{i,j} = \sum_{j=i}^{N+1} J_{\theta_{i,j}}{}^T F_j \tag{7.1}$$

where $F_j = \begin{bmatrix} 0 & 0 & -M_j g \end{bmatrix}$ is a gravity force vector related to the j_{th} link, M_j is the mass of the j_{th} link and g is the gravity coefficient; N is the number of DOF of robot and F_{N+1} is a gravity force vector related to payload; The transpose of the Jacobian matrix is used as a force transformation to find the effective joint torques $\tau_{i,j}$ in the i_{th} joint due to the gravity force in the j_{th} link. The Jacobian matrix is defined as

$$J_{\theta_{i,j}} = z_i \times l_{i,j} \tag{7.2}$$

where $l_{i,j}$ is the 3×1 vector between the origin of the i_{th} frame and the mass center of the j_{th} link.

It is should be noted that in a robot static configuration, a robot joint torque causes a twist deformation about a rotation shaft (which represents the entire drive train from the motor to the associated robot link). The shaft can be considered as a torsional spring in the compliance modeling. Characteristics of a torsion spring is investigated because they are related to modeling of rotational joint compliance. The characteristics of torsion springs are basically presented by non-linear functions, for example [3]:

$$T = k_1 * \delta\theta + k_2 * (\delta\theta)^3 \tag{7.3}$$

where T is the spring torque, $\delta\theta$ is the spring rotational deformation. When the robot joint deformation is small, the linear part becomes dominant. Now, the functional relationship between the joint torque and its deformation can be assumed to be linear in this calibration process.

Assuming the robot joint can be modeled as a linear torsional spring, joint stiffness value of the i_{th} joint is represented by a constant value k_i. The small deflection of the i_{th} joint is obtained by the following equation

$$\Delta\theta_{c_i} = \frac{\tau_i}{k_i} = \tau_i c_i \tag{7.4}$$

where c_i is the joint compliance representing the inverse of joint stiffness k_i. Accordingly, joint deflections of N DOF robot is expressed as

$$\Delta\theta_c = \tau C \tag{7.5}$$

where $\Delta\theta_c = \left[\Delta\theta_{c_1} \; \Delta\theta_{c_2} \; \cdots \; \Delta\theta_{c_k} \; \cdots \; \Delta\theta_{c_N}\right]$ is a N×1 joint deflection vector and $C = \left[c_1 \; c_2 \; \cdots \; c_k \; \cdots \; c_N\right]^T$ is a N×1 joint compliance vector. The diagonal matrix

$\tau = diag\left(\tau_1, \ldots, \tau_K, \ldots, \tau_N\right)$ represents the effective torque in the robot joints at the static equilibrium position.

7.3 SKCM with compliance parameters determined by TLBO

The real position vector P_{real} of the end effector of the robot could be express by the following equation:

$$P_{real} = P_{kin} + \Delta P_{kin} + \Delta P_c + \Delta P_{extra} \tag{7.6}$$

where P_{kin} is the position of the end effector calculated by the kinematic parameter, ΔP_{kin}, ΔP_c, ΔP_{extra} are position errors due to the kinematic parameter errors, joint deflection, and other un-modeled sources, respectively.

In this chapter, a new calibration method (SKCM-C by TLBO) is proposed to simultaneous identify the kinematic and joint compliance parameters to compensate the kinematic parameter errors and the joint shifting. The TLBO method is adopted to identify the joint compliance parameters while the least-square method is using to identify the kinematic parameters together.

The TLBO has been proposed by Rao et al[97]. The method is one of the most famous heuristic method . Based on many advantaged of TLBO such as feasible, strong and fast convergence. Generally, the TLBO method is a replicate of human teaching- learning process. It contains two main phases: (1) In the teacher phase, the solution with the best performance becomes the teacher, the other solutions become student. (2) In the student phases, the students learn from each other and from the teacher. Every student will change itself base on this learning process. Then, the process is repeated again until conversion.

In this study, the TLBO is adopted to identify the compliance parameters. Therefore, the deflection of joints could be calculated. At the same time, a normal kinematic calibration is operated that is also added the elastic of joints. The SKCM-C by TLBO is described as the following.

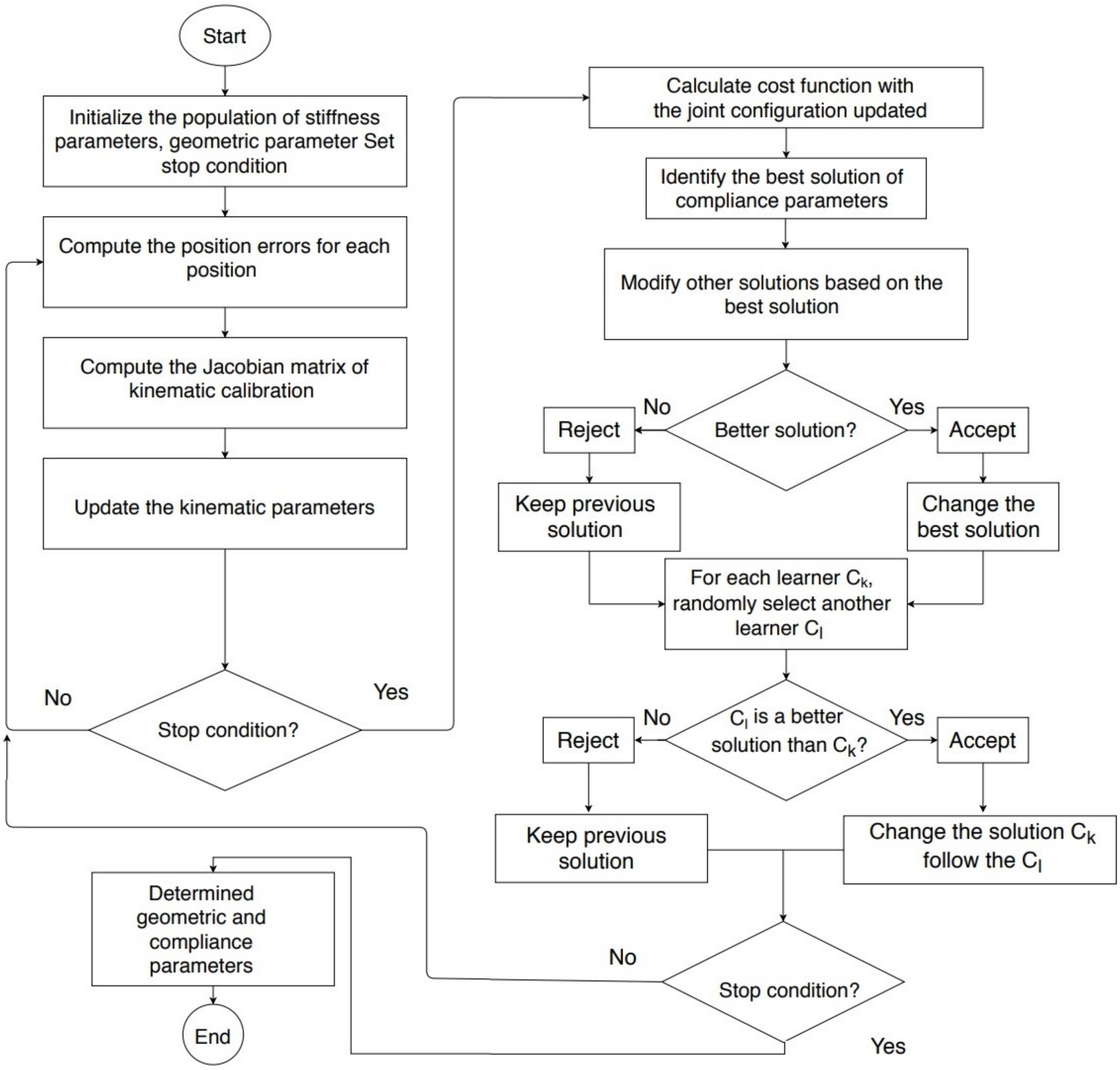

Start
Initialize the population of stiffness parameters, geometric parameter Set stop condition
Compute the position errors for each position
Compute the Jacobian matrix of kinematic calibration
Update the kinematic parameters
No
Stop condition?
Yes
Calculate cost function with the joint configuration updated
Identify the best solution of compliance parameters
Modify other solutions based on the best solution
No
Better solution?
Yes
Reject
Accept
Keep previous solution
Change the best solution
For each learner C$_k$, randomly select another learner C$_l$
No
C$_l$ is a better solution than C$_k$?
Yes
Reject
Accept
Keep previous solution
Change the solution C$_k$ follow the C$_l$
No
Stop condition?
Yes
Determined geometric and compliance parameters
End

Fig. 7.1. Flowchart of the SKCM-C by TLBO .

The algorithm starts with the initial of 25 nominal geometric parameters and m solution of 4 random compliance parameters C_k. A set Q_i including *n* robot configuration is selected for the calibration process. For each solution of compliance parameters, the measured position vector of the robot $\Delta P_{i,k}$ is calculated by following:

$$\Delta P_{j,k} = P_{m_{i,k}} - P_{comp_{i,k}} \tag{7.7}$$

where $P_{m_{i,k}}$, $P_{comp_{i,k}}$ are the positions of the end effector of the *i* robot configuration of the k solution by measuring and computing, respectively. The least squares method is adopted to find the kinematic errors:

$$\Delta\phi = \left[(J^T J)^{-1} J^T\right] \Delta P_{j,k} \tag{7.8}$$

where $\Delta\phi = [\Delta\alpha \Delta a \Delta\beta \Delta b \Delta d \Delta\theta]^T$ is a (25 × 1) vector of kinematic errors, *J* is the (3 ×25) Jacobian matrix that related $\Delta\phi$ to $\Delta P_{j,k}$. The joint configurations are calculated by:

$$\theta_{i,k} = \theta_{E_{i,k}} + \theta_{offset_{i,k}} + \theta_{c_{i,k}} \tag{7.9}$$

where $\theta_{E_{i,k}}$ is the joint configuration ith read by encoder. $\theta_{offset_{i,k}}$ is the offset of configuration ith, and $\theta_{c_{i,k}}$ is the joint deflections calculated by Equation (7.5) .The Eq. (7.8) is applied iteratively until the errors cannot be reduce any further for each compliance parameter solution.

The cost function to evaluate each solution k of compliance parameter is chosen by the maximum square errors:

$$E_k = l_1 mean(\sum_{j=1}^{3} P_{ji}^{\ 2}) + l_2 max(\sum_{j=1}^{3} P_{ji}^{\ 2}) + l_3 std(\sum_{j=1}^{3} P_{ji}^{\ 2}) \tag{7.10}$$

where j=3 is the number dimensions of the error, $l_1 = 1$; $l_2 = 0.5$; $l_3 = 0.3$ are the weights that represent the contribution of mean, max and standard deviation function in forming the cost function of each solution k.

Assume $C_k = \left\{c_{k_1}, c_{k_2}, \ldots, c_{k_n}\right\}$ is the positions of kth learners that represents the set k of compliance parameters. The mean position of the current class is called C_{mean} and the best position of the current teacher is noted as $C_{teacher}$ (which is evaluated by the Eq.7.10). The teacher phase of TLBO is expressed as the following equation:

$$C_{new,k} = C_{old,k} + rand(.) * (C_{teacher} - TF * C_{mean}) \qquad (7.11)$$

where $C_{new,k}$ and $C_{old,k}$ are the new and old positions of the kth learner. $C_{teacher}$ is the position of the current teacher and $rand(.)$ is a random number within the range [0, 1]. TF is the teaching factor. It is set to 1 or 2 randomly. All the learners should be re-evaluated in this phase. If $C_{new,k}$ has a higher performance than $C_{old,k}$ then $C_{new,k}$ will be chosen and will replace $C_{old,k}$, otherwise is not changed.

For each kth learner in the student phase, another learner lth (l≠ k) is randomly selected from the class. The learning process is shown by the following equation:

$$C_{new,k} = \begin{cases} C_{old,k} + rand() * (C_{old,k} - C_{old,l}) * \\ C_{old,k} + rand() * (C_{old,l} - C_{old,k}) ** \end{cases}$$
$$\begin{array}{ll} * & : If\ \ f(C_{old,i}) < f(C_{old,l}) \\ ** & : otherwise \end{array} \qquad (7.12)$$

If $C_{new,k}$ has a higher performance than $C_{old,k}$ then $C_{new,k}$ will be chosen and will replace $C_{old,k}$, otherwise is not changed. In the following step, the stop conditions are checked. If they are satisfied, then the process is stopped. The joint parameters are set following the $C_{teacher}$. The process of the SKCM-C by TLBO is shown in Fig. 7.1.

7.4 Simulation study and results

A simulation study is implemented to verify the effectiveness and the correctness of the proposed calibration method. The simulation processes of the robot are also performed with conventional calibration method (KM), and simultaneous kinematic and compliance calibration method (SKCM) [24]. A simulated Y100 robot is generated by adding the geometric errors to the corresponding nominal D–H parameters of the YS100 robot in Table 7.1. The stiffness parameters of the robot are assumed by the Table 7.2

Table 7.1 Simulated D-H parameters of the Hyundai robot YS100.

i	α_{i-1}(deg)	a_{i-1}(m)	β_{i-1}(deg)	b_{i-1}(m)	d_i(deg)	θ_i(deg)
1	0.1	-0.5	0.1	0.05	0.5	0.2
2	90.2	0.33	-	-	0	0.1
3	-0.1	0.86	0.01	-	-0.001	-1
4	89.9	0.21	-	-	1.02	-1.3
5	-90.2	0.002	-	-	0.002	-1.2
6	90.1	0.003	-	-	0.185	0
T	-	-0.3	-	0.05	0.45	-

Table 7.2. Simulated stiffness parameters of the Hyundai robot YS100

	K_2	K_3	K_4	K_5
Stiffness	1.5*10^6	1*10^6	1.2 *10^6	1*10^6

The robot is simulated with the geometric link errors, joint compliance errors, link deflection errors, and measurement noises. A set of 30 robot joint readings is generated such that the simulated robot end-effector positions that are computed by its forward kinematics cover its entire workspace. The positions are also corrupted by the (3×1) vector of random values (with the assumption of a normal distribution N(0, σ), with a standard deviation σ=0.01 mm) to simulate the measurement noise.

The initial value of TLBO : k=50 solution of 4 random compliance parameters . By utilizing the SKCM-C by TLBO, the kinematic and joint compliance parameters are identified. These parameters are presented in Table 7.3 and Table 7.4 including 4 joint compliance parameters and 25 kinematic parameters.

Table 7.3 Simulated D-H parameters of the Hyundai robot YS100 using SKCM-C by TLBO.

i	α_{i-1}(deg)	a_{i-1}(m)	β_{i-1}(deg)	b_{i-1}(m)	d_i(deg)	θ_i(deg)
1	2.091	-0.498	0.207	0.0524	0.55	5.207
2	90.201	0.33	-	-	0	1
3	-0.101	0.86	0.065	-	-0.001	-1.002
4	89.893	0.21	-	-	1.02	-1.3
5	-90.199	0.002	-	-	0.002	-1.206
6	90.098	0.0031	-	-	0.185	0
T	-	-0.3	-	0.05	0.45	-

Table 7.4. Simulated stiffness parameters of the Hyundai robot YS100 using SKCM-C by TLBO

	K_2	K_3	K_4	K_5
Stiffness	1499400	1010400	1227800	1021300

It is clearly to see from these tables that the geometric parameters are precisely identified by the proposed method. Moreover, the stiffness parameters are correctly determined by the proposed method . The precision of K_2, K_3, K_4, K_5 are 99.94%, 98.996%, 97.68%, 97.87% respectively.

Overall, the absolute position accuracy of robot YS100 in the simulation process by the two methods are shown in Table 7.5.

Table 7.5. Absolute position accuracy of the YS100 robot (Simulation).

	Mean(mm)	Max.(mm)	Std.(mm)
Nominal robot model	3.3921	5.3390	1.0916
KM	0.3687	0.7606	0.1558
Proposed method	0.0119	0.024	0.0061

The Table 7.5 shows that the position errors generated by the proposed method is better than the conventional calibration methods. The mean of position errors generated by the proposed method is more precise by 99.64% than the errors before calibration (from 3.3921 mm to 0.0119 mm), by 96.77% than the errors by conventional calibration method (from 0.3687 mm to 0.0119 mm). The proposed technique also has better maximum position error as well as the best standard deviation.

7.5 Experiment and validation results

In order to verify the effectiveness of the proposed method, the SKCM-C by TLBO is used to perform on a 6 DOF serial robot (Hyundai YS100). In addition, the calibration results are validated by comparing the positions of the end-effector predicted by the calibrated manipulator with the measured positions at the robot configurations that are not used in calibration. Moreover, the calibration results obtained by the proposed calibration algorithm are compared with those obtained by the conventional kinematic calibration to show its capability.

7.5.1 Experimental calibration results.

The calibration process is carried on a YS100 robot with 110 kg payload, and API laser tracker and an accompanying laser tracker . The API laser tracker has the accuracy of 0.01 mm/m, repeatability of ±0.006 mm/m. The reflector is fixed at a particular location of the robot end-effector. In order to acquire suitable measurement data for robot parameter identification, the robot moves its end-effector to positions that entirely cover the workspace. The 3D coordinates of the end effector are measured by the Laser Tracker and saved in a computer. At the same time, the associated robot joint readings also are recorded. These measurements will be randomly grouped in three sets . A set of 25 robot configurations (Q1) is used in parameter identification.

The initial value of TLBO : m=50 solution of 4 random compliance parameters . By utilizing the SKCM-C by TLBO method that is clearly described in the section 7.3, the kinematic and joint compliance parameters are identified. These parameters are presented in Table 7.6 and Table 7.7 including 4 joint compliance parameters and 25 kinematic parameters.

Table 7.6 Identified D-H parameters of the Hyundai robot YS100.

i	α_{i-1}(deg)	a_{i-1}(m)	β_{i-1}(deg)	b_{i-1}(m)	d_i(deg)	θ_i(deg)
1	-0.1748	-0.0163	0.5113	0.0489	0.4868	0.2340
2	90.0140	0.3198	-	-	0(X)	1.4122
3	-0.0160	0.8700	0.0132	-	-0.0018	-1.6557
4	89.9848	0.2004	-	-	1.0299	-1.3602
5	-90.0122	0.0001	-	-	0.0008	-2.2655
6	90.065	0.0021	-	-	0.1850(X)	0(X)
T	-	-0.2815	-	0.0484	0.4239	-

("-": unavailable, "X": un-identifiable)

Table 7.7. Identified stiffness parameters of the Hyundai robot YS100

	K_2	K_3	K_4	K_5
Stiffness	1.732*10^6	1.196*10^6	0.297*10^6	0.588*10^6

The results in Table 7.8 and Fig. 7.2 show that the proposed method achieve the better performance over the conventional kinematic calibration method. The mean of position errors generated by the proposed method is more precise by 97.77% than the errors before calibration (from 12.607 mm to 0.2802 mm), by 26.16% than the errors by KM (from 0.3847 mm to 0.2802 mm). The proposed technique also has better maximum position error as well as better standard deviation.

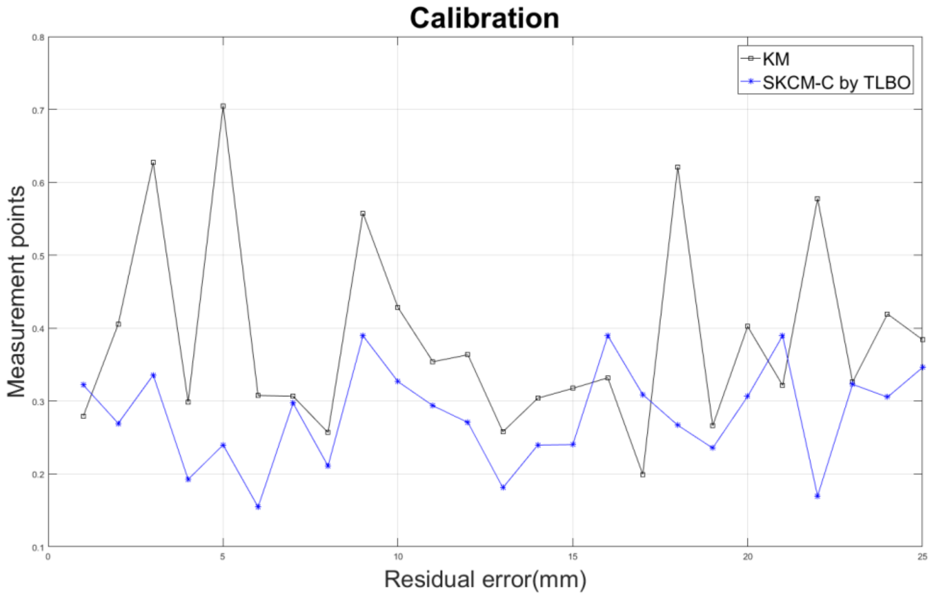


Fig. 7.2. Absolute position error of the YS100 robot after calibration.

Table 7.8. Absolute position accuracy of the YS100 robot (Calibration).

	Mean(mm)	Max.(mm)	Std.(mm)
Nominal robot model	12.607	27.1121	5.4769
KM	0.3847	0.7049	0.1327
Proposed method	0.2802	0.3897	0.0674

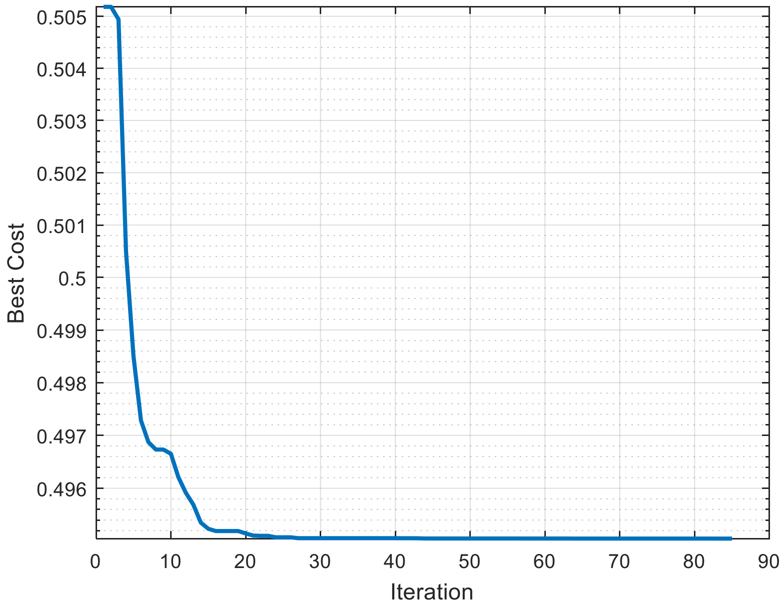


Fig. 7.3. Convergence of the calibration process.

The Fig. 7.3 shows the convergence of the method. Owing to the ability of the SKCM-C by TLBO the process converges fast.

7.5.2 Experimental validation results

In order to show the general capability over the entire robot workspace, the method should be investigated by another robot configuration. The other set of 45 robot configurations (Q2) is randomly selected overall the workspace to show the general capability over the entire robot workspace of the method. The processes are carried out for the KM method and the proposed method.

The calibration results with Q2 data set are shown in Table X and Fig.5. The Fig. 5 clearly shows the residual error of each poses using 2 methods in the validation process. It is clearly to see from the figure that the proposed method is better than the KM both the positions used in the calibration process and the general position in overall workspace. The mean of position errors generated by the proposed method is more precise by 98.17% than the errors before calibration (from 12.371 mm to 0.2257 mm), by 41.25% than the errors by KM (from 0.3842 mm to 0.2257 mm). It should be noted that the proposed technique also decreases the maximum position error by 52.38% than the errors by KM method (from 0.793 mm to 0.3776 mm).

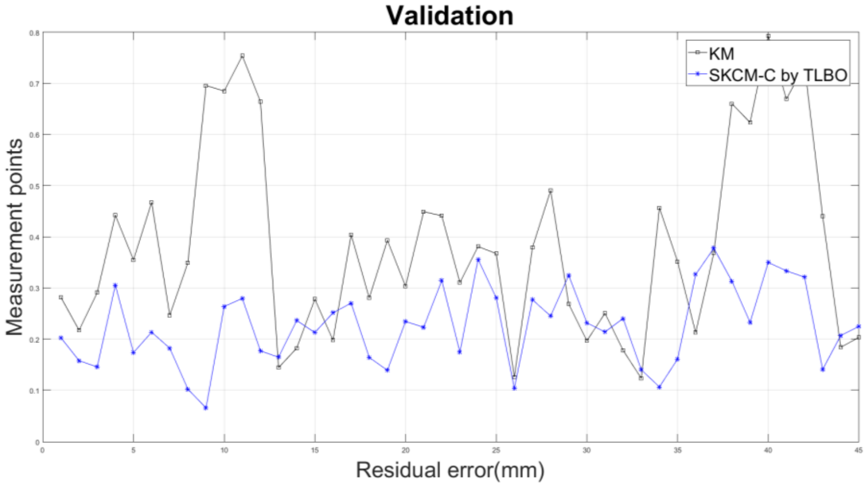


Fig. 7.4. Absolute position error of the YS100 robot after validation.

Table 7.9. Absolute position accuracy of the YS100 robot (Validation).

	Mean(mm)	Max.(mm)	Std.(mm)
Nominal robot model	12.371	25.8132	5.673
KM	0.3842	0.7930	0.1864
After proposed technique	0.2257	0.3776	0.0765

7.6. Conclusions

In this study, a new calibration method for the industrial robot manipulator is proposed, which simultaneous identifies joint compliance and kinematic parameters by combining of kinematic calibration technique and teaching-learning-based optimization method. The proposed method combined the geometric and non-geometric calibration to consider the geometric errors and the shifting of joints together to enhance positional precision of the industrial robotic manipulator. Due to the fusion of the two fundamental methods, the proposed calibration technique has many advantages such as fast convergence, accurate knowledge of error sources and significantly increase the accurate of the robot. The proposed modeled-based calibration technique is also performed on YS100 robot to demonstrate its effectiveness and feasibility. The enhanced position accuracy of the manipulator after the calibration confirms the feasibility and more positional accuracy over the other calibration methods.

Additionally, the adopted SKCM-C by TLBO can be said to have a good convergence capability in this calibration process. This advantage allows that the proposed method is more feasible in real offline programming environment.

Chapter 8: CONCLUSIONS AND FUTURE WORKS

8.1 Conclusions

In this study, new calibration methods are proposed for enhancing robot positional accuracy of the industrial manipulators. The modeled based calibration methods are introduced to give an overview of calibration process. To archive better positional precision results, a new method includes the kinematic calibration and non-geometric compensation with a RBF compensator that compensates for compliance errors based on the effective torques. The advantages of the suggested method are easy for implementing, removing the need for torque sensors, high ability to enhance the precision of the manipulator.

To further improving the calibration result, a new calibration method is proposed by combining the joint deflection model with the conventional kinematic model of a manipulator, the geometric errors and joint deflection errors can be simultaneously considered to increase its positional accuracy. Then, a neural network is designed to additionally compensate the unmodeled errors, specially, non-geometric errors (SKCM-NN).

The SKCM-NN is designed to additionally compensate the unmodeled errors, specially, non-geometric errors. However, the conventional backpropagation neural network has many drawbacks such as easily falling into local minima, not easy to converge. For increasing the ability of the neural network, IWO method is used for better convergence capability than the back propagation neural network in this calibration process. However, the IWO is sensitive to initial parameters. If the initial parameters are not properly chosen, the NN would lose its ability of compensating the residual errors.

Then, the TLBO is used for determining weights and biases of the neural network. Without choosing any initial parameters, the TLBO-NN shows its ability of convergence, for better convergence capability than the back propagation neural network, and less computation in comparing to IWO-NN. From the investigation of this thesis, the SKCM-TLBONN is the best among the proposed methods.

This thesis also proposed a new calibration method for the industrial robot manipulator is proposed, which directly identifies joint compliance parameters by combining kinematic calibration technique and teaching-learning-based optimization method. The proposed method combined the geometric and non-geometric calibration to consider the geometric errors and the shifting of joints together to enhance the positional precision of the industrial robotic manipulator. Due to the fusion of the two fundamental methods, the proposed calibration technique has many advantages such as fast convergence, less computing, accurate knowledge of error sources and significantly increase the accuracy of the robot

In order to demonstrate the effectiveness of the proposed method, real experimental studies are carried out on manipulators. The enhanced position accuracy of the manipulator after the calibration confirms the feasibility and more positional accuracy over the other calibration methods.

8.2 Future works

Although the hybrid calibration methods have been proposed in this thesis, the research work still needs to be studied in future for enhanced robot calibration techniques.

The following are some of attractive topics:

- Modeling robot joint stiffness as a polynomial function of applied torques to achieve more precise joint compliance error model. An optimization method will be used to determine the coefficient of the polynomial function.
- Determining optimal poses by using heuristic optimization method.

Publications

SCI(E) Journals:

1. Nguyen, Hoai-Nhan, Phu-Nguyen Le, and Hee-Jun Kang. "A new calibration method for enhancing robot position accuracy by combining a robot model–based identification approach and an artificial neural network–based error compensation technique." *Advances in Mechanical Engineering* 11.1 (2019): 1687814018822935.
2. Le, Phu-Nguyen, and Hee-Jun Kang. "A Robotic Calibration Method Using a Model-Based Identification Technique and an Invasive Weed Optimization Neural Network Compensator." *Applied Sciences* 10.20 (2020): 7320.
3. Le, Phu-Nguyen, and Hee-Jun Kang. "Robot Manipulator Calibration Using a Model Based Identification Technique and a Neural Network with the Teaching Learning-Based Optimization." *IEEE Access* 8 (2020): 105447-105454.
4. Le, Phu-Nguyen, and Hee-Jun Kang. "A new manipulator calibration method for the identification of kinematic and compliance errors using optimal pose selection." *Neurocomputing.(Recommended from ICIC2020 conference, summited)*

Chapters:

1. Le, Phu-Nguyen, and Hee-Jun Kang. "A New Hybrid Calibration Method for Robot Manipulators by Combining Model–Based Identification Technique and a Radial Basis Function–Based Error Compensation." *International Conference on Intelligent Computing*. Springer, Cham, 2019.
2. Le, Phu-Nguyen, and Hee-Jung Kang. "A New Robotic Manipulator Calibration Method of Identification Kinematic and Compliance Errors." *International Conference on Intelligent Computing*. Springer, Cham, 2020.

References

1. Whitney, D.E., Lozinski, C.A., Rourke, J.M.: Industrial robot forward calibration method and results. J. Dyn. Syst. Meas. Control. 108, 1–8 (1986).

2. Judd, R., Knasinski, A.: A technique to calibrate industrial robots with experimental verification. In: Proceedings. 1987 IEEE International Conference on Robotics and Automation. pp. 351–357 (1987).

3. Renders, J.-M., Rossignol, E., Becquet, M., Hanus, R., others: Kinematic calibration and geometrical parameter identification for robots. IEEE Trans. Robot. Autom. 7, 721–732 (1991).

4. Becquet, M.: Analysis of flexibility sources in robot structure. In: IMACS/IFAC. Symp. Modeling and Simulation of Distributed Parameters, Hiroshima, Japan. pp. 419–424 (1987).

5. Caenen, J.-L., Angue, J.-C.: Identification of geometric and nongeometric parameters of robots. In: Proceedings., IEEE International Conference on Robotics and Automation. pp. 1032–1037 (1990).

6. Jang, J.H., Kim, S.H., Kwak, Y.K.: Calibration of geometric and non-geometric errors of an industrial robot. Robotica. 19, 311–321 (2001).

7. Gong, C., Yuan, J., Ni, J.: Nongeometric error identification and compensation for robotic system by inverse calibration. Int. J. Mach. Tools Manuf. 40, 2119–2137 (2000). https://doi.org/10.1016/S0890-6955(00)00023-7.

8. Bernhardt, R., Albright, S.L.: Robot calibration. Chapman and Hall, London et al. 1993.

9. Roth, Z., Mooring, B., Ravani, B.: An overview of robot calibration. IEEE J. Robot. Autom. 3, 377–385 (1987).

10. Mooring, B.W., Roth, Z.S., Driels, M.R.: Fundamentals of manipulator calibration. Wiley New York (1991).

11. Hartenberg, R.S., Denavit, J.: A kinematic notation for lower pair mechanisms based on matrices. J. Appl. Mech. 77, 215–221 (1955).

12. Sciavicco, L., Siciliano, B., Villani, L., Oriolo, G.: Robotics: Modelling, planning and Control, ser. Advanced Textbooks in Control and Signal Processing, (2011).

13. Gupta, K.C.: Kinematic analysis of manipulators using the zero reference position description. Int. J. Rob. Res. 5, 5–13 (1986).

14. Cheng, L.-P., Kazerounian, K.: Study and enumeration of singular configurations for the kinematic model of human arm. In: Proceedings of the IEEE 26th Annual Northeast Bioengineering Conference (Cat. No. 00CH37114). pp. 3–4 (2000).

15. Mooring, B., Tang, G.: An Improved Method for Identifying the Kinematic Parameters in a Six-Axis Robot. Proc. ASME Comput. Eng. Conf. 79–84 (1984).

16. Okamura, K., Park, F.C.: Kinematic calibration using the product of exponentials formula. Robotica. 14, 415–421 (1996).

17. Chen, G., Kong, L., Li, Q., Wang, H., Lin, Z.: Complete, minimal and continuous error models for the kinematic calibration of parallel manipulators based on POE formula. Mech. Mach. Theory. 121, 844–856 (2018).

18. Chen, G., Wang, H., Lin, Z.: Determination of the identifiable parameters in robot calibration based on the POE formula. IEEE Trans. Robot. 30, 1066–1077 (2014).

19. Wang, H., Lu, X., Cui, W., Zhang, Z., Li, Y., Sheng, C.: General inverse solution of six-degrees-of-freedom serial robots based on the product of exponentials model. Assem. Autom. 38, 361–367 (2018).

20. Dumas, C., Caro, S., Garnier, S., Furet, B.: Joint stiffness identification of six-revolute industrial serial robots. Robot. Comput. Integr. Manuf. 27, 881–888 (2011).

21. Slavković, N.R., Milutinović, D.S., Kokotović, B.M., Glavonjić, M.M., Živanović, S.T., Ehmann, K.F.: Cartesian compliance identification and analysis of an articulated machining robot. FME Trans. 41, 83–95 (2013).

22. Zhou, J., Nguyen, H.-N., Kang, H.-J.: Simultaneous identification of joint compliance and kinematic parameters of industrial robots. Int. J. Precis. Eng. Manuf. 15, 2257–2264 (2014).

23. John, J.C., others: Introduction to robotics: mechanics and control. Read. Addison-Wesley. (1989).

24. Hayati, S., Mirmirani, M.: Improving the absolute positioning accuracy of robot manipulators. J. Robot. Syst. 2, 397–413 (1985).

25. Hayati, S., Tso, K., Roston, G.: Robot geometry calibration. In: Proceedings. 1988 IEEE International Conference on Robotics and Automation. pp. 947–951 (1988).

26. Klug, C., Schmalstieg, D., Gloor, T., Arth, C.: A complete workflow for automatic forward kinematics model extraction of robotic total stations using the Denavit-Hartenberg convention. J. Intell. Robot. Syst. 95, 311–329 (2019).

27. Faria, C., Vilaça, J.L., Monteiro, S., Erlhagen, W., Bicho, E.: Automatic Denavit-Hartenberg Parameter Identification for Serial Manipulators. In: IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society. pp. 610–617 (2019).

28. Morar, C.A., H\uagan, M., Doroftei, I., Marinca, \cStefan: Analog Matrix Multiplier Dedicated to the Denavit-Hartenberg Algorithm. In: 2019 International Symposium on Signals, Circuits and Systems (ISSCS). pp. 1–4 (2019).

29. Zhuang, H., Roth, Z.S., Hamano, F.: A complete and parametrically continuous kinematic model for robot manipulators. In: Proceedings., IEEE International Conference on Robotics and Automation. pp. 92–97 (1990).

30. Zhuang, H., Wang, L.K., Roth, Z.S.: Error-model-based robot calibration using a modified CPC model. Robot. Comput. Integr. Manuf. 10, 287–299 (1993).

31. Nguyen, H.N., Zhou, J., Kang, H.J.: A calibration method for enhancing robot accuracy through integration of an extended Kalman filter algorithm and an artificial neural network. Neurocomputing. 151, 996–1005 (2015). https://doi.org/10.1016/j.neucom.2014.03.085.

32. Jiang, Z., Zhou, W., Li, H., Mo, Y., Ni, W., Huang, Q.: A new kind of accurate calibration method for robotic kinematic parameters based on the extended Kalman and particle filter algorithm. IEEE Trans. Ind. Electron. 65, 3337–3345 (2017).

33. Du, G., Liang, Y., Li, C., Liu, P.X., Li, D.: Online Robot Kinematic Calibration Using Hybrid Filter with Multiple Sensors. IEEE Trans. Instrum. Meas. (2020).

34. Nakamura, Y., Ghodoussi, M.: Dynamics computation of closed-link robot mechanisms with nonredundant and redundant actuators. Int. Conf. Robot. Autom. 5, 294–302 (1989).

35. Zhou, J., Nguyen, H.-N., Kang, H.-J.: Selecting optimal measurement poses for kinematic calibration of industrial robots. Adv. Mech. Eng. 6, 291389 (2014).

36. Wang, W., Song, H., Yan, Z., Sun, L., Du, Z.: A universal index and an improved PSO algorithm for optimal pose selection in kinematic calibration of a novel surgical robot. Robot. Comput. Integr. Manuf. 50, 90–101 (2018).

37. Chen, X., Zhang, Q., Sun, Y.: Non-kinematic calibration of industrial robots using a rigid--flexible coupling error model and a full pose measurement method. Robot. Comput. Integr. Manuf. 57, 46–58 (2019).

38. Aoyagi, S., Kohama, A., Nakata, Y., Hayano, Y., Suzuki, M.: Improvement of robot accuracy by calibrating kinematic model using a laser tracking system-compensation of non-geometric errors using neural networks and selection of optimal measuring points using genetic algorithm. In: 2010 IEEE/RSJ International conference on intelligent robots and systems. pp. 5660–5665 (2010).

39. Nguyen, H.-N., Le, P.-N., Kang, H.-J.: A new calibration method for enhancing robot position accuracy by combining a robot model--based identification approach and an artificial neural network--based error compensation technique. Adv. Mech. Eng. 11, 1687814018822935 (2019).

40. Meggiolaro, M.A., Dubowsky, S., Mavroidis, C.: Geometric and elastic error calibration of a high accuracy patient

positioning system. Mech. Mach. Theory. 40, 415–427 (2005).

41. Kamali, K., Joubair, A., Bonev, I.A., Bigras, P.: Elasto-geometrical calibration of an industrial robot under multidirectional external loads using a laser tracker. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). pp. 4320–4327 (2016).

42. Müller, R., Scholer, M., Blum, A., Kanso, A.: Identification of the Dynamic Parameters of a Robotic Tool Based on Integrated Torque Sensors. In: 2019 23rd International Conference on Mechatronics Technology (ICMT). pp. 1–6 (2019).

43. Besset, P., Olabi, A., Gibaru, O.: Advanced calibration applied to a collaborative robot. In: 2016 IEEE International Power Electronics and Motion Control Conference (PEMC). pp. 662–667 (2016).

44. Xia, C., Liu, Y., Lei, B., Xiang, X.: Research on a Generalized Regression Neural Network Model of Thermocouple and it's Spread Scope. In: 2008 Fourth International Conference on Natural Computation. pp. 109–113 (2008).

45. Corino, V.D.A., Matteucci, M., Cravello, L., Ferrari, E., Ferrari, A.A., Mainardi, L.T.: Long-term heart rate variability as a predictor of patient age. Comput. Methods Programs Biomed. 82, 248–257 (2006).

46. Duelen, G., Schröer, K.: Robot calibration—method and results. Robot. Comput. Integr. Manuf. 8, 223–231 (1991).

47. Hudgens, J.C.: A Compliance Parameter Estimation Method for Serial Manipulators. In: ASME DSC. pp. 15–23 (1991).

48. Veitschegger, W., Wu, C.-H.: Robot accuracy analysis based on kinematics. IEEE J. Robot. Autom. 2, 171–179 (1986).

49. Khalil, W., Caenen, J.L., Enguehard, C.: Identification and calibration of the geometric parameters of robots. In: Experimental Robotics I. pp. 528–538 (1990).

50. Park, I.-W., Lee, B.-J., Cho, S.-H., Hong, Y.-D., Kim, J.-H.: Laser-based kinematic calibration of robot manipulator using differential kinematics. IEEE/ASME Trans. mechatronics. 17, 1059–1067 (2011).

51. To, M., Webb, P.: An improved kinematic model for calibration of serial robots having closed-chain mechanisms. Robotica. 30, 963 (2012).

52. Martinelli, A., Tomatis, N., Tapus, A., Siegwart, R.: Simultaneous localization and odometry calibration for mobile robot. In: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453). pp. 1499–1504 (2003).

53. Song, Y., Zhang, J., Lian, B., Sun, T.: Kinematic calibration of a 5-DoF parallel kinematic machine. Precis. Eng. 45, 242–261 (2016).

54. Naruse, H., Nobiki, A., Yabuta, T., Tateda, M.: High-accuracy multiviewpoint stereo measurement using the maximum-likelihood method. IEEE Trans. Ind. Electron. 44, 571–578 (1997).

55. Alici, G., Shirinzadeh, B.: A systematic technique to estimate positioning errors for robot accuracy improvement using laser interferometry based sensing. Mech. Mach. Theory. 40, 879–906 (2005).

56. Wang, D., Bai, Y., Zhao, J.: Robot manipulator calibration using neural network and a camera-based measurement system. Trans. Inst. Meas. Control. 34, 105–121 (2012).

57. Takanashi, N.: 6 DOF manipulators absolute positioning accuracy improvement using a neural-network. In: EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications. pp. 635–640 (1990).

58. Zhong, X., Lewis, J., N-Nagy, F.L.: Inverse robot calibration using artificial neural networks. Eng. Appl. Artif. Intell. 9, 83–93 (1996).

59. Meggiolaro, M.A., Scriffignano, G., Dubowsky, S.: Manipulator calibration using a single endpoint contact

constraint. In: Proceedings of ASME Design Engineering Technical Conference, Baltimore, USA (2000).

60. Zhong, X.-L., Lewis, J.M.: A new method for autonomous robot calibration. In: Proceedings of 1995 IEEE International Conference on Robotics and Automation. pp. 1790–1795 (1995).

61. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: Robotics: modelling, planning and control. Springer Science & Business Media (2010).

62. Luh, J., Zheng, Y.-F.: Computation of input generalized forces for robots with closed kinematic chain mechanisms. IEEE J. Robot. Autom. 1, 95–103 (1985).

63. Stanton, D., Mayer, J.R.R.: Robot calibration within CIM-SEARCH/I. Robot calibration. 1st ed. London. 57–76 (1993).

64. Judd, R.P., Knasinski, A.B.: A Technique to Calibrate Industrial Robots with Experimental Verification, https://ieeexplore.ieee.org/abstract/document/1088010/, (1990). https://doi.org/10.1109/70.88114.

65. Robert, H.-N., others: Theory of the backpropagation neural network. Proc. 1989 IEEE IJCNN. 1, 593–605 (1989).

66. Goh, A.T.C.: Back-propagation neural networks for modeling complex systems. Artif. Intell. Eng. 9, 143–151 (1995).

67. Kang, H.-J., Jeong, J.-W., Shin, S.-W., Suh, Y.-S., Ro, Y.-S.: Autonomous kinematic calibration of the robot manipulator with a linear laser-vision sensor. In: International Conference on Intelligent Computing. pp. 1102–1109 (2007).

68. Brisan, C., Hiller, M.: Aspects of calibration and control of PARTNER robots. In: 2006 IEEE International Conference on Automation, Quality and Testing, Robotics. pp. 272–277 (2006).

69. Manne, R.: Analysis of two partial-least-squares algorithms for multivariate calibration. Chemom. Intell. Lab. Syst. 2, 187–197 (1987).

70. MOORING, B.W.: An improved method for identifying the kinematic parameters in a six axis robots. Comput. Eng. Proc. Int. Comput. Eng. Conf. Exhib. 1984. 1, 79–84 (1984).

71. Jiang, Z., Zhou, W., Li, H., Mo, Y., Ni, W., Huang, Q.: A New Kind of Accurate Calibration Method for Robotic Kinematic Parameters Based on the Extended Kalman and Particle Filter Algorithm. IEEE Trans. Ind. Electron. 65, 3337–3345 (2018). https://doi.org/10.1109/TIE.2017.2748058.

72. Li, Y., Liu, X., Peng, Z., Liu, Y.: The identification of joint parameters for modular robots using fuzzy theory and a genetic algorithm. Robotica. 20, 509–517 (2002).

73. Lightcap, C., Hamner, S., Schmitz, T., Banks, S.: Improved positioning accuracy of the PA10-6CE robot with geometric and flexibility calibration. IEEE Trans. Robot. 24, 452–456 (2008).

74. Swevers, J., Ganseman, C., Tukel, D.B., De Schutter, J., Van Brussel, H.: Optimal robot excitation and identification. IEEE Trans. Robot. Autom. 13, 730–740 (1997).

75. Zhou, J., Kang, H.-J.: A hybrid least-squares genetic algorithm--based algorithm for simultaneous identification of geometric and compliance errors in industrial robots. Adv. Mech. Eng. 7, 1687814015590289 (2015).

76. Li, A., Wu, D., Ma, Z.: Robot calibration based on multi-thread particle swarm optimization. In: 2008 6th IEEE International Conference on Industrial Informatics. pp. 454–457 (2008).

77. Xie, X., Li, Z., Wang, G.: Manipulator calibration based on PSO-RBF neural network error model. In: AIP Conference Proceedings. p. 20026 (2019).

78. Tao, P.Y., Yang, G.: Calibration of industrial robots with product-of-exponential (poe) model and adaptive neural networks. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 1448–1454 (2015).

79. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. (1985).

80. Baker, M.R., Patil, R.B.: Universal approximation theorem for interval neural networks. Reliab. Comput. 4, 235–

239 (1998).

81. Gao, G., Liu, F., San, H., Wu, X., Wang, W.: Hybrid optimal kinematic parameter identification for an industrial robot based on BPNN-PSO. Complexity. 2018, (2018).

82. Wang, Z., Chen, Z., Wang, Y., Mao, C., Hang, Q.: A robot calibration method based on joint angle division and an artificial neural network. Math. Probl. Eng. 2019, (2019).

83. Wang, X., Tang, Z., Tamura, H., Ishii, M., Sun, W.D.: An improved backpropagation algorithm to avoid the local minima problem. Neurocomputing. 56, 455–460 (2004).

84. Wang, D.-S., Xu, X.-H.: Genetic neural network and application in welding robot error compensation. In: 2005 International Conference on Machine Learning and Cybernetics. pp. 4070–4075 (2005).

85. Jiang, G., Luo, M., Bai, K., Chen, S.: A precise positioning method for a puncture robot based on a PSO-optimized BP neural network algorithm. Appl. Sci. 7, 969 (2017).

86. Zhang, Y., Jin, Z., Chen, Y.: Hybrid teaching--learning-based optimization and neural network algorithm for engineering design optimization problems. Knowledge-Based Syst. 187, 104836 (2020).

87. Nayak, J., Naik, B., Pelusi, D., Krishna, A.V.: A Comprehensive Review and Performance Analysis of Firefly Algorithm for Artificial Neural Networks. In: Nature-Inspired Computation in Data Mining and Machine Learning. pp. 137–159. Springer (2020).

88. Giri, R., Chowdhury, A., Ghosh, A., Das, S., Abraham, A., Snasel, V.: A modified invasive weed optimization algorithm for training of feed-forward neural networks. In: Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics. pp. 3166–3173 (2010). https://doi.org/10.1109/ICSMC.2010.5642265.

89. Mehrabian, A.R., Lucas, C.: A novel numerical optimization algorithm inspired from weed colonization. Ecol. Inform. 1, 355–366 (2006).

90. Le, P.-N., Kang, H.-J.: A New Hybrid Calibration Method for Robot Manipulators by Combining Model--Based Identification Technique and a Radial Basis Function--Based Error Compensation. In: International Conference on Intelligent Computing. pp. 20–31 (2019).

91. Li, S., Sun, Y.: A novel numerical optimization algorithm inspired from garden balsam. Neural Comput. Appl. (2018). https://doi.org/10.1007/s00521-018-3905-3.

92. Huang, L., Asteris, P.G., Koopialipoor, M., Armaghani, D.J., Tahir, M.M.: Invasive weed optimization technique-based ANN to the prediction of rock tensile strength. Appl. Sci. 9, 5372 (2019).

93. Du, G., Zhang, P., Liu, X.: Markerless Human-Manipulator Interface Using Leap Motion with Interval Kalman Filter and Improved Particle Filter. IEEE Trans. Ind. Informatics. 12, 694–704 (2016). https://doi.org/10.1109/TII.2016.2526674.

94. Du, G., Zhang, P.: A Markerless Human-Robot Interface Using Particle Filter and Kalman Filter for Dual Robots. IEEE Trans. Ind. Electron. 62, 2257–2264 (2015). https://doi.org/10.1109/TIE.2014.2362095.

95. He, W., Xue, C., Yu, X., Li, Z., Yang, C.: Admittance-Based Controller Design for Physical Human-Robot Interaction in the Constrained Task Space. IEEE Trans. Autom. Sci. Eng. 1–13 (2020). https://doi.org/10.1109/TASE.2020.2983225.

96. Yu, X., He, W., Li, H., Sun, J.: Adaptive Fuzzy Full-State and Output-Feedback Control for Uncertain Robots With Output Constraint. IEEE Trans. Syst. Man, Cybern. Syst. 1–14 (2020). https://doi.org/10.1109/TSMC.2019.2963072.

97. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching--learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput. Des. 43, 303–315 (2011).

98. Zou, F., Wang, L., Hei, X., Chen, D.: Teaching--learning-based optimization with learning experience of other learners and its application. Appl. Soft Comput. 37, 725–736 (2015).

99. Niknam, T., Zare, M., Aghaei, J.: Scenario-based multiobjective volt/var control in distribution networks including renewable energy sources. IEEE Trans. Power Deliv. 27, 2004–2019 (2012).

100. Haghdar, K.: Optimal DC source influence on selective harmonic elimination in multilevel inverters using teaching--learning-based optimization. IEEE Trans. Ind. Electron. 67, 942–949 (2019).