

COMPUTATIONALLY EFFICIENT AND ROBUST  
KINEMATIC CALIBRATION METHODOLOGIES AND  
THEIR APPLICATION TO INDUSTRIAL ROBOTS

Dissertation

Submitted to

The School of Engineering of the  
UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree of

Doctor of Philosophy in Electrical Engineering

By

Temesguen Messay-Kebede, B.S., M.S.

UNIVERSITY OF DAYTON

Dayton, Ohio

December, 2014

COMPUTATIONALLY EFFICIENT AND ROBUST KINEMATIC  
CALIBRATION METHODOLOGIES AND THEIR APPLICATION TO  
INDUSTRIAL ROBOTS

Name: Messay-Kebede, Temesguen

APPROVED BY:

---

Raúl Ordóñez, Ph.D.  
Advisory Committee Chairman  
Professor  
Electrical & Computer Engineering

---

Russell C. Hardie, Ph.D.  
Committee Member  
Professor  
Electrical & Computer Engineering

---

John Loomis, Ph.D.  
Committee Member  
Associate Professor  
Electrical & Computer Engineering

---

Ruihua Liu, Ph.D.  
Committee Member  
Associate Professor  
Electrical & Computer Engineering

---

John G. Weber, Ph.D.  
Associate Dean  
School of Engineering

---

Eddy M. Rojas, Ph.D., M.A., P.E.  
Dean  
School of Engineering

© Copyright by

Temesguen Messay-Kebede

All rights reserved

2014

## **ABSTRACT**

# **COMPUTATIONALLY EFFICIENT AND ROBUST KINEMATIC CALIBRATION METHODOLOGIES AND THEIR APPLICATION TO INDUSTRIAL ROBOTS**

Name: Messay-Kebede, Temesguen  
University of Dayton

Advisor: Dr. Raúl Ordóñez

Robot kinematic calibration is the process of enhancing the positioning accuracy of a given manipulator and must be performed after robot manufacture and assembly or during periodical maintenance. This dissertation presents new computationally efficient and robust kinematic calibration algorithms for industrial robots that make use of partial measurements. These include a calibration method that requires the supply of Cartesian coordinates of the calibration points (3DCAL) and another calibration technique that only requires the radial measurements from the calibration points to some reference (1DCAL). Neither method requires orientation measurements nor the explicit knowledge of the whereabouts of a reference frame. Contrary to most other similar works, both methods make use of a simplified version of the original Denavit-Hartenberg (DH) kinematic model. The simplified DH(-) model has not only proven to be robust and effective in calibrating industrial manipulators but it is also favored from a computational efficiency viewpoint since it consists of comparatively fewer error parameters. We present a conceptual approach

to develop a set of guidelines that need to be considered in order to properly construct the DH(-) model such that it is parameterically continuous and non-redundant. We also propose an automated method to provide a characterization of the parameters that can be insightful in identifying redundant/irrelevant parameters and deducing the DH(-) error model of a manipulator. The method is a hybrid scheme comprised of the Simulated Annealing (SA) algorithm and a local solver/optimizer and it conducts a statistical analysis on the estimates of a given error parameter that is indicative of its relevance. For the type of industrial robots used in this dissertation, we made note that calibrating the home position only is sufficient to attain adequate results for most robotics applications. Hence, we put forward for consideration of a yet simpler calibration model; the DH(-)(-) model. We employ the Trust Region (TR) method to minimize the objective functions (solve for the error parameters of the simplified error models) of both frameworks (3DCAL and 1DCAL). We also compare the performance of the proposed methods to that of a state-of-the-art commercial system (Motocal) using the same materials, data and internationally recognized performance standards. Our experimental results suggest that our methods are more robust and yield better results compared to that of MotoCal.

To my family

## **ACKNOWLEDGMENTS**

In the very first place I would like to thank my dissertation advisor Dr. Raúl Ordóñez who has offered great help and guidance throughout the completion of my degree. I am appreciative of his patience and I thank him for offering exceptional encouragement and support in various ways. I would also like to thank Dr. Russell C. Hardie for participating in my committee and, above all, for being a wonderful mentor. He has inspired and enriched my growth as a student and researcher. I also need to thank my department chair Dr. Guru Subramanyam for the financial support and for giving me the extra time to complete this thesis. Without them this thesis would not have been possible.

I also would like to thank the director of Yaskawa Motoman Robotics, Inc. Mr. Wade Hickle, Senior Engineer Mr. Eric Marcil and Chief Engineer Mr. George Sutton for their guidance and for making the much needed materials available to me. I appreciate Dr. John Loomis and Dr. Ruihua Liu for participating in my committee. Thanks go to Mr. Tom Schenck of Measurement Specialties, Inc. for allowing me to conduct experiments in his workplace. Finally, I would like to give my special thanks to my dearest friends for all their help and encouragement.

## TABLE OF CONTENTS

	<b>Page</b>
ABSTRACT . . . . .	iii
DEDICATION . . . . .	v
ACKNOWLEDGMENTS . . . . .	vi
LIST OF TABLES . . . . .	ix
LIST OF FIGURES . . . . .	xi
CHAPTER:	
1. INTRODUCTION . . . . .	1
2. MATERIALS . . . . .	13
2.1 Manipulators: The HP20D, MH5-Hi and MH6 . . . . .	13
2.2 Measurement Systems: 3D and 1D CompuGauge . . . . .	15
2.3 Coordinate Measuring Machine (CMM): Cordax RS-220 DCC . . . . .	18
3. PROPOSED KINEMATIC CALIBRATION METHODS . . . . .	20
3.1 The DH(-) and DH(-)(-) Kinematic Calibration Models: Conceptual (Analytical) Approach . . . . .	20
3.2 3DCAL . . . . .	31
3.3 1DCAL . . . . .	34
3.4 Optimization Scheme: Identifying Error Parameters . . . . .	37
3.5 The DH(-) Kinematic Calibration Model: Automated/Numerical Method . . . . .	42



4.	EXPERIMENTAL RESULTS AND DISCUSSION . . . . .	52
4.1	3DCAL Performance Analysis . . . . .	52
4.2	1DCAL Performance Analysis . . . . .	61
4.3	Discussion . . . . .	65
5.	CONCLUSION . . . . .	74
	BIBLIOGRAPHY . . . . .	77
	APPENDICES:	
A.	Geometry of the HP20D manipulator [1, 2]. . . . .	90
B.	Geometry of the MH5 manipulator [1, 3]. . . . .	91
C.	Geometry of the MH6 manipulator [2, 4]. . . . .	92
D.	Expressional details of homogeneous transformation matrices. . . . .	93
E.	Graphical User Interface (GUI). . . . .	95

## LIST OF TABLES

<b>Table</b>	<b>Page</b>
2.1 DH parameters of the HP20D manipulator. . . . .	14
2.2 DH parameters of the MH5-Hi manipulator. . . . .	14
2.3 Axis Resolution of the MH5 and MH5-Hi. . . . .	14
2.4 DH parameters of the MH6 manipulator. . . . .	15
2.5 Specifications of the 3D CompuGauge System [5]. . . . .	17
3.1 Original DH convention based continuous kinematic calibration model applied to HP20D manipulator. . . . .	23
3.2 DH(-) kinematic calibration model of the HP20D manipulator. . . . .	27
3.3 DH(-) kinematic calibration model of the MH5-Hi manipulator. . . . .	28
3.4 DH(-) kinematic calibration model for MH6 manipulator. . . . .	29
3.5 DH(-)(-) kinematic calibration models . . . . .	30
3.6 “Stage 1” using the “automated approach”: auto-relevance assessment of the various error parameters using 3DCAL and the HP20D manipulator. . . .	49
4.1 3DCAL and MotoCal “Stage 2” calibration results using 3D CompuGauge and the DH(-) error model of the HP20D manipulator. . . . .	54
4.2 3DCAL and MotoCal “Stage 2” calibration results using 3D CompuGauge and the DH(-)(-) error model (home position and tool data) of the HP20D manipulator. . . . .	55

4.3	3DCAL and MotoCal “Stage 2” calibration results using CMM and the DH(-) error model of the MH5-Hi manipulator. . . . .	57
4.4	3DCAL and MotoCal “Stage 2” calibration results using CMM and the DH(-)(-) error model (home position only) of the MH5-Hi manipulator. . .	58
4.5	Summary of the improvements in positioning accuracy using 3DCAL and MotoCal. . . . .	62
4.6	1DCAL and MotoCal “Stage 2” calibration results using 1D CompuGauge and the DH(-) error model of the MH6 manipulator. . . . .	63
4.7	1DCAL and MotoCal “Stage 2” calibration results using 1D CompuGauge and the DH(-)(-) error model (home position) of the MH6 manipulator. . . .	63
4.8	Summary of the improvements in positioning accuracy using 1DCAL and MotoCal. . . . .	65
4.9	3DCAL and MotoCal “Stage 2” calibration results using CMM and the DH(-) error model of the MH5-Hi manipulator along with the automated/numerical assessment of each parameter deemed salient. . . . .	69

## LIST OF FIGURES

Figure	Page
2.1 The 3D CompuGauge System and measurement attachment (tool) [5]. . . . .	16
2.2 1D CompuGauge System [5, 6]. . . . .	18
2.3 The Cordax RS-220 DCC Coordinate Measuring Machine (CMM). . . . .	19
3.1 Drawing of two consecutive parallel joint axes and corresponding coordinate frames assigned according to the DH convention [7, 8]. . . . .	22
3.2 Illustration of DH parameters redundancy. . . . .	25
3.3 3DCAL block diagram. The DH(-) model of the HP20D is used as an example. . . . .	33
3.4 1DCAL block diagram. In this Figure, the DH(-) model of the MH6 is used as an example . . . . .	36
3.5 Top level block diagram illustrating the product usage protocol. . . . .	43
3.6 3DCAL: TR optimization session details for a given starting point (out of the potential 50). . . . .	48
4.1 Calibration points used to calibrate the HP20D Manipulator [9]. . . . .	53
4.2 Alternate 1DCAL approach block diagram where the DH(-) model of the MH6 manipulator is used as an example. . . . .	71
A.1 Geometry of the HP20D manipulator. All dimensions are metric (mm). . . . .	90

B.1	Geometry of the MH5 manipulator. All dimensions are metric (mm and degrees). . . . .	91
C.1	Geometry of the MH6 manipulator. All dimensions are metric (mm). . . . .	92
E.1	Graphical User Interface (GUI) for the proposed calibration framework. . . . .	95

# CHAPTER 1

## INTRODUCTION

Repeatability (also known as test-retest reliability) and accuracy/precision are important characteristics of industrial robots [10–13]. The former is relevant when programming the robot by manually teaching the end-effector poses (the so called *teach and play* programming methodology). It refers to the manipulators ability to return to a taught position. However, in off-line programming, the latter is of essence because the location would not have been taught to the robot but would be specified/defined in a virtual space with respect to an absolute or relative coordinate system [10–14]. Meaning, accuracy refers to the robots' ability to move to a commanded location in its workspace. Both are measures of *pose* (position and orientation) performance. The demand of improving those two qualities/facets of industrial robots has therefore been growing continuously over the past two decades.

It is a known fact that today's industrial manipulators have satisfactory repeatability (better built) but poor accuracy due to numerous sources of errors. As listed in [12, 15–17], the errors can be caused by temperature drifts and other nonlinearities, manufacturing and assembly errors, limited resolution of the motor encoders, computer round-off errors, steady-state control errors and installation errors (including due to component replacement). Robot calibration is the process of enhancing the positioning accuracy of a manipulator

through software rather than modifying the mechanical structure (i.e., design) of the robot itself [18–20]. Different calibration techniques (or a combination of) need to be employed to solve the various types of error model. Generally speaking, the methods can be classified into two main/broad categories: geometric/kinematic calibration and non-kinematic calibration, which models errors other than geometric defaults (e.g., dynamic parameters, backlash, environmental related variability such as temperature drifts, etc.) [12, 15, 21]. The leading source of lack of accuracy is the deviation between the mathematical model of the manipulator in the controller and the actual geometry of the system [11, 12, 15, 17, 22, 23]. An accurate representation of the geometry of a robot is very crucial because the efficiency of methods for planning and controlling robot motions to perform specified tasks is dependent on such a mathematical model [20, 24–28]. Hence, in this dissertation, we present kinematic calibration techniques that make use of partial pose measurements. A kinematic calibration procedure involves modelling, measuring, identifying parameters and implementing compensation [18, 29, 30].

The first procedural step is to establish/derive a mathematical function that relates the robot joint angles to the pose of the robot end-effector and that takes into account the geometric error parameters that need to be modeled. The standard Denavit-Hartenberg (DH) convention [31] is universally accepted/used for kinematic modelling in robotics [12, 25, 27, 31–33]. However, a standard DH based kinematic calibration model is singular for manipulators with two consecutive parallel (or near parallel) joint axes (i.e., a slight deviation of the structure would yield a significant variation of the model parameters) [7, 8]. Since the majority of industrial robots possess at least two parallel joint axes, significant efforts were made to solve such a problem. Authors either proposed to use a modified version of the standard DH convention (e.g., adding an extra parameter to the original DH

model) or introduced their own model to resolve the presented challenge. For example, Hayati *et al.* [7, 8] introduced new rules to determine the link parameters of parallel or near parallel joint axes. They added an extra revolute parameter to the standard DH model and developed specific equations to handle the special case. Stone [34] derived a model and an identification approach to estimate what he refers to as “S-model” parameters, based on circle-point analysis. He used what he designated as the plane of rotation, center and radius of rotation as joint features in his approach. In their work, Veitschegger and Wu [35] presented a more detailed model that applies to consecutive parallel joints and includes second-order terms. Another example is the complete and parameterically continuous (CPC) model described in [36] and [37]. The proposed method by Abderrahim *et al.* [18, 29] combines the methodologies of Stone [34] and Hayati *et al.* [8]. Another solution is to employ the product of exponentials (POE) presented in [38–40]. POE guarantees the smooth mapping between Lie algebras and Lie groups, and hence naturally avoids singularity. However, a potential drawback of some of the discussed alternate approaches/solutions is that the compensation is not directly implementable in controllers of existing industrial manipulators. In other words, control software like the one found in Yaskawa Motoman Robotics, Inc. (e.g., [2, 3, 41]) or other similar controllers do not use the same parameters to solve the kinematic equations.

Fairly recently, Nubiola and Bonev [12, 15] proposed an elegant solution to overcome the problem of disproportion at parallel or near parallel joint axes. In their method, they forced axis 3 to be parallel to axis 2 for simplicity. Meaning, they chose not to perturb the zero valued DH parameter  $\alpha_s$  (also known as link twists [20, 24–27]) and were able to successfully calibrate a manipulator by making use of the original DH model. In this thesis, we extend that solution and propose to further “simplify” the standard DH based kinematic



calibration model so as to make it continuous and non-redundant. As done by [12, 15], we also opt to not consider the error in the zero valued DH parameter  $\alpha_s$  in order to overcome the problem of discontinuity [7, 8, 42]. We believe our approach is applicable here because our focus is to develop a calibration scheme for industrial manipulators. In particular, one that is suited for improving the accuracy of Yaskawa Motoman Robotics, Inc. or other similar manipulators. Since access to modify the DH parameter  $\alpha$  in the control software currently used within controllers such as [2, 3, 41] is not possible, we put forward for consideration to not consider the errors related to all link twist parameters (including the non-zero valued  $\alpha_s$ ). Furthermore, another problem with the error model (original DH convention based kinematic calibration model), is redundancy due to ineffectual or linearly dependent error parameters. In which case it makes it difficult to distinguish which of the error parameters are contributors to the positioning inaccuracy (i.e., which of those error parameters are relevant). Thus, we allow the end-user to select which permissible DH parameters to perturb/estimate subject to some rules/guidelines. We have established some useful guidelines based on empirical studies and generalized geometric knowledge of kinematic modelling that need to be considered in order to prevent/avoid redundancy in the error model. Our guidelines dictate that some of the DH parameters (in addition to the DH  $\alpha$  parameter) be kept “frozen” during the calibration process. The simplified DH(-) model, in addition of being continuous and non-redundant, is therefore also favored from computational efficiency viewpoint since it consists of comparatively fewer error parameters. We shall also show that it is more robust compared to existing calibration methods since it ensures that a reliable end-result is attained.

We have also devised an automated method to provide a parameter assessment that can be insightful in identifying redundant/irrelevant parameters and correctly constructing

the DH(-) model of a given manipulator. In particular, we believe that this automated assessment might be of great asset when dealing with structures of complex geometry (e.g., manipulators with higher Degree of Freedom (DOF) compared to the ones considered in this work). The method is a hybrid scheme comprised of the Simulated Annealing (SA) algorithm [43–47] and the Trust Region (TR) algorithm [48–52]. It provides a statistical analysis on the estimates of a given error parameter that is suggestive of its relevance. Rather than using a scatter-search mechanism or uniformly distributed start points within bounds to generate multiple starting points (initial conditions) as done in [53–55] to execute a Global Search (GS), we make use of SA. We then employ the TR method for convergence purposes (as a local solver/optimizer) to examine each starting point generated via SA. In other words, we utilize SA to obtain a set of starting points and analyze each point using the TR solver in order to carry out a GS. To our best knowledge, we have not seen that done in any other work. Other types of GS techniques like the ones described in [53–55] call for higher number of starting points and an exhaustive analysis of those points in an attempt to find global minima within specified bounds of solution space. Our SA based GS approach appears particularly advantageous because a comparatively fewer number but richer set of starting points are examined (i.e., it does not involve as much use of the solver in question). With a proper choice of temperature model and tuning parameters [43–47], SA can intelligently explore the solution space and procure potential/key starting points. Note that the purpose of this hybrid scheme is not to find the optimal solution (solve for the error parameters) but to provide valuable insight regarding the relevance of each error parameter and to make the task of deducing the DH(-) calibration model for an arbitrary manipulator unchallenging/effortless (i.e., a diagnostic utility if one will). Once the scheme has converged for the selected error parameters and a set of calibration data, the variance

of each solution for a given error parameter (across the multiple solutions found using our SA-TR based GS) is examined so as to provide an automated relevance/redundancy assessment of the selected parameters. Error parameters that portray high variance but no significant change in the output of the objective/cost functions are highly likely to be linearly dependent or not pertinent. Thus, for the error parameters that the auto-rating is above a given threshold, only a basic understanding of the geometry of the robot is required from the user to designate which ones are non-pertinent and hence that need to be discarded in order to correctly derive the DH(-) kinematic calibration model for the manipulator at hand. In effect, the automated evaluation alleviates the burdensome need of a comprehensive analysis of the entire model and narrows down the scope of the problem such that the user can focus on the targeted redundant/irrelevant error parameters only (i.e., it supports/assists the user in the endeavor of finding the DH(-) model of a manipulator).

Moreover, we shall also show that, for the type of robots (e.g., [1, 4, 56]) and controllers (e.g., [2, 3, 41]) that are put to use in this dissertation, precision inaccuracy is mostly due to incorrectness of the offsets values used to describe manipulator's home position (also known as ABS data [2, 3, 41]). Although in some cases estimating the other two DH parameters (link lengths and offsets [25, 27, 31]) can be helpful, we shall demonstrate that calibrating the home position of manipulators like [1, 4, 56] is sufficient to attain accuracy that is satisfactory for most robotics applications. Hence, we also further simplify the DH(-) model and introduce the DH(-)(-) model, a simpler model that consists of calibrating the home position (and tool if applicable) only. Thus, in light of those reasons, we believe it is justifiable to make use of a simplified versions of the original DH model to calibrate industrial robots such as [1, 4, 56] that are governed by control software like the ones found in [2, 3, 41]. We shall demonstrate that although incomplete, the DH(-) and DH(-)(-)

kinematic calibration models are capable of exceptional performance and can be used to calibrate a wide range of industrial robots provided that the simplified models are constructed properly.

In practice, another crucial choice (used in the second procedural step of calibration) is the measurement system. The efficiency of the calibration method is determined by the accuracy of the observations (measurements) itself [12, 15, 57, 58]. A number of different measurement systems have been used for robot calibration and/or validation. Generally speaking, the measurement systems used for the measurement step during robot calibration can be classified into two groups: *complete pose measurement* and *partial pose measurement*. A complete pose measurement of the tool pose would consist of three position coordinates, and three orientation angles (6D). This type of measurement method yields the maximum information for a given configuration (i.e., at each observation). Examples of kinematic calibration techniques that make use of complete pose measurement include those described in [15, 18, 58–64]. More specifically, [59] makes use of a coordinate measuring machine with a touch probe and a specially designed end-effector to calibrate a robot. In the work of [18, 60–62] a 6D vision/camera-based measuring systems is used to calibrate manipulators. Another example is the work described in [63], where a laser tracker with a 6D probe is put to use for calibration. A telescoping bar based 6D measurement system is used in [15, 58] for calibration purposes. We also have the work of [64], where the authors improve the accuracy of a Daewoo DR06 industrial robot using an Optotrak 6D optical measurement system. A different way of carrying out 6D measurements is presented in [65]. In that work, the orientation of an industrial robot system is calibrated using vision (referred as “major orientation error” by the authors) and the so called “relative position and minor orientation error” is calibrated using 3D force/torque sensor [65].

Incomplete/partial measurements of the tool pose (that is, less than six measured values per observation ranging from 3D to 1D) can also be used to identify the robot parameters. For example, [57] uses laser tracking interferometry to conduct experiments in robot metrology. In that experiment, the target is described in a spherical coordinate system. Additionally, [66] uses two theodolite triangulation to calibrate an Automatrix AID 900 robot. A three theodolite based triangulation system is used in [67] to calibrate a PUMA 760 robot. Two theodolites are put to use in [68] to locate a single target that is mounted on the robot to be calibrated. The position of the robot tool is computed by measuring three distances to the target rather than the angles returned by theodolites. Similar to the work of [68], [33, 69] computes the three distances but using time of flight from an acoustic emitter at the robot end-effector to microphones located at a number of measurement points. The 3D end-effector measurements required by their calibration methods described in [8, 70] are deduced using a touch probe and a reference artifact. In [71–75], the authors utilize a telescoping ballbar to acquire measurements for robot calibration processes. A vision-based 3D position measuring device with limited range is employed in [76] and [77] to calibrate a robot. A 3D vision-based measurement system is also used by [23] to calibrate an ABB IRB-2400 and a PUMA-500 at different regions and volumes within their workspace. Wider 3D range position measuring equipments that make use of laser tracking are employed in [12, 29, 78]. A radial-distance transducer is used to measure the distance of the robot endpoint from a fixed point in the workspace (1D) and to identify kinematic parameters in [79]. Different types of calibrated fixtures can also be used to measure position information during a robot calibration [32, 80–82].

What is paramount to note from the research efforts discussed thus far (type of measurements used in their methods and reported performance analyses), is that incomplete pose

information in accordance with a specific data-collection scheme can be used to successfully calibrate a robot (yields satisfactory/useful results for most industrial robotic application). Also to note from the survey is that, in most cases, complete pose measuring devices were found to be relatively more expensive (compared to partial pose measuring equipments). For these reasons, it behooves us to develop a kinematic parameter identification solution that requires the supply of incomplete pose information. Moreover, considering the variability among existing partial pose measuring devices, we believe that, in order for a solution to have practical relevance, it ought to be adaptable to the supply of various types of partial/incomplete pose information. Hence, our overall kinematic calibration framework is comprised of two methodologies/systems: one that uses 3D position measurements (3DCAL) and another that requires radial distance measurements only (1DCAL). Note that neither method requires orientation measurements. In this dissertation, we made use of Compugauge [5] and the Cordax RS-220 DCC Coordinate Measuring Machine (CMM) to acquire the partial pose measurements of the calibration points.

For the third procedural calibration step (identifying parameters), in our framework, we have formulated kinematic parameters identification process as a simplified optimization problem. In our 3DCAL and 1DCAL systems, we make use of the DH(-) and DH(-)(-) error models (simplified versions of original DH model). The 3DCAL requires the Cartesian coordinates (position only) of the robot's tool for some designated configurations. Unlike [12, 66], our 3D approach does not require knowledge of the position of the robot's base frame as the Euclidean distances among the observations themselves are the only measurements that are put into use. Aforementioned, our 1DCAL system requires the supply of radial distance measurements only (i.e., absolute distance from the tool endpoint to a fixed reference anchored outside the workspace of the robot). Our 1D method

is similar to one proposed by [79] in the sense that the objective function to be minimized is also based on the difference of the computed radial distances of the robot's endpoint with respect to a fixed reference and the measured lengths. However, we differ in that our 1D method does not require the explicit knowledge of the fixed reference whereabouts with respect to the robot's base frame. That is to say, we believe our approach is more flexible to the location of the fixed reference (compared to the approach described in [79]) because the position of the fixed reference is estimated and then iteratively refined along with the selected DH error parameters via our algorithm during the calibration process. In this study, the TR approach and the Levenberg-Marquardt method (LM) [83–85] algorithm are implemented to minimize the objective functions of both systems (3DCAL and 1DCAL). Even though both (TR and LM) algorithms converged to the same numerical values for all cases of nonlinear least square estimation of the parameters, we found TR to be more efficient compared to LM and hence the former is favored and is used to optimize over the cost functions of both systems.

For the fourth calibration procedural step (implementing compensations), we allow user interaction. It must be recalled that access to modify the parameters in some controllers is not obvious, and may not be possible for second and third parties. Hence, as mentioned in the above, we let the user select which DH parameters to estimate subject to our rules that need to be applied to successfully derive the DH(-) error model (i.e., in order to render the original DH convention based error model continuous and non-redundant). It is important for the compensation of a kinematic calibration technique to be practical so that it is transferable to control software of existing industrial manipulators [18]. From that perspective (i.e., considering the widespread use of the convention in existing robot control

software), we believe our approach is well-suited/desirable for this procedure since DH parameters are obtained directly from both methods (3DCAL and 1DCAL).

It is a difficult task to make definitive comparison between previously published calibration systems due to variability in the number of calibration points used to calibrate, number of points used to validate performance of calibration scheme, type of manipulator, number and type of parameters that are identified, kinematic model, accuracy of the measurement equipment, performance metric, etc. It is well known that the performance results of a calibration system can differ significantly depending on those variables. Fortunately, a comprehensive comparison is made possible here thanks to Yaskawa Motoman Robotics, Inc. and Measurement Specialties, Inc., who allowed us to make use of the Cordax RS-220 DCC Coordinate Measuring Machine (CMM), CompuGauge [5] and MotoCal [6]; commercial systems that are currently used in their setting to calibrate various manipulators. In this dissertation, our experimental results are compared to those of MotoCal [6]. We have used identical manipulators, measurement equipment, performance metric and similar data to put our performance into clear context. We shall show that our calibration methods are not only competitive with a state-of-the-art system of the same genre but more robust.

The remainder of this dissertation is organized as follows. We begin by describing the materials that we used in Chapter 2. They include manipulators and measurement equipments used to carry out numerous experiments. In Chapter 3, we present two reliable techniques that can be used to properly construct the DH(-) and DH(-)(-) error models. The 3DCAL, 1DCAL and the optimization scheme algorithms are also described in that Chapter. Experimental results and related discussion are presented in Chapter 4. The results include performance results and comparison of our approaches to that of a commercial system (MotoCal) using the same materials. Finally, in Chapter 5, we offer conclusions.



Where relevant, some of the previously published methods described in this Chapter are discussed further in the thesis.

## CHAPTER 2

### MATERIALS

#### 2.1 Manipulators: The HP20D, MH5-Hi and MH6

To test our 3DCAL system, we made use of the Yaskawa Motoman Robotics Inc. HP20D manipulator [1] and the control software of the DX100 controller [2]. The HP20D is a versatile high speed 6 DOF robot with all six revolute joints [1]. It is an anthropomorphic robot comprised of a spherical wrist [20,25,27]. The HP20D is used in a variety of applications such as cutting, dispensing, handling, machine tending, material removal, packaging, press tending and welding [1]. Details concerning the specifications of the robot and controller are presented in [1]. The geometry of the manipulator is presented in Appendix A. The DH Table according to the convention described in [25] is presented in Table 2.1.

We also made use of the MH5-Hi [56] and the FS100 controller [3] to further test our 3DCAL approach. The MH5 can be used for similar applications as the HP20D. Note that it offers the widest work envelope in it's class (706 mms reach) [56]. Technical data of the MH5 are provided in [3, 56]. Its geometry is shown in Appendix B and the corresponding DH Table is provided in Table 2.2. The "Hi" came to be incorporated in the name of the robot because it was rebuilt such that to increase its axes resolution (i.e, the gears of the

Table 2.1: DH parameters of the HP20D manipulator.

<b>Link</b>	<b><math>d_i</math> in mm</b>	<b><math>a_i</math> in mm</b>	<b><math>\alpha_i</math> in degrees</b>	<b><math>\theta_i</math> in degrees</b>
1	0	150	90	$\theta_1$
2	0	760	0	$\theta_2 + 90$
3	0	140	90	$\theta_3$
4	795	0	-90	$\theta_4$
5	0	0	90	$\theta_5 - 90$
6	105	0	0	$\theta_6$

Table 2.2: DH parameters of the MH5-Hi manipulator.

<b>Link</b>	<b><math>d_i</math> in mm</b>	<b><math>a_i</math> in mm</b>	<b><math>\alpha_i</math> in degrees</b>	<b><math>\theta_i</math> in degrees</b>
1	0	88	90	$\theta_1$
2	0	310	0	$\theta_2 + 90$
3	0	40	90	$\theta_3$
4	305	0	-90	$\theta_4$
5	0	0	90	$\theta_5$
6	80	0	0	$\theta_6$

Table 2.3: Axis Resolution of the MH5 and MH5-Hi.

<b>Joint/Axis Index</b>	<b>Manufacturer MH5 in pulses/rev</b>	<b>Rebuilt MH5-Hi in pulses/rev</b>
1	391,791.3043	1,253,732.1739
2	436,906.6666	873,813.3333
3	364,088.8888	728,177.7777
4	327,680.0000	1,048,576.0000
5	327,680.0000	491,520.0000
6	204,800	409,600.0000

manufacturer were replaced with finer components). This is done in order to effectively address one of the causes of error discussed in Chapter 1 (“limited resolution of the motor

encoders”) and hence to further increase positioning accuracy from a mechanical/structural viewpoint. Table 2.3 shows the axis resolution of the MH5 (robot from manufacturer) and the one for the MH5-Hi (newly rebuilt) in terms of “pulses” per “revolution”, where “pulse” is a motor’s smallest revolutionary increment.

The MH6 is a similar 6DOF serial robot that is used for alike applications as the HP20D [4]. We made use of the MH6 manipulator and DX100 controller to validate our 1DCAL method [2, 4]. The dimensions of the robot are depicted in Appendix C. Specifications of the MH6 robot can be found in [2, 4]. Its DH Table is shown in Table 2.4.

Table 2.4: DH parameters of the MH6 manipulator.

<b>Link</b>	$d_i$ <b>in mm</b>	$a_i$ <b>in mm</b>	$\alpha_i$ <b>in degrees</b>	$\theta_i$ <b>in degrees</b>
1	0	150	90	$\theta_1$
2	0	614	0	$\theta_2 + 90$
3	0	155	90	$\theta_3$
4	640	0	-90	$\theta_4$
5	0	0	90	$\theta_5 - 90$
6	95	0	0	$\theta_6$

## 2.2 Measurement Systems: 3D and 1D CompuGauge

For the purpose of measuring the Cartesian coordinates of the calibration points for our 3DCAL system using the HP20D robot, we made use of 3D CompuGauge [5]. CompuGauge is a system that allows 3D measurement and performance testing of robot positioning. It is comprised of hardware and software.

As shown in Figure 2.1, the hardware (measurement device itself) consists of two triangulation beams. The “measurement attachment” (tool) depicted/zoomed-in in that

figure is mounted on the robot. The four “measurement cables” coming from both triangulation beams are connected to the tool (to the robot test point). High resolution, low inertia optical encoders are used to constantly measure the extension of the cables. The data is interpreted by the software to deduce an accurate measurement of the test point with respect to the user-supplied/user-defined reference frame [5]. Note that only three out of the four cables that originate from the fixed cable-fed device are put to use to calculate the coordinates of a given test point. The extra cable is dedicated for verification purposes.

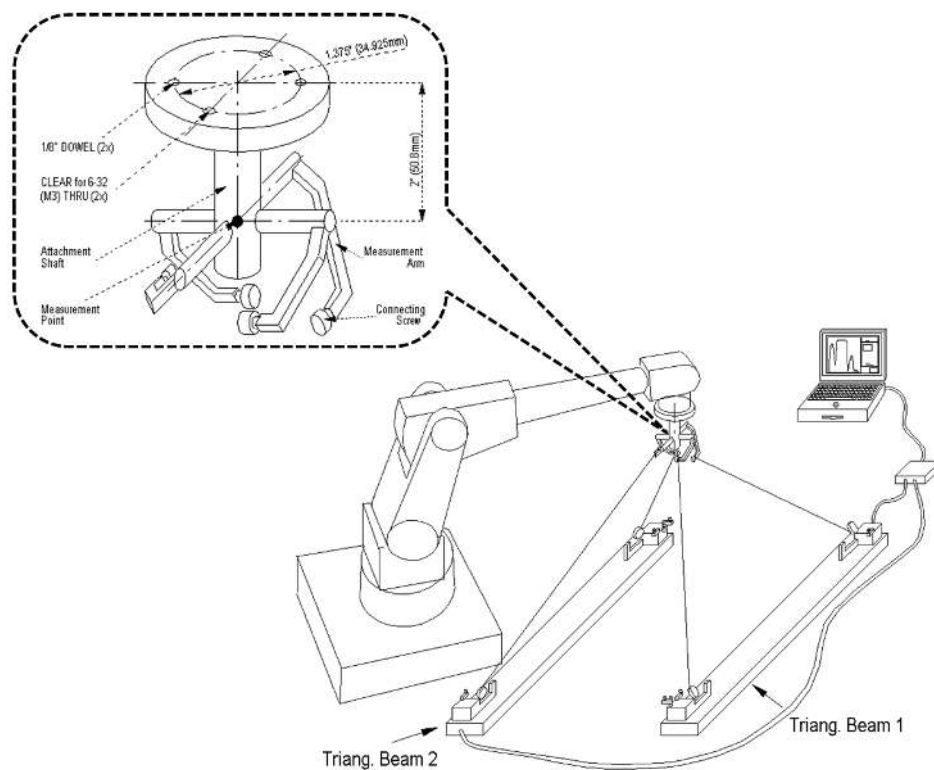


Figure 2.1: The 3D CompuGauge System and measurement attachment (tool) [5].

In addition to data acquisition, the software also allows to perform various robot performance tests following two different internationally recognized performance standards: “Manipulating Industrial Robots - Performance Criteria and Related Testing Methods” (ISO 9283) and “American National Standard for the Evaluation of Point-to-Point and Static Performance Characteristics of Industrial Robots and Robot Systems” (ANSI/RIA R15.05) [5]. In this dissertation, we make use of that software capability to measure and report our performance. More will be said about the performance metric and analysis in Chapter 4. Specifications of the discussed measurement system are presented in Table 2.5.

Table 2.5: Specifications of the 3D CompuGauge System [5].

<b>Performance</b>	
Resolution:	0.010 mm
Repeatability:	0.020 mm
Accuracy:	$\pm 0.150$ mm
Measurement space:	1500 mm $\times$ 1500 mm $\times$ 1500 mm
Tracking rate:	up to 5 m/s
Sampling frequency:	from 25 to 1000 Hz
Data range:	up to 32000 points/file
Measurement Time:	up to 10 min
<b>Mechanical (per “triangulation beam”)</b>	
Dimensions (approx.) :	(L) 1730 mm $\times$ (W) 165 mm $\times$ (H) 165 mm
Weight (approx.):	7 kg
<b>Environmental</b>	
Operating temperature:	20 $\pm$ 2°C (68 $\pm$ 5)

In order to acquire the radial measurements of calibration points for our 1DCAL system, we make use of 1D CompuGauge. 1D CompuGauge consists of an encoder, a specially designed end-effector that is to be mounted on the robot and software. In essence, it is a simplified version of 3D CompuGauge. The encoder is be anchored outside of the manipulator’ workspace; preferably 45 cm to one side of the robot base and approximately

the same height as the base as shown in Figure 2.2. This location permits the largest working envelope, reduces interference with robot components, and increases calibration accuracy. The software provides similar capabilities as the one of 3D CompuGauge; a means for data acquisition and robot performance testing/analysis based on alike standards.

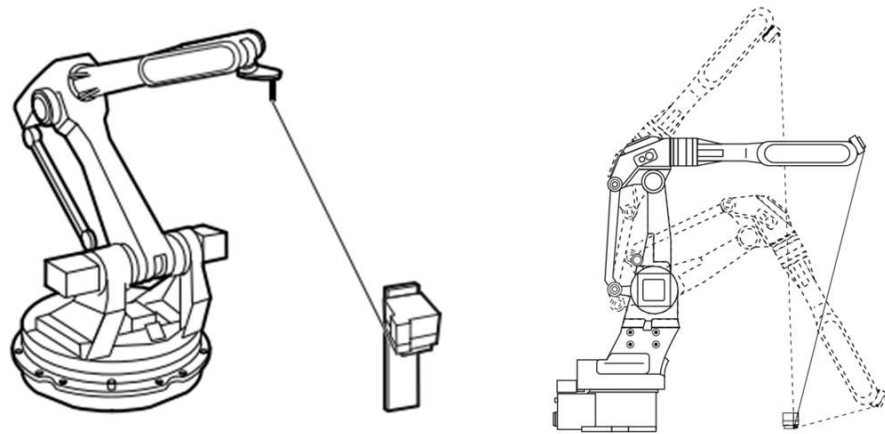


Figure 2.2: 1D CompuGauge System [5, 6].

### 2.3 Coordinate Measuring Machine (CMM): Cordax RS-220 DCC

In a final test of our 3D approach we made use of a Coordinate Measuring Machine (CMM). CMM is highly accurate device typically used for measuring the geometrical characteristics of a specimen. Figure 2.3 The Cordax RS-220 DCC CMM that we used to measure the Cartesian coordinates of the calibration points. In our experiment, the MH5-Hi is installed on the CMM machine and the measurements are defined by virtue of a specially designed end-effector that is mounted on the robot and a laser probe attached to the third moving axis of the CMM machine. The coordinates of the calibration points are collected by using the laser probe that is positioned automatically via Direct Computer



Figure 2.3: The Cordax RS-220 DCC Coordinate Measuring Machine (CMM).

Control (DCC). The advantage of using DCC based CMM machine for the purpose of taking measurements and evaluating calibration algorithms is that it can be programmed to repeatedly measure identical calibration points. The CMM system that we utilized has the following specifications: An average volumetric accuracy in the order  $0.016 \pm 0.030$  mms, repeatability in the order of  $0.013 \pm 0.008$  mms and measurement space capacity of  $3000 \text{ mm} \times 1200 \text{ mm} \times 1000 \text{ mm}$ .



## CHAPTER 3

### PROPOSED KINEMATIC CALIBRATION METHODS

In this Chapter, we describe the proposed calibration methods. The concepts used to develop a set of guidelines that need to be considered/applied in order to properly construct the DH(-) and DH(-)(-) kinematic calibration models are discussed in Chapter 3.1. 3DCAL is described in Chapter 3.2. 1DCAL is presented in Chapter 3.3. In Chapter 3.4, two optimization techniques for the purpose of minimizing the cost functions of 3DCAL and 1DCAL (identifying the error parameters of the simplified/non-redundant kinematic calibration models [DH(-) and DH(-)(-) models]) are compared. Following that, an automated/numerical parametric relevance assessment technique that can greatly aid the end-user in deducing the DH(-) model of a given manipulator is presented in Chapter 3.5. That is accomplished by making use of a novel global optimization scheme. In that Chapter, we also correlate/compare the results of conceptual/analytic and numerical approaches in order to demonstrate the agreement among the two in finding the simplified DH(-) model.

#### **3.1 The DH(-) and DH(-)(-) Kinematic Calibration Models: Conceptual (Analytical) Approach**

A kinematic calibration model is a mathematical function that relates the robot joint angles to the pose of the robot end-effector and that takes into account the geometric error parameters that need to be modeled [20, 24–27]. As told in Chapter 1, the standard/

original DH model is currently and widely used in robotics and hence is our first candidate under consideration. Hayati [7, 8] first made note that when the geometric structures deviated slightly, the original DH model parameters varied significantly (i.e., parameters don't vary continuously.) More specifically a small misalignment of parallel joint axes which corresponds to error in nominally zero valued DH parameter  $\alpha$  causes a large change in the other DH parameters [7, 8].

Figure 3.1 illustrates a schematic of two consecutive parallel joint axes and corresponding coordinate frames assigned according to the DH convention described in [25]. Looking at joint axes labelled  $\theta_{i-1}$  and  $\theta_i$ , we have two consecutive perfectly parallel joint axes. For that scenario, the DH rule dictates to locate the origin of the coordinate frame that is to be assigned to  $link_i$  (frame 1) where the common normal to  $Z_i$  and  $Z_{i-1}$  intersect  $Z_i$ . The DH parameters that relate/describe the two frames (frame 0 and frame 1) are  $d_i = 0$ ,  $a_i = L$ ,  $\alpha_i = 0^\circ$  and the joint variable offset  $\theta_i = 0^\circ$ . Now suppose that due to some manufacturing tolerances the  $Z_i$  axis is misaligned by a small angle  $\beta$  as shown in Figure 3.1. In that Figure, we labelled the misaligned axis of actuation which corresponds to  $\theta'_i$  as  $Z'_i$ .  $Z_{i-1}$  and  $Z'_i$  now intersect at some distance away from frame 0. In such case, according to the DH convention, one is to locate the origin of the coordinate frame that is to be assigned to  $link_i$  at that intersection. Therefore, the true DH parameters in this case (the ones that relate frames 0 and frame 1') are  $d_i = -D$ ,  $a_i = 0$ ,  $\alpha_i = -\beta^\circ$  and  $\theta_i = -90^\circ$ , where  $D$  as shown in Figure 3.1 is a large positive scalar which is measured from origin of frame 0 to that of frame 1'. Thus a small error in the alignment of axes of actuation that are parallel causes a large error in the other three DH parameters ( $d_i$ ,  $a_i$ , and  $\theta_i$ ) [7, 8].

Notwithstanding that, Nubiola and Bonev [12, 15] recently proposed an elegant solution to overcome the problem of disproportion at parallel or near parallel joint axes. In their

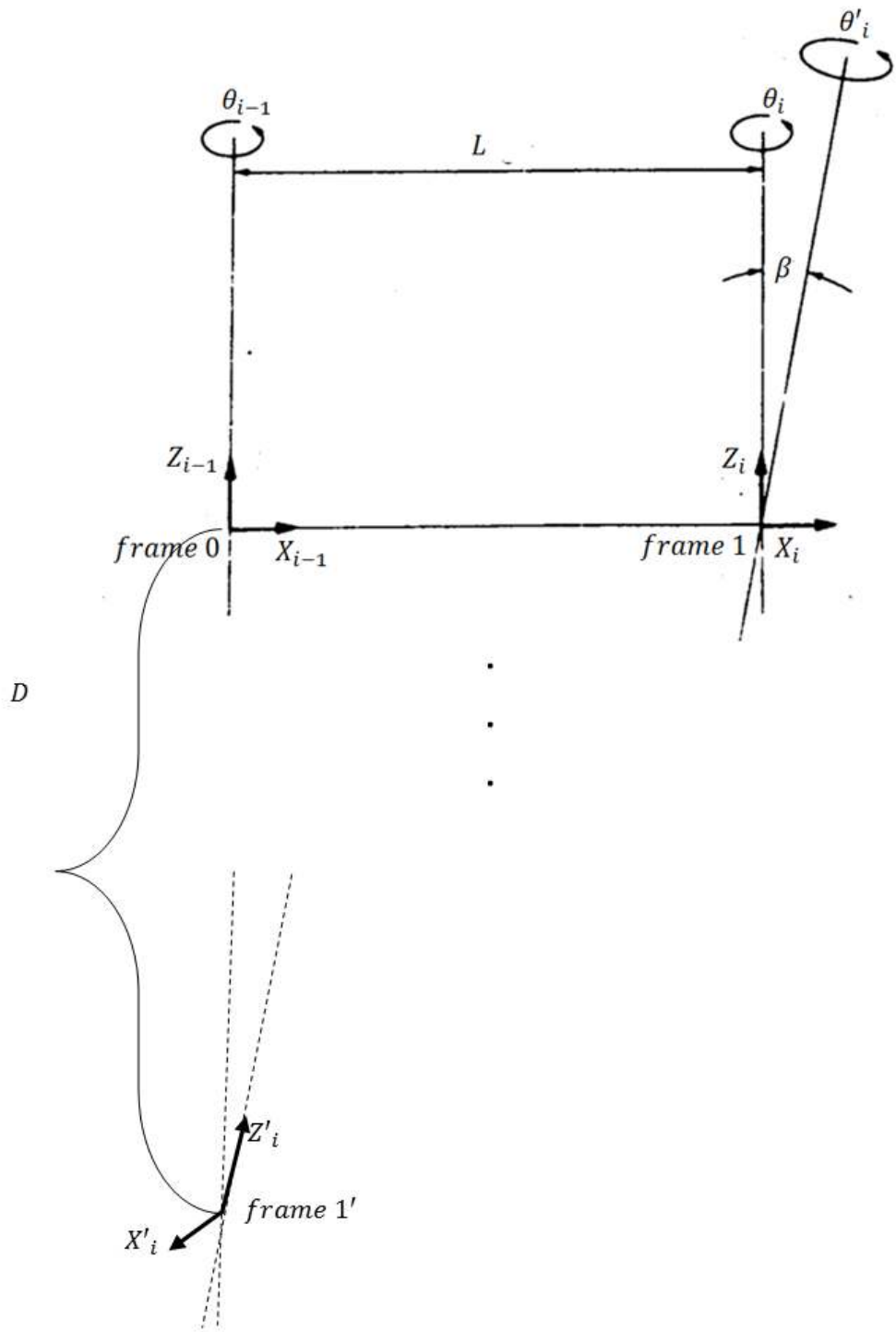


Figure 3.1: Drawing of two consecutive parallel joint axes and corresponding coordinate frames assigned according to the DH convention [7, 8].

method, they forced axis 3 to be parallel to axis 2 for simplicity (i.e., they chose not to perturb the zero valued DH parameter  $\alpha$ ) and were able to successfully calibrate a manipulator by making use of the DH model. In this work, we apply that solution and make use of an original DH convention based kinematic calibration model that meets the “Continuity” criterion described in [39, 42] to calibrate various manipulators. Table 3.1 presents the continuous kinematic calibration model applied to the HP20D manipulator.

Table 3.1: Original DH convention based continuous kinematic calibration model applied to HP20D manipulator.

<b>Link</b>	<b><math>d_i</math> in mm</b>	<b><math>a_i</math> in mm</b>	<b><math>\alpha_i</math> in degrees</b>	<b><math>\theta_i</math> in degrees</b>
1	$0 + \delta d_1$	$150 + \delta a_1$	$90 + \delta \alpha_1$	$\theta_1 + \delta \theta_1$
2	$0 + \delta d_2$	$760 + \delta a_2$	0	$(\theta_2 + 90) + \delta \theta_2$
3	$0 + \delta d_3$	$140 + \delta a_3$	$90 + \delta \alpha_3$	$\theta_3 + \delta \theta_3$
4	$795 + \delta d_4$	$0 + \delta a_4$	$-90 + \delta \alpha_4$	$\theta_4 + \delta \theta_4$
5	$0 + \delta d_6$	$0 + \delta a_5$	$90 + \delta \alpha_5$	$(\theta_5 - 90) + \delta \theta_5$
6	$105 + \delta d_6$	$0 + \delta a_6$	0	$\theta_6 + \delta \theta_6$

The HP20D robot’s kinematic calibration model can be mathematically represented as

$$\mathbf{T}(\mathbf{q})_{\text{tool}}^0 = \mathbf{A}_1(q_1)\mathbf{A}_2(q_2)\mathbf{A}_3(q_3)\mathbf{A}_4(q_4)\mathbf{A}_5(q_5)\mathbf{A}_6(q_6)\mathbf{A}_{\text{tool}}, \quad (3.1)$$

where  $\mathbf{q}$  is the vector of the joint values ( $[q_1, q_2, \dots, q_6]$  which in our case =  $[\theta_1, \dots, \theta_6]$  since we are dealing with 6DOF manipulators that possess axes with all revolute joints).

$\mathbf{A}_i$  and  $\mathbf{A}_{\text{tool}}$  can be calculated using Eqs. (3.2) and (3.3) respectively,

$$\mathbf{A}_i = \text{Rot}(z, \theta_i + \delta \theta) \text{Trans}(z, d_i + \delta d_i) \text{Trans}(x, a_i + \delta a_i) \text{Rot}(x, \alpha_i + \delta \alpha_i), \quad (3.2)$$

$$\mathbf{A}_{\text{tool}} = \text{Trans}(x, X_{\text{tool}} + \delta X_{\text{tool}}) \text{Trans}(y, Y_{\text{tool}} + \delta Y_{\text{tool}}) \text{Trans}(z, Z_{\text{tool}} + \delta Z_{\text{tool}}) \mathbf{R}_{\text{tool}}, \quad (3.3)$$

where

$$\mathbf{R}_{\text{tool}} = RPY(RX_{\text{tool}} + \delta RX_{\text{tool}}, RY_{\text{tool}} + \delta RY_{\text{tool}}, RZ_{\text{tool}} + \delta RZ_{\text{tool}}). \quad (3.4)$$

$RPY$  in Eq. (3.4) is a function that, given the Roll/ $RZ_{\text{tool}}$ , Pitch/ $RY_{\text{tool}}$  and Yaw/ $RX_{\text{tool}}$  angles of the end-effector mounted on manipulator, returns a  $4 \times 4$  homogenous transformation matrix  $\mathbf{R}_{\text{tool}}$  representative of the orientation of the tool frame [24]. See Appendix D for details of the abbreviated homogeneous transformation matrices. Note that the DH calibration model is comprised of 28 error deviations/parameters (where six of those are dedicated for estimating the location/position and orientation of the tool-frame/end-effector itself). Let  $\mathbf{p} = [\mathbf{x}, \mathbf{y}, \mathbf{z}]^T$  denote the calculated position of the end-effector. In other words, the vector  $\mathbf{p}$  contains the first three elements of the last column of  $\mathbf{T}(\mathbf{q})_{\text{tool}}^0$  for some configuration  $\mathbf{q}$ . The robot's DH kinematic calibration model can be expressed in a compact form as

$$\mathbf{p} = f(\mathbf{q}, \mathbf{e}), \quad (3.5)$$

where vector  $\mathbf{e} \in \Omega$  contains all 28 geometric error parameters (deviations  $\delta d_i$ ,  $\delta a_i$ ,  $\delta \alpha_i$  and  $\delta \theta_i$  that are added to corresponding nominal values) and  $\Omega$  is the set of allowable deviations (e.g.,  $e_i \in [-\underline{e}_i, \bar{e}_i]$  where  $-\underline{e}_i$  and  $\bar{e}_i$  are robot manufacturer dependent range limits).

We believe that forcing the alignment of parallel or near parallel joint axes as done by Nubiola and Bonev [12, 15] is well applicable in our case because access to modify the DH parameter  $\alpha$  in Yaskawa Motoman Robotics, Inc. controllers is not possible [2, 3, 41]. Hence, in addition to that, we extend the idea and elect to “freeze” all DH  $\alpha$  parameters (i.e., making use of the nominal values) in our calibration scheme.

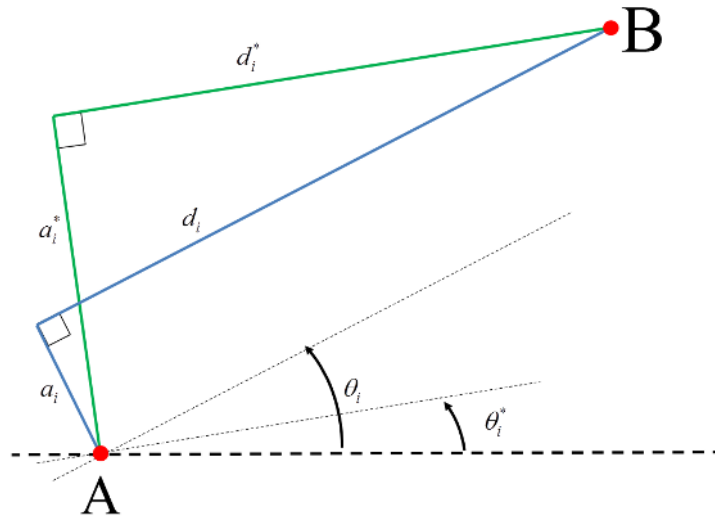


Figure 3.2: Illustration of DH parameters redundancy.

Moreover, by examining the rank of the Jacobian of their least-square optimization method, Nubiola and Bonev [12, 15] found that several error parameters are often redundant. In order to attain a stable solution, they chose to take out what they refer to as useless/redundant error parameters [12, 15]. For example, the error parameters  $\delta d_2$  and  $\delta d_3$  are dependent (i.e., they are redundant error parameters) because of the parallel axes of joints 2 and 3 (which are forced to be perfectly parallel) and so they chose to not consider  $\delta d_3$  in their model [12, 15]. Another form of redundancy that can be problematic in achieving convergence (stable solution) for an optimization scheme like the one we will be employing can be demonstrated using the example shown in Figure 3.2. The Figure shows that different combinations of the other three parameters  $\theta_i$ ,  $a_i$ , and  $d_i$  lead to the same transformation from point  $A$  to  $B$ .

In order to circumvent the discussed issues, our product usage guidelines include the following terms. Since axis 2 and 3 of the HP20D are considered perfectly parallel, only

one (out of the two) DH parameter that describes an offset along that dimension (e.g.,  $\delta d_2$  or  $\delta d_3$ ) is to be considered in the model. Moreover, for the reason explained in the above, one of the two DH parameters ( $a_i$  or  $d_i$ ) of a transformation that describes the spatial relation of any two given links (i.e.,  $A_i$ ) must be kept “frozen” at its nominal value. In fact, in this work, we will not be considering zero valued DH parameters ( $a_i$  and  $d_i$ ) in our kinematic calibration model. The reason for that lies in a long standing experience (achieved over the course of calibrating numerous Yaskawa Motoman Robotics, Inc. manipulators) and extensive experiment done by us, that including those particular parameters does not contribute to a noteworthy overall performance gain in return.

Furthermore, we will not be considering  $\delta\theta_1$ . As mentioned in Chapter 1, our approach is invariant to the fixed frames (robot’s and measurement equipment’s base/reference frames). The Euclidean distances between/among the calibration points are the measurements put to use. Hence, estimating that particular error parameter will only translate and/or rotate all calibration points while the Euclidean distances among those observations remain unchanged. Since that will bring about undesired redundancy, as it will be demonstrated by an automated parameter assessment method in Chapter 3.5, in our approach the offset of the first joint angle  $\theta_1$  must be kept “frozen” during calibration as well.

Finally, another aspect of kinematic calibration is the need to determine the tool transform when necessary. The provided measurement-attachment/tool shown in Figure 2.1 is well defined and ideally is to be mounted directly onto the robot. Unfortunately, for the HP20D, we used a fabricated mounting plate due to hardware incompatibility and hence the error parameters of  $A_{\text{tool}}$  in Eq. (3.3) need to be identified. More will be said about the experimental procedure that we performed to obtain an initial tool definition in Chapter 4. What needs to be emphasized here is (after an initial guess of the tool specifications) the

guidelines that need to be set in order to attain a good estimate (i.e., avoid redundancy). It follows that, if the attached tool is well defined, then error parameters listed in the last/sixth row of Table 3.1 ( $\delta d_6$  and  $\delta \theta_6$ ) can be estimated. However, if one chooses to estimate tool definition (as done in this work for the HP20D robot) then those error parameters need to be kept frozen for the obvious reason to avoid redundancy. This will also be justified and further discussed in Chapters 3.5 and 4. In the case of the HP20D, we made use of a flat fabricated mounting plate (approximately 20mm thick), one that guarantees certainty of the orientation of the tool frame and hence we decided to estimate the position of the tool frame only ( $X_{tool}$ ,  $Y_{tool}$  and  $Z_{tool}$ ). Let “DH(-) model” denote the kinematic calibration model after modifying the model described in Table 3.1 as discussed thus far (i.e., after applying the discussed guidelines). Table 3.2 presents the DH(-) kinematic calibration model for the HP20D manipulator that we will use to evaluate our 3DCAL methodology.

Table 3.2: DH(-) kinematic calibration model of the HP20D manipulator.

<b>Link</b>	<b><math>d_i</math> in mm</b>	<b><math>a_i</math> in mm</b>	<b><math>\alpha_i</math> in degrees</b>	<b><math>\theta_i</math> in degrees</b>
1	0	$150 + \delta a_1$	90	$\theta_1$
2	0	$760 + \delta a_2$	0	$(\theta_2 + 90) + \delta \theta_2$
3	0	$140 + \delta a_3$	90	$\theta_3 + \delta \theta_3$
4	$795 + \delta d_4$	0	-90	$\theta_4 + \delta \theta_4$
5	0	0	90	$(\theta_5 - 90) + \delta \theta_5$
6	105	0	0	$\theta_6$

The “simplified” model is now comprised of only 11 error parameters (where 3 parameters are dedicated for estimating the location of the tool frame). Note that although our kinematic calibration model is incomplete (i.e., does not meet the “completeness” criteria defined in [42, 86]), we shall show that it is capable of exceptional performance (i.e.,



yields accuracy that is more than satisfactory for most robotics applications). More will be said about this in Chapter 4. Also to note is that the DH(-) model is without a doubt favorable (compared to that of DH or other complex models discussed in Chapter 1) from a computational efficiency stand point of view since it consists of fewer degrees of error parameters.

Table 3.3: DH(-) kinematic calibration model of the MH5-Hi manipulator.

<b>Link</b>	$d_i$ <b>in mm</b>	$a_i$ <b>in mm</b>	$\alpha_i$ <b>in degrees</b>	$\theta_i$ <b>in degrees</b>
1	0	$88 + \delta a_1$	90	$\theta_1$
2	0	$310 + \delta a_2$	0	$(\theta_2 + 90) + \delta \theta_2$
3	0	$40 + \delta a_3$	90	$\theta_3 + \delta \theta_3$
4	$605 + \delta d_4$	0	-90	$\theta_4 + \delta \theta_4$
5	0	0	90	$\theta_5 + \delta \theta_5$
6	$80 + \delta d_6$	0	0	$\theta_6 + \delta \theta_6$

Similarly the kinematic calibration model of the MH5-Hi manipulator described in Chapter 2.1 can be simplified according to the guidelines mentioned in the above. Table 3.3 presents the continuous and non-redundant DH(-) error model of the MH5-Hi. Note that, in the case of validating our 3DCAL approach using the DH(-) model of the MH5-Hi manipulator, we made use of the CMM machine (described in Chapter 2.3) and an end-effector/tool of known dimensions to measure the Cartesian coordinates of the calibration points. Hence, unlike in the case of the HP20D, we opt to estimate the error parameters  $\delta d_6$ , and  $\theta_6$  instead of calibrating tool related parameters. In our experiment, we are able to accurately calculate  $\mathbf{A}_{\text{tool}}$  using

$$\mathbf{A}_{\text{tool}} = \text{Trans}(x, 0.000)\text{Trans}(y, 100.000)\text{Trans}(z, -17.78)\mathbf{R}_{\text{tool}}, \quad (3.6)$$

where  $\mathbf{R}_{\text{tool}}$  is a  $3 \times 3$  identity matrix. All dimensions in that equation are metric (mm). Note that the dimensions of the tool are deduced using the CMM machine itself in a separate experiment. Also to note is that the kinematic calibration model of the MH5-Hi is comprised of 10 parameters.

Table 3.4: DH(-) kinematic calibration model for MH6 manipulator.

<b>Link</b>	<b><math>d_i</math> in mm</b>	<b><math>a_i</math> in mm</b>	<b><math>\alpha_i</math> in degrees</b>	<b><math>\theta_i</math> in degrees</b>
1	0	$150 + \delta a_1$	90	$\theta_1$
2	0	$614 + \delta a_2$	0	$(\theta_2 + 90) + \delta \theta_2$
3	0	$155 + \delta a_3$	90	$\theta_3 + \delta \theta_3$
4	$640 + \delta d_4$	0	-90	$\theta_4 + \delta \theta_4$
5	0	0	90	$(\theta_5 - 90) + \delta \theta_5$
6	$95 + \delta d_6$	0	0	$\theta_6 + \delta \theta_6$

In order to validate our 1DCAL approach we made use of the DH(-) model of the MH6 manipulator (described in Chapter 2.1) and the 1D CompuGauge measurement equipment that is described in Chapter 2.2. The continuous and non-redundant DH(-) model for the MH6 manipulator can be constructed by applying the same guidelines as for the other two manipulators (HP20D and MH5-Hi) and is presented in Table 3.4. As done in the case of the MH5-Hi manipulator, we have also decided to calibrate  $\theta_6$  and  $d_6$  because we made use of a well defined tool that we directly mounted on the robot (i.e., in the case of employing 1DCAL to calibrate the MH6 we do not take into account the error parameters related to tool frame/data).  $\mathbf{A}_{\text{tool}}$  can be accurately calculated using

$$\mathbf{A}_{\text{tool}} = \text{Trans}(x, 0.951)\text{Trans}(y, -149.920)\text{Trans}(z, 138.040)\mathbf{R}_{\text{tool}}, \quad (3.7)$$

where  $\mathbf{R}_{\text{tool}}$  is a  $3 \times 3$  identity matrix. All dimensions in that equation are metric (mm). Note that the DH(-) model of the MH6 consists of only 10 error parameters (just like the DH(-) model of the MH5-Hi manipulator).

Table 3.5: DH(-)(-) kinematic calibration models

<b>The DH(-)(-) kinematic calibration model of the HP20D Manipulator</b>				
<b>Link</b>	<b><math>d_i</math> in mm</b>	<b><math>a_i</math> in mm</b>	<b><math>\alpha_i</math> in degrees</b>	<b><math>\theta_i</math> in degrees</b>
1	0	150	90	$\theta_1$
2	0	760	0	$(\theta_2 + 90) + \delta\theta_2$
3	0	140	90	$\theta_3 + \delta\theta_3$
4	795	0	-90	$\theta_4 + \delta\theta_4$
5	0	0	90	$(\theta_5 - 90) + \delta\theta_5$
6	105	0	0	$\theta_6$

<b>The DH(-)(-) kinematic calibration model of the MH5-Hi Manipulator</b>				
<b>Link</b>	<b><math>d_i</math> in mm</b>	<b><math>a_i</math> in mm</b>	<b><math>\alpha_i</math> in degrees</b>	<b><math>\theta_i</math> in degrees</b>
1	0	88	90	$\theta_1$
2	0	310	0	$(\theta_2 + 90) + \delta\theta_2$
3	0	40	90	$\theta_3 + \delta\theta_3$
4	305	0	-90	$\theta_4 + \delta\theta_4$
5	0	0	90	$\theta_5 + \delta\theta_5$
6	80	0	0	$\theta_6 + \delta\theta_6$

<b>The DH(-)(-) kinematic calibration model of the MH6 Manipulator</b>				
<b>Link</b>	<b><math>d_i</math> in mm</b>	<b><math>a_i</math> in mm</b>	<b><math>\alpha_i</math> in degrees</b>	<b><math>\theta_i</math> in degrees</b>
1	0	150	90	$\theta_1$
2	0	614	0	$(\theta_2 + 90) + \delta\theta_2$
3	0	155	90	$\theta_3 + \delta\theta_3$
4	640	0	-90	$\theta_4 + \delta\theta_4$
5	0	0	90	$(\theta_5 - 90) + \delta\theta_5$
6	95	0	0	$\theta_6 + \delta\theta_6$

As briefly mentioned in Chapter 1 and as it will be demonstrated in Chapter 4, when it comes to industrial manipulators like the ones of Yaskawa Motoman Robotics, Inc. and alike a great amount of positioning inaccuracy is mostly due to erroneous offsets values

used to describe manipulator's home position (also known as ABS data in controllers found in [2, 3, 41]). We therefore further simplify the DH(-) model and introduce the DH(-)(-) model where the model takes into account the home position (and tool if applicable) only. Based off the logic deliberated in the passage above concerning the unstable parameter  $\delta\theta_1$ , that error parameter shall not be taken into account in this simpler error model. Without a doubt, the newly introduced model is highly incomplete (according to [42, 86] and compared to the DH(-) model) but to our pleasant surprise it yields/produces results that we believe are significant for a majority of robotics applications. Table 3.5 presents the DH(-)(-) kinematic calibration models for the three manipulators (HP20D, MH5-Hi and the MH6 respectively). It should be noted that the DH(-)(-) error model of the HP20D is a function of 7 error parameters (where 3 are tool frame related estimates since we chose to calibrate the tool of vaguely known dimensions). In contrast, the two DH(-)(-) models of the MH5-Hi and MH6 are functions of only 5 error parameters because in our experiments we chose not to calibrate the tool in the case of those two robots (i.e., since the specification of the tools that are directly mounted on those robots is known). Also to note is that these models with notable fewer parameters can be processed a lot quicker compared to any other error model cited and proposed/DH(-) thus far (i.e., they are more preferred for the mathematical calculation that needs to be executed).

## 3.2 3DCAL

In this Chapter, we describe the 3DCAL technique that is employed to identify the error parameters of the DH(-) and DH(-)(-) models of the HP20D and MH5-Hi manipulators. Figure 3.3 shows a block diagram of the calibration algorithm where the HP20D manipulator and its DH(-) error model are used as an example. A natural extension of Eq. (3.5)

described in Chapter 3.1 is

$$\mathbf{P} = f(\mathbf{Q}, \mathbf{e}), \quad (3.8)$$

where the matrix  $\mathbf{P}$  is now comprised of a set of calculated Cartesian coordinate vectors  $[\mathbf{p}_1(\mathbf{q}_1, \mathbf{e}), \mathbf{p}_2(\mathbf{q}_2, \mathbf{e}), \dots, \mathbf{p}_N(\mathbf{q}_N, \mathbf{e})]$  of  $N$  calibration points, and matrix  $\mathbf{Q}$  contains the corresponding set of joint values (i.e., set of  $N$  configuration vectors).

As shown in that Figure 3.3, the Euclidean distance matrix  $\mathbf{D}$  of matrix  $\mathbf{P}$  is calculated [87–90]. Matrix  $\mathbf{D}$  is a hollow (diagonal elements are all equal to zero) and symmetric matrix [87–90] and is of size  $N \times N$  for a given set of  $N$  calibration data points. Let  $\mathbf{l}$  denote a vector that contains the  $(N \times (N - 1))/2$  unique (distinct) non-zero values of matrix  $\mathbf{D}$ . Furthermore, let matrix  $\tilde{\mathbf{P}}$  and vector  $\tilde{\mathbf{l}}$  represent arrays that contain the corresponding measurements of those calibration points acquired using the 3D CompuGauge measurement equipment described in Chapter 2.2 (i.e., a set of measured Cartesian coordinate vectors and the distinct non-zero values of their Euclidean distance matrix, respectively). Then, as depicted in Figure 3.3, the multi-variable objective function of our 3DCAL system that is to be minimized (off-line) using our optimization scheme can be expressed as

$$y_{3DCAL}(\mathbf{e}) = \frac{1}{(N \times (N - 1))/2} \sum_{i=1}^{(N \times (N-1))/2} \Delta l_i^2, \quad (3.9)$$

where  $\Delta l_i = l_i - \tilde{l}_i$  (the residual/difference between the calculated and measured distances of the  $i^{th}$  Euclidean distance for a total of  $(N \times (N - 1))/2$  unique Euclidean distances).

The optimal error parameter vector  $\mathbf{e}^*$  can then be calculated as

$$\mathbf{e}^* = \underset{\mathbf{e} \in \Omega}{arg \min} y_{3DCAL}(\mathbf{e}). \quad (3.10)$$

Note that the error function  $y_{3DCAL}(\mathbf{e})$  that is to be minimized is based on the discrepancy between the calculated and measured Euclidian distances among the tool end-points for

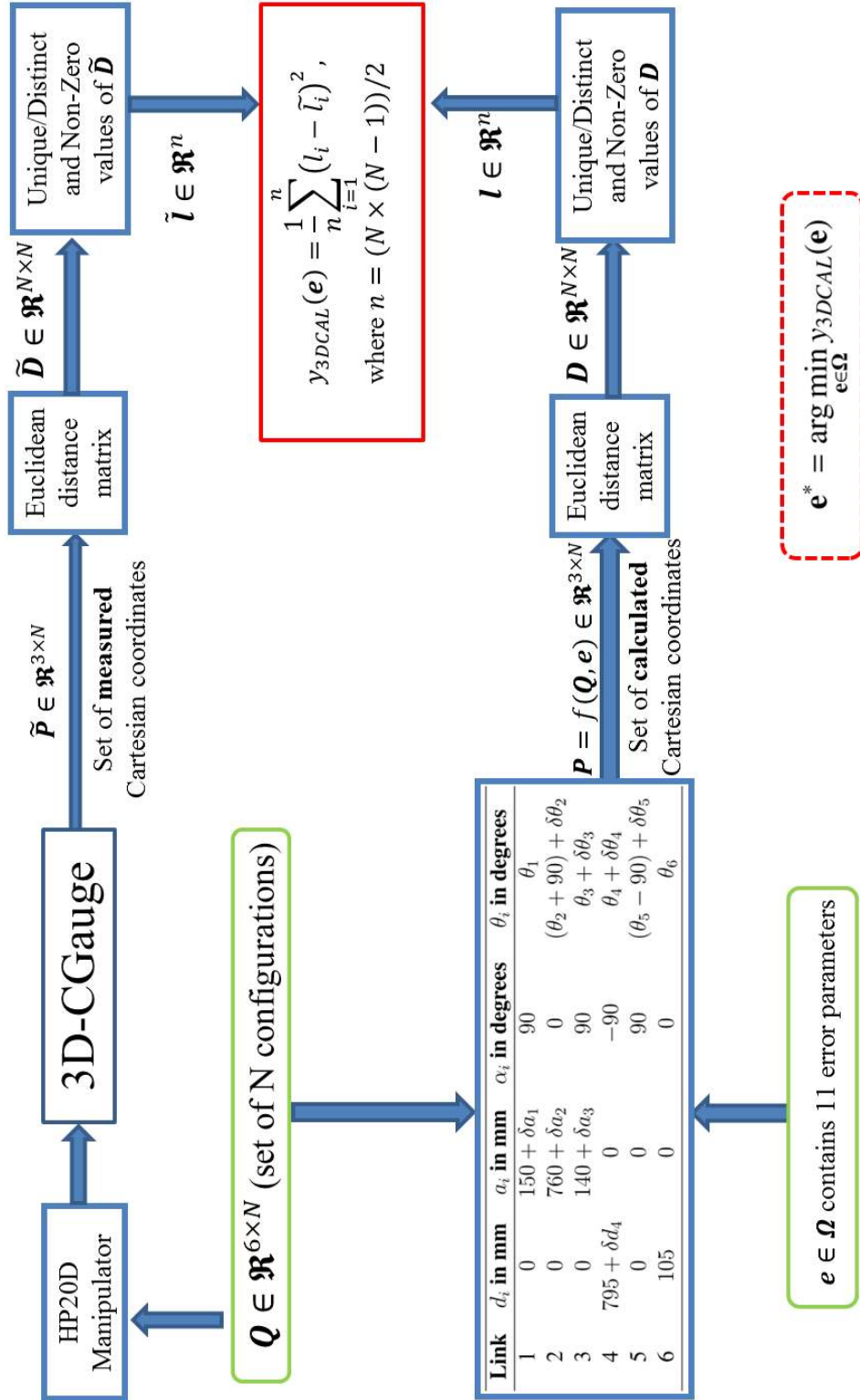


Figure 3.3: 3DCAL block diagram. The DH(-) model of the HP20D is used as an example.

some designated configurations. In other words, we do not explicitly make use of the Cartesian coordinates of the calibration points with respect to a fixed frame (neither the robot's nor 3DCompuGauge's base frames). We believe our approach is therefore more practical and robust compared to the ones described in [12, 66]. Those two methods call for an extra procedure to determine a common reference coordinate frame/system beforehand and that extra step alone could introduce errors that could potentially propagate down to the end-results.  $\Omega$  is robot and manufacturer dependent solution space. More will be said on how to define the dimensions of  $\Omega$  in Chapter 4.3.

Of course, at this point, the challenge is to optimize over the cost function of Eq. (3.9) (i.e., calculating the optimal error parameter vector  $\mathbf{e}^*$  as described in Eq. (3.10)). It is basically a nonlinear least-squares optimization problem that can be solved using either the Trust Region (TR) algorithm [48–52], or Levenberg-Marquardt (LM) algorithm [83–85]. The two are powerful optimization algorithms for large-scale nonlinear problems. In Chapter 3.4, we will compare the two and make a recommendation with regards to which one out of the two is effective and efficient in identifying the error parameters of the DH(-) and DH(-)(-) error models of the two manipulators (HP20D and MH5-Hi) using the discussed 3DCAL method.

### **3.3 1DCAL**

The system presented in this Chapter requires the supply of radial measurements of the calibration point with respect to some fixed reference. Figure 3.4 presents a block diagram for the 1DCAL approach where the MH6 manipulator and its DH(-) error model are used as an example for illustration purposes. Our 1DCAL system is a simplified version of 3DCAL and builds on the work of Goswami *et. al.* [79]. As briefly explained in Chapter 1,

unlike [79] the location of the fixed reference is estimated and then iteratively refined along with the selected DH error parameters via our algorithm during the calibration process.

As it is illustrated in Figure 3.4, we propose to define the main objective function that is to be minimized to estimate the coordinates/location of the fixed reference (encoder)  $\mathbf{p}_e = [x_e, y_e, z_e]^T \in \mathbf{V}$ , and the vector  $\mathbf{e}$  that contains the selected DH error-parameters/deviations as

$$y_{1DCAL}(\mathbf{p}_e, \mathbf{e}) = \frac{1}{N} \sum_{i=1}^N (r_i - \tilde{r}_i)^2, \quad (3.11)$$

where  $r_i$  and  $\tilde{r}_i$  in that equation are the computed and measured radial distances at the  $i^{th}$  calibration point, respectively, out of a total of  $N$  calibration points. More specifically,  $r_i$  can be calculated as

$$r_i = \|\mathbf{p}_e - \mathbf{p}_i\|, \quad (3.12)$$

where the  $i^{th}$  vector  $\mathbf{p}_i$  that describes the position of the end-effector can in turn be calculated by invoking Eq. (3.5) (i.e.,  $\mathbf{p}_i = f(\mathbf{q}_i, \mathbf{e})$ ). This is another nonlinear least-squares optimization problem where either of the two powerful optimization algorithms mentioned in Chapter 3.2 can be used to to optimize over Eq. (3.11) such that the solution/minimizer

$$\mathbf{e}^* = \arg \min_{\mathbf{e} \in \Omega, \mathbf{p}_e \in \mathbf{V}} y_{1DCAL}(\mathbf{p}_e, \mathbf{e}), \quad (3.13)$$

is found.

$\mathbf{V}$  is a spherical volume. More will be said about the dimensionality of  $\mathbf{V}$  in Chapter 4.3. Note that our approach consists of solving for both minimizers  $\mathbf{e}^*$  and  $\mathbf{p}_e^*$  simultaneously (although the former vector contains the relevant information to calibrate the manipulator). Also to note is that those two vectors belong to two drastically different sub-spaces (i.e.,  $\Omega$  and  $\mathbf{V}$ , respectively). We believe a zero valued vector  $\in \Omega$  is an appropriate initial-condition/starting-point for the purpose of solving for  $\mathbf{e}^*$  since it contains deviations of



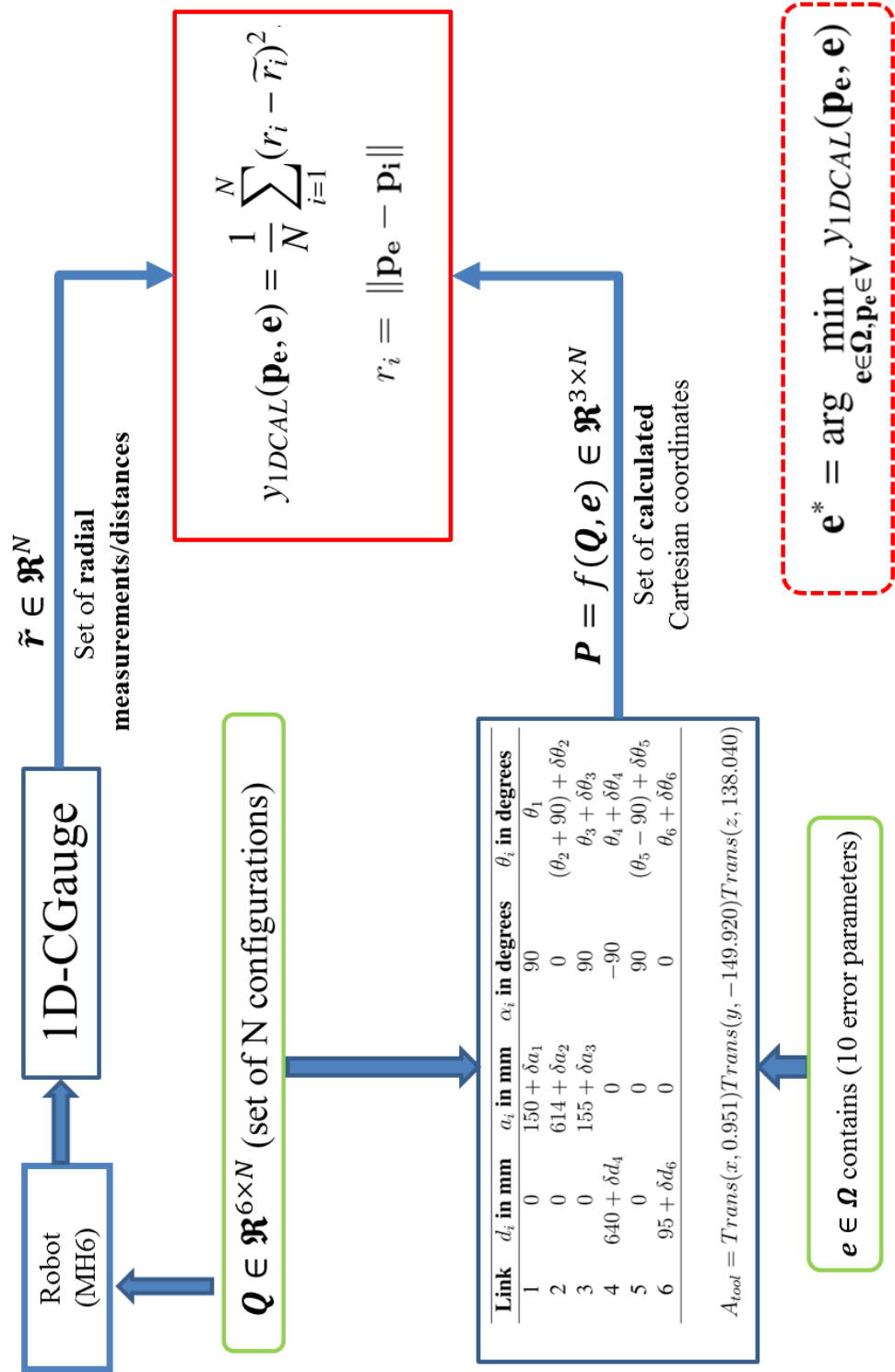


Figure 3.4: 1DCAL block diagram. In this Figure, the DH(-) model of the MH6 is used as an example

small magnitude. However, the vector  $\mathbf{p}_e$  contains the location of the encoder and hence a zero valued vector  $\in \mathbf{V}$  is far from the minimizer/solution  $\mathbf{p}_e^*$ . Thus, to facilitate computation, we take on the task of supplying a different and adequate initial-condition/starting-point in order to solve for  $\mathbf{p}_e^*$  more efficiently. We initially optimize over

$$y_{1DCAL}(\mathbf{p}_e, \mathbf{0}) = \frac{1}{N} \sum_{i=1}^N (r_i - \tilde{r}_i)^2, \quad (3.14)$$

so as to find the minimizer

$$\mathbf{p}_{e0}^* = \mathit{arg} \min_{\mathbf{0} \in \Omega, \mathbf{p}_e \in \mathbf{V}} y_{1DCAL}(\mathbf{p}_e, \mathbf{0}), \quad (3.15)$$

that is relatively closer to the solution  $\mathbf{p}_e^*$ . Hence, as an initial procedure, we execute Eq. (3.15) and make use of  $\mathbf{0}$  and  $\mathbf{p}_{e0}^*$  as an initial starting points to solve for  $\mathbf{e}^*$  and  $\mathbf{p}_e^*$ , respectively, while optimizing over Eq. (3.11). Once more we are in need of an effective optimization scheme to estimate the error parameters of the MH6 using 1DCAL. In the next Chapter, we compare/discuss the TR and LM optimization schemes in details and provide our suggestion concerning which of the two is preferable in terms of effectiveness and efficiency to solve the non-linear least squares optimization problems that we have formulated thus far.

### 3.4 Optimization Scheme: Identifying Error Parameters

In this Chapter, we shall thoroughly describe the optimization scheme employed to solve-for/identify the error parameters of the simplified models (DH(-) and DH(-)(-)) using both frameworks (3DCAL and 1DCAL). Aforementioned, both Eqs (3.9) and (3.11) are nonlinear least-squares optimization problems that can be solved using either Trust Region approach (TR) [48–52] or Levenberg-Marquardt method (LM) [83–85]. TR and LM are two efficient optimization algorithms that can handle large-scale nonlinear optimization

problems. The details of the two optimization algorithms can be found in [48–52, 83–85, 91]. Here, a brief overview of the implemented methods is presented. Given an objective/cost function  $y(\mathbf{e})$  (like Eqs (3.9) and (3.11)), the minimization problem can be mathematically approximated using the Taylor development limited to the second order of the Lagrangian function of  $y(\mathbf{e})$  as

$$\min y(\mathbf{e}_k + \Delta\mathbf{e}) \approx \min m(\mathbf{e}_k + \Delta\mathbf{e}) = y(\mathbf{e}_k) + \mathbf{g}_k^T \Delta\mathbf{e} + \frac{1}{2} \Delta\mathbf{e}^T \mathbf{H}_k \Delta\mathbf{e}, \quad (3.16)$$

where  $m(\mathbf{e}_k + \Delta\mathbf{e})$  is the local quadratical model/approximation of  $y(\mathbf{e})$  around  $\mathbf{e}_k$ .  $\mathbf{e}_k$  is a vector that contains the guesses of the error parameters and  $\Delta\mathbf{e}$  is small perturbation of  $\mathbf{e}_k$ . Vector  $\mathbf{g}_k$  is the gradient of the objective function with respect to  $\mathbf{e}_k$ . It represents the direction of the maximum rate of change of the objective function surface and is ideally zero in magnitude at the minimum. Matrix  $\mathbf{H}_k$  is the Hessian matrix of the objective function and it contains convexity related information of the objective function.

In LM, an iteration/Newtonian step is taken according to

$$\Delta\mathbf{e} = -(\mathbf{H}_k + \lambda_k \mathbf{I})^{-1} \mathbf{g}_k, \quad (3.17)$$

where  $\lambda_k$  is an adjustable scalar and  $\mathbf{I}$  is an identify matrix of size that is equal to  $\mathbf{H}$ . The elements of the update vector  $\Delta\mathbf{e}$  are added to  $\mathbf{e}_k$  to give an improved set of parameters. If at the  $k^{th}$  iteration the condition  $y(\mathbf{e}_k + \Delta\mathbf{e}) > y(\mathbf{e}_k)$  is satisfied (meaning the iteration was unsuccessful), the step is rejected and  $\lambda_k$  is increased. Choosing a relatively large value for  $\lambda_k$  makes  $\mathbf{H}_k$  negligible and the search procedure resembles the well known “the Steepest–Decent” (also known as “Gradient–Search method”) where the Newton step can be approximated as  $\Delta\mathbf{e} = -\mathbf{g}_k$ . In essence, the algorithm relies on gradient information since the quadratic model does not fit properly. Note that large values of  $\lambda_k$  makes the inversion of  $\mathbf{H}_k + \lambda_k \mathbf{I}$  possible even if  $\mathbf{H}_k$  is not positive definite. In fact, the sole purpose

of introducing that parameter is to overcome the difficulties caused by singularities and near singularities of the Hessian matrix. For every  $k^{th}$  successful iteration (i.e.,  $y(\mathbf{e}_k + \Delta\mathbf{e}) < y(\mathbf{e}_k)$ ), the step is accepted and  $\lambda_k$  is decreased in order to exploit the curvature information contained inside the Hessian matrix. In effect, as the search continues while decreasing  $\lambda_k$ , the successive Newtonian steps resembles the ‘‘Gauss–Newton’’ steps (i.e.,  $\Delta\mathbf{e} = -\mathbf{H}_k^{-1}\mathbf{g}_k$ ). In practice,  $\lambda_k$  is chosen in a way which is intended to use Steepest–Descent when far from the solution and Gauss–Newton near the solution. Hence, LM presents a useful combination/mixture of the steepest descent and the Gauss–Newton techniques (i.e., it is a robust algorithm that can solve singular systems).

In actuality, choosing a large  $\lambda$  value in LM decreases the Newton step size (i.e.,  $\|\Delta\mathbf{e}\|$  is reduced). It can be proven that, if we impose a proper limitation on the step size, we can achieve convergence even if the Hessian matrix is indefinite [49, 91]. That is the principle of the TR based approach. Simply put, in TR, the steps are subject to a *Trust Region* that is iteratively modified (i.e., the size of the region is reduced or increased based on a similar validation check like that of LM). Mathematically speaking, according to the TR algorithm, Eq. 3.16 is modified as

$$\min_{\|\Delta\mathbf{e}\| \leq \delta_k} q(\mathbf{e}_k + \Delta\mathbf{e}) = y(\mathbf{e}_k) + \mathbf{g}_k^T \Delta\mathbf{e} + \frac{1}{2} \Delta\mathbf{e}^T \mathbf{H}_k \Delta\mathbf{e}, \quad (3.18)$$

where  $\delta_k$  is the trust region of the model  $q(e_k)$  (i.e., the region where we trust the model is valid). The basic algorithmic steps of TR are as follow:

1. While the stopping criterion is not met; compute the provisional update step  $\Delta\mathbf{e}$  as

$$\Delta\mathbf{e} = -\mathbf{H}_k^{-1}\mathbf{g}_k.$$

2. Compute the ratio

$$\frac{ar_k}{pr_k} = \frac{y(\mathbf{e}_k + \Delta\mathbf{e}) - y(\mathbf{e}_k)}{m(\mathbf{e}_k + \Delta\mathbf{e}) - m(\mathbf{e}_k)},$$

where  $ar_k$  and  $pr_k$  are the *actual* and *predicted* reductions, respectively.

3. Compute a new set of parameters as

$$\mathbf{e}_{k+1} = \begin{cases} \mathbf{e}_k, & \text{if } ar_k/pr_k < \eta_1 \\ \mathbf{e}_k + \Delta\mathbf{e}, & \text{otherwise.} \end{cases}$$

4. Modify the trust region as

$$\delta_{k+1} = \begin{cases} \gamma_1 \delta_k, & \text{if } ar_k/pr_k < \eta_1 \\ \delta_k, & \text{if } \eta_1 \leq ar_k/pr_k \leq \eta_2 \\ \max\{\delta_k, \gamma_2 \delta_k\}, & \text{if } ar_k/pr_k > \eta_2. \end{cases}$$

5. Increment index  $k$  and go to step 1.

In the steps of the algorithm presented in the above, the condition parameters are limited to  $0 < \eta_1 < 0.5 < \eta_2 \leq 1$  and the scaling (“shrinking” or “growing”) factors are subject to  $0 < \gamma_1 < 1 < \gamma_2$ . The algorithm is quite insensitive to the constraints. We made use of typical values. The typical values used in this work are  $\eta_1 = 0.25$ ,  $\eta_2 = 0.75$ ,  $\gamma_1 = 0.5$  and  $\gamma_2 = 2$ . These conditions and scaling factors ensure that TR makes use of a trust region within which the quadratic model of the Lagrangian function is believed to be accurate (i.e., the model/approximation is valid in a neighborhood of  $\mathbf{e}_k$  that is restricted by the trust region radius  $\delta_k$ ).

While we found the described/generic LM and TR approaches to be effective in solving all optimization problems presented in this dissertation, for sake of completeness, we made note of some interesting variations of those techniques. Recently, [92] suggested to replace the equation of the provisional update/newtonian step  $\Delta\mathbf{e}$  described in step 1 of TR with Eq. (3.17). Moreover, [92] goes on in making the trust region modification described in step 4 of TR as a function of  $\lambda_k$  [92]. In other words, this *modified LM/TR algorithm* modifies the parameter  $\lambda_k$  in every iteration, which in turn modifies  $\delta_k$  implicitly [92]. We

also have [93] who propose *a new LM method* where for the  $k^{th}$  unsuccessful iteration, rather than completely rejecting the step, the step is accepted but the provisional update/newtonian step  $\Delta e$  is subject to a damping factor that is obtained by *Wolfe* or *Armijo* line search [93].

To summarize it all, LM and TR are both restricted Newton Step based methods and hence exhibit quadratical speed of convergence for iterations near the solution  $e^*$ . Moreover, they both can handle singular system of non-linear equations. However when we are far from the solution, LM will slow down drastically since it relies mostly on the Gradient Search method. In contrast, the solution trajectory of TR can potentially perform a very long leap and expedite the search process. This efficient facet of TR has been proven by numerous literatures including [91] and [94]. In particular, as sated by [91], “[TR] methods are an evolution of [LM] algorithms. [TR] methods are able to follow the negative curvature of the objective function. [LM] algorithms are NOT able to do so and are thus slower.” We strongly concur with those literatures as TR demonstrated a slightly faster convergence rate for *all cases* of nonlinear least square minimization problems presented/encountered in this dissertation. These include Eqs 3.9, 3.11 and 3.14. For example, in the case of minimizing Eq. 3.14, given the same initial condition (a zero vector) and identical tolerance based stopping-criterion/termination-condition, LM carried out 13 iterations while TR only performed 7 iterations to converge to the same solution. Hence, even though they both converged to the same numerical solutions for all calibration cases, we favor TR as it is computationally efficient for this application. In this work, the termination condition for TR that we used is until the deviation error is small enough (in the order of  $10^{-12}$ ). We also made use of TR (as a local solver) for the purpose of carrying out a parametric redundancy auto-assessment using a Simulated Annealing based Global Search that we are about to

introduce in the next Chapter (Chapter 3.5). We found the assessment to be very handy in facilitating the derivation of the simplified kinematic error model DH(-).

### **3.5 The DH(-) Kinematic Calibration Model: Automated/Numerical Method**

In this Chapter, we propose an automated method that calls for numerical analysis to provide a redundancy assessment of the error parameters. Note that the sole purpose of this evaluation is to provide insight in identifying redundant parameters so as to facilitate the derivation/construction of the DH(-) model of a given manipulator. That is to say, the user may opt to use the “conceptual/analytical approach” described in Chapter 3.1 or the automated method that we are about to explain in order to deduce the DH(-) model. Once DH(-) model is found the error parameters can be solved using the efficient solver discussed in Chapter 3.4. Figure 3.5 depicts a block diagram of the discussed product usage protocol. The first step/“stage” is to find the DH(-) model of the manipulator in question. As shown in that figure, “stage 1” can be carried out either using the “concept approach” or “the automated approach”. Note that “the automated approach” does increase the burden on the user (running an optimization-scheme/Global-Search using available data) but is undoubtedly helpful in deriving the DH(-) model of the robot in question (in particular, it can greatly help if the geometry of the structure is complex and the guidelines of the “analytical approach” can no longer be extended in an obvious fashion). Provided that the DH(-) model is correctly identified, the user can complete the second stage of the calibration process; solving for the non-redundant error parameters using TR. The friendly Graphical User Interface (GUI) that we have devised to allow the user to carry out the calibration procedure per the discussed protocol is presented in Appendix E.

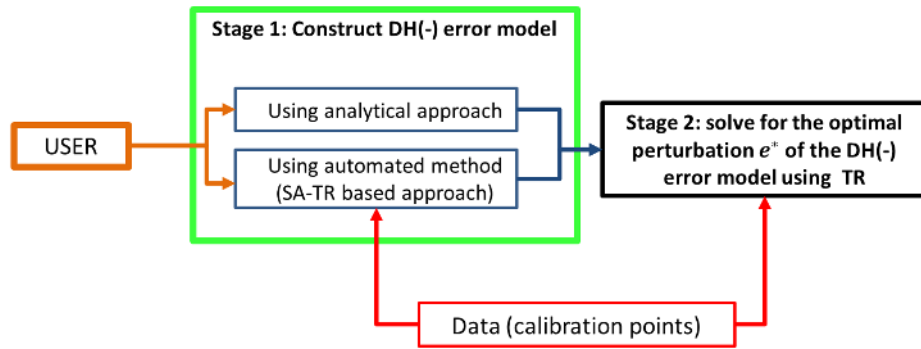


Figure 3.5: Top level block diagram illustrating the product usage protocol.

As mentioned in the previous Chapter and as it will be demonstrated in Chapter 4, TR is found to be very adequate to identify the error parameters of the DH(-) and DH(-)(-) kinematic calibration models (i.e., solving non-redundant systems). However, when it comes to redundant error models (e.g., original and continuous DH based error model of the HP20D manipulator shown in Table 3.1), TR and similar solvers are sensitive to the supplied initial condition/guess (starting point) when searching for global solution(s) as error models with redundancy are problems that contain multiple global maxima or minima. We propose to take advantage of such sensitivity in order to identify redundant parameters by performing a Global Search (GS) using multiple starting points. We seek to examine each starting point using TR and inspect if the selected error parameters converge to the same numerical values in which case those will be deemed non-redundant. Conversely, error parameters that converge to different numerical values but without any significant impact in the output of the objective function are adjudged ineffectual. We chose TR as a local solver (to examine each starting point) since not only it exhibits excellent local convergence but it is also efficient.



We propose to make use of another type of solver to gather a rich set of starting points. We then run TR using those starting points in order to perform a GS for the solution. In essence, we have a GS mechanism similar to the ones described in [53–55]. Unlike [53–55], we make use of Simulated Annealing (SA) [43–47] to generate starting point candidates. Let us briefly describe SA’s algorithm operation before embarking on to the use of it to produce a rich set of initial conditions.

SA is a stochastic optimization algorithm on its own right [43–47]. The method models the physical process of “heating” the material and then “cooling” the temperature to decrease defects, thus minimizing the system energy. The basic operation of SA begins by allowing the solution trajectory to take large random steps in the solution space (i.e., heating of the solution space). If the output of the objective function is smaller than the previous (i.e., if a solution that further minimizes/reduces output of cost function compared to the precedent is found), the current/new solution is accepted. Else, the new solution is only retained with some probability. The greater the increase in the output of the objective function, the lower the probability of acceptance. By registering points that raise the objective, the algorithm avoids being trapped in local minima, and is able to explore globally for more possible solutions. As the algorithm progresses, the temperature decreases (i.e., “cooling regime”) in order for algorithms’ search to converge to a minimum [43–47]. After a certain number of new points are accepted, the temperature is raised to a higher value in hope to restart the search and move out of a local minimum (re-annealing process). Since the position of the final point in the solution/state space is subject to stopping criteria (which could have terminated the algorithm while the solution trajectory is taking “uphill” path), it is not always the optimal answer. Therefore, the best solution visited over the entire SA session is outputted.

Given an objective function, like the one described in Eq. (3.9) for example, which maps the multi-dimensional solution sub-space  $\Omega \in \mathbb{R}^m \rightarrow \mathbb{R}$  where  $m$  is the number of selected error parameters (i.e., dimensionality of the solution sub-space) and  $\mathbf{e} \in \Omega$  is a vector in that sub-space, the flow of simulated annealing for a given temperature schedule can be enumerated as follows:

1. Choose a random vector  $\mathbf{e}_0$ ; initial guess at the solution.
2. Evaluate  $y_0 = y_{3DCAL}(\mathbf{e}_0)$ .
3. Take a random step in the solution space to a neighboring vector  $\mathbf{e}_1$
4. Evaluate  $y_1 = y_{3DCAL}(\mathbf{e}_1)$ .
5. Continue iterating in this fashion until stopping criteria is met. During the iteration,
6. if  $y_{k+1} < y_k$  at the  $k^{th}$  iteration, then update the solution to  $\mathbf{e}_{k+1}$  (i.e.,  $\mathbf{e}_{k+1}$  is the new/current solution).
7. If on the other hand at any iteration  $y_{k+1} > y_k$  then retain  $\mathbf{e}_{k+1}$  as the solution with probability  $p_k = e^{-\Delta/T}$  where  $\Delta = |y_{k+1} - y_k|$  (i.e., difference of objective function values between two consecutive steps) and  $T$  is the system's temperature.
8. Reduce the system temperature according to schedule and the final solution is  $\mathbf{e}_k$  such that  $y_k$  is the smallest value that has been achieved at any point in the algorithm after terminating the algorithm.

Note that in this work it was considered to use SA as a stand alone optimizer for this application but we found SA to be inadequate/unsuccessful in converging to an acceptable result (the optimal solution). Difficult and arduous tasks that need to be mastered for SA

to be effective include a proper choice of initial temperature and temperature model for each dimension of the solution space, setting tuning parameters (e.g., annealing constant, re-annealing interval, etc.), determining search stopping criteria (e.g., maximum number of iterations, change in function or parameters value [tolerance], time allowed, etc.) [43–47]. However, we made note of SA’s potent capability of intelligently exploring the solution sub-space  $\Omega$ .

It is for that reason that in our study we decided to combine the best of two worlds. We propose a hybrid optimization scheme that consists of the outstanding probing capability of SA (i.e., intelligent search with a fast temperature schedule) with the first-rate convergence properties of TR. In our approach we first retain the results obtained over the course of multiple SA runs using the maximum number of iterations as a stopping criterion for each run (50 distinct outcomes to be precise). We then employ the “local optimizer” TR using SA’s results as potential starting points to carry out a GS. In other words, each (for a total of 50) output produced by SA is further examined with the TR agent (instead of solely relying on the convergence properties [cooling regime] of SA). We believe our hybrid optimization scheme is computationally efficient compared to the ones described in [53–55] because those methods call for a higher number of starting points (obtained via a scatter-search mechanism or uniformly distributed starting points within bounds) and an exhaustive inspection using similar non-linear programming.

As mentioned in the latter, the purpose of this hybrid optimization scheme is to provide insight about the relevance of the selected parameters. That is, insight regarding unwanted redundance within the model due to some ineffectual/irrelevant parameters (i.e., linearly dependent parameters and such). The parameters that present noticeable jitter but without significant change in the output of the objective function are highly likely to be redundant

and hence some of those parameters need to be discarded/removed in order to render the model non-redundant. In other words, a modest investigation into the flagged parameters can effectively guide the user in simplifying the model (i.e., correctly finding the DH(-) model). Those “suspicious” parameters can be identified by simply examining the variance across the multiple solutions of a given error parameter (multiple solutions found using our SA-TR based GS).

Let us look at an example to verify/validate the effectiveness of the automated redundancy detection algorithm that we just discussed (i.e., variance analysis). Let us also try to possibly correlate the results to the ones found using the “conceptual/analytical approach” as the guidelines established conceptually are meant to avoid redundancy. In other words, the goal of the two approaches (conceptual and automated) is to construct/find the DH(-) model of a robot and hence one would expect a strong agreement among the two analyses. Table 3.6 presents results if one were to elect to estimate all non-zero DH parameters for the HP20D manipulator using our 3DCAL method (a total of 14 error parameters). The results were generated by employing TR as a local optimizer using each of the 50 starting points obtained through SA. The details of the optimization session for one the 50 starting point is presented in Figure 3.6 (i.e., optimal solution [“current point”], iteration versus Norm of residuals, step size, number of objective function evaluations, etc.). Other details of the experiment shall be discussed in Chapter 4. The goal here is to demonstrate that the established guidelines to avoid redundancy and to make the model less complex (i.e., to construct the DH(-) model) are in accordance with the automated assessment and hence are justifiable and well-founded.

The automated evaluation (variance analysis) presented in Table 3.6 is consistent with the guidelines that we analytically established in Chapter 3.1 in order to identify irrelevant

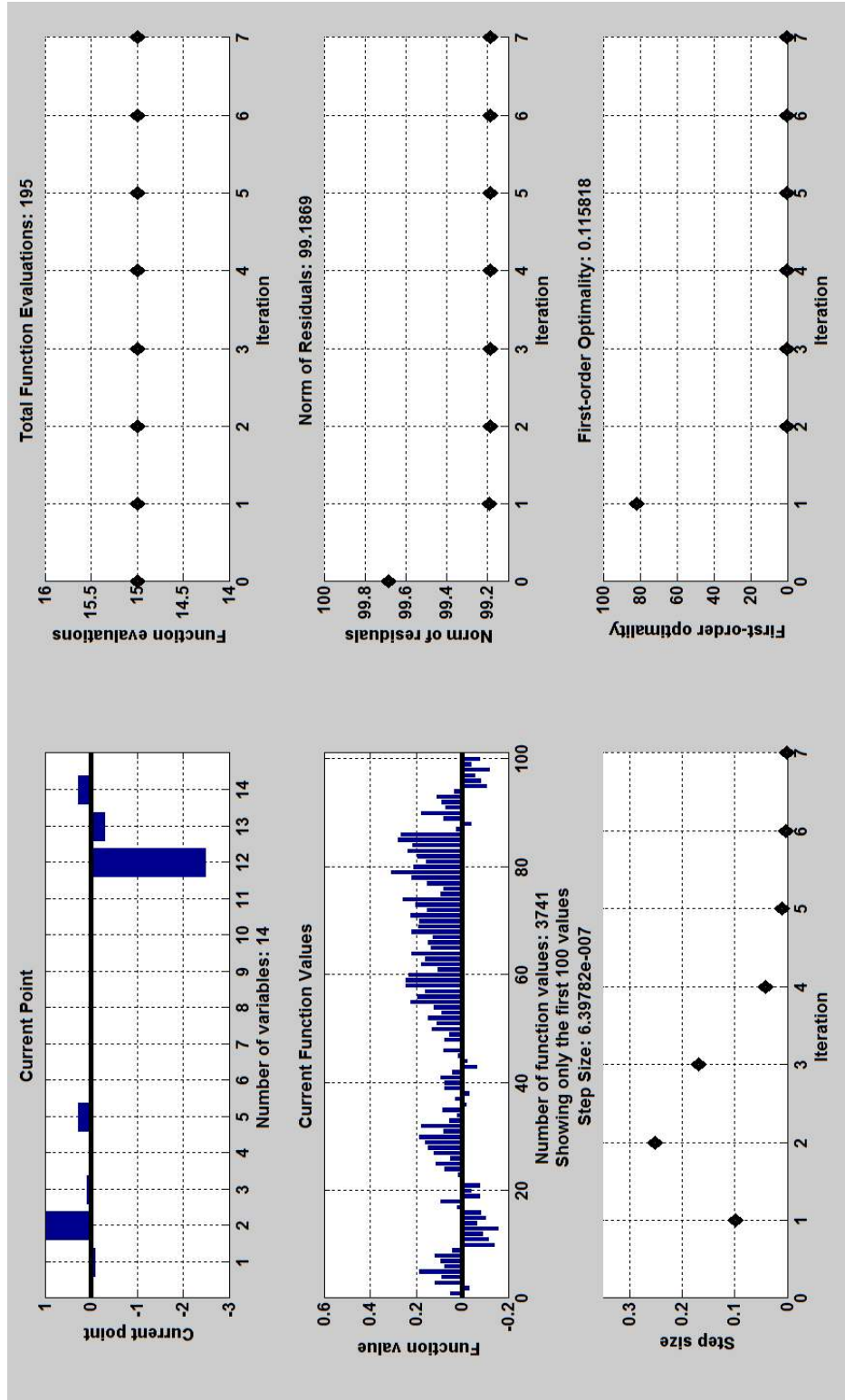


Figure 3.6: 3DCAL: TR optimization session details for a given starting point (out of the potential 50).

Table 3.6: “Stage 1” using the “automated approach”: auto-relevance assessment of the various error parameters using 3DCAL and the HP20D manipulator.

<b>Selected Parameters</b>	<b>Nominal Values</b>	<b>Variance</b>
$a_1$ in mm	150.000	0.001
$a_2$ in mm	760.000	0.000
$a_3$ in mm	140.000	0.006
$d_4$ in mm	795.000	0.001
$d_6$ in mm	105.000	<b>1.184</b>
$\delta\theta_1$ in degrees	0.000	<b>2.263</b>
$\delta\theta_2$ in degrees	0.000	0.000
$\delta\theta_3$ in degrees	0.000	0.000
$\delta\theta_4$ in degrees	0.000	0.000
$\delta\theta_5$ in degrees	0.000	0.000
$\delta\theta_6$ in degrees	0.000	<b>2.117</b>
$X_{tool}$ in mm	2.549	0.001
$Y_{tool}$ in mm	0.342	0.002
$Z_{tool}$ in mm	70.079	<b>1.184</b>

parameters and to render the model non-redundant (i.e., to correctly construct the DH(-) model of the manipulator). It follows as expected, the error corresponding to the first joint angle ( $\delta\theta_1$ ) can not be estimated using the proposed method since all calibration points are shifted by the same amount while the Euclidean distances among the points remain unaltered. This is the reason why a great level of parametric variance is observed without a change in the output of the objective function. Mathematically speaking

$$\frac{\partial}{\partial\delta\theta_1}y_{3DCAL}(\mathbf{e}) = 0.$$

Also to note is the great variances of the error associated with DH parameter  $d_6$ , the error corresponding to the last joint angle ( $\delta\theta_6$ ) and the  $Z$  dimension of the tool frame. As told in Chapter 3.1, the user needs to choose whether to estimate the error parameters of the last row of the DH table (i.e.,  $A_6(q_6)$ ) or tool definition  $A_{tool}$  but not both. Note that parameters  $d_6$  and the  $Z$  dimension of the tool are actually linearly dependent as they describe the

same dimension of the robot and tool for any configuration (just like the error parameters  $\delta d_2$  and  $\delta d_3$  that are dependent because of the parallel axes of joints 2 and 3). That is to mathematically say

$$\frac{\partial^3}{\partial \delta \theta_6 \partial \delta d_6 \partial \delta Z_{tool}} y_{3DCAL}(\mathbf{e}) = 0.$$

holds true for numerous combinations/solutions of those error parameters.

Restating in part what was mentioned previously, we denote Table 3.6 as “Stage 1” because it can be viewed as an intermediate procedural calibration step that the user may choose to perform at the expense of an optimization session using available data in order to deduce which of the error parameters associated to non-zero valued DH parameters are redundant. Based on this insight, the user can elect which of those need to be kept frozen so as to correctly construct the DH(-) error model of the manipulator and perform “Stage 2”; identifying the error parameters of the simplistic and non-redundant kinematic calibration model (the DH(-) model). Note that the results and comparative analyses presented in this Chapter demonstrate that the guidelines that we conceptually/analytically established for the purpose of deduce the DH(-) model are righteous/well-founded and as reliable as the “automated”/numerical approach. Considering the burden of “Stage 1” on the user, we herby recommend to make use of the justified “conceptual”/analytical approach to construct DH(-) model unless the manipulator that needs to be calibrated presents great geometry/kinematic related challenges.

After successfully completing the course of DH(-) model search (“Stage 1”), the error parameters can be identified/solved-for using the TR optimization scheme described in Chapter 3.4 (“Stage 2”). Note that the DH(-) and DH(-)(-) models are non-redundant and hence the objective/cost functions corresponding to those error models may possibly be convex-like in nature [95]. It follows that it is sufficient to run TR with an arbitrary starting

point to attain a reliable and robust solution as convergence is inevitable (i.e., a SA-TR based GS is simply unnecessary.) More will be said about this in Chapter 4.3. The reader should be well informed that we have compared both optimization techniques (TR and LM) over the course of completing “Stage 2” for all calibration cases. Although both optimizers converge to the same numerical solutions for all “Stage 2” calibration cases, TR indicated a quicker convergence rate (on average 10 iterations for any given arbitrary starting point compared to  $> 17$  when employing LM.) For that reason, we declare TR as the best candidate to execute the second stage of the calibration processes for all situations.



## CHAPTER 4

### EXPERIMENTAL RESULTS AND DISCUSSION

In this Chapter, we present numerous results to demonstrate the effectiveness of the proposed calibration systems. We present performance analyses of 3DCAL and 1DCAL using Compugauge's performance standard/metric described in Chapter 2.2. We also present results of 3DCAL using the MH5-hi and CMM using a detailed statistical performance analyses. The performance of our approaches are compared to the performance of a state-of-the-art commercial system (MotoCal) using same materials and performance metric. This Chapter concludes with a discussion (of the systems and results) and thoughts for future improvements and work.

#### 4.1 3DCAL Performance Analysis

In order to test our 3DCAL system, we first made use of the HP20D manipulator described in Chapter 2.1. Let us begin by briefly describing the method we used to gather the calibration points. We used a uniformity sampled grid that is comprised of 27 calibration points and that covers most of the working envelope in front of the robot. Then we incorporated orientations to those 27 calibration points. Namely, orientation about the  $x$  - axis of the base frame in the order of -25 and 25 degrees. And, orientation about the  $y$  - axis of the base frame in the order of -25 and 25 degrees. This gave rise a total of 135

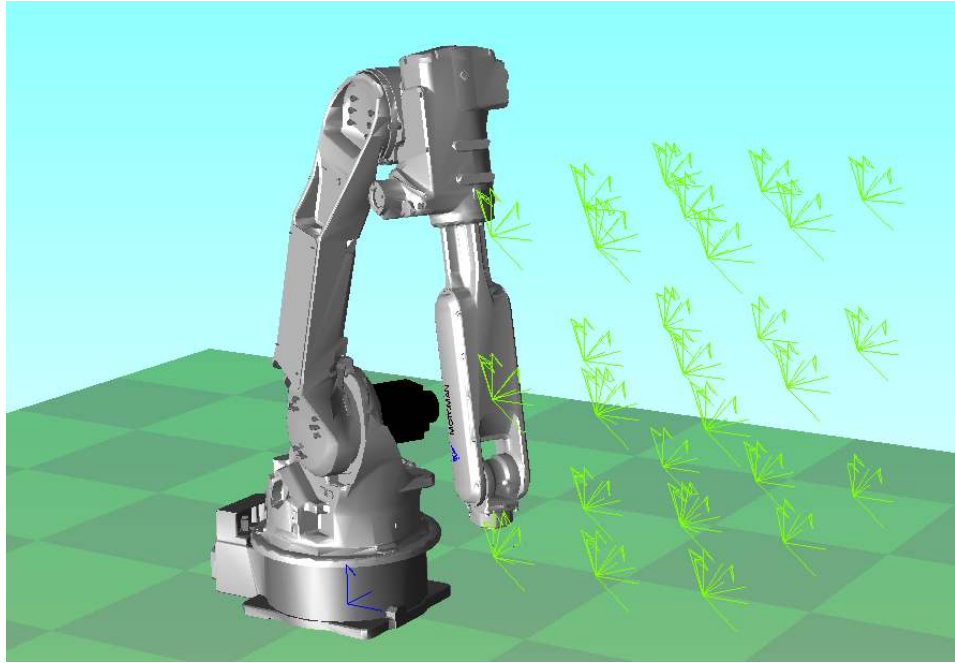


Figure 4.1: Calibration points used to calibrate the HP20D Manipulator [9].

calibration points. Figure 4.1 depicts the calibrations points. Note that all configurations shown in that figure are non-singular.

Unfortunately, not all calibration points could be measured using CompuGauge due to hardware limitations (occurrence of cable twist for some position measurements). Note that this is the reason why we generated orientation-limited calibration points in the first place. After carefully testing each point, we concluded that only 87 calibration points out of the 135 are valid (could be measured).

Table 4.1 shows the estimated errors using the simplified kinematic model  $DH(-)$  for the HP20D manipulator. We employed TR using an arbitrary starting point to optimize over the objective function associated with the  $DH(-)$  error model. Note that the termination condition that we used is until the deflection error is small enough (in the order of  $10^{-12}$ ).

Table 4.1: 3DCAL and MotoCal “Stage 2” calibration results using 3D CompuGauge and the DH(-) error model of the HP20D manipulator.

Selected parameters	Nominal values	3DCAL	MotoCal
		Estimated values	Estimated values
$a_1$ in mm	150.000	149.901	N/A
$a_2$ in mm	760.000	760.971	761.000
$a_3$ in mm	140.000	140.071	N/A
$d_4$ in mm	795.000	794.966	794.960
$\delta\theta_2$ in degrees	0.000	0.135	0.143
$\delta\theta_3$ in degrees	0.000	1.001	1.002
$\delta\theta_4$ in degrees	0.000	-0.156	-0.156
$\delta\theta_5$ in degrees	0.000	0.614	0.614
$X_{tool}$ in mm	2.549	0.046	0.066
$Y_{tool}$ in mm	0.342	0.015	0.020
$Z_{tool}$ in mm	70.079	70.615	70.625
<b>Performance Analysis: Alignment Test</b>			
$\Delta X$ in mm	7.4553	0.2798	0.2867
$\Delta Y$ in mm	0.8770	0.3039	0.3117
$\Delta Z$ in mm	1.7363	0.2001	0.2167

The table shows result for what we referred to as “Stage 2” of calibration process (i.e., after removing ineffectual and/or redundant error parameters). As mentioned in Chapter 3.2, we opt to estimate the tool definition in this experiment/case due to the fact that we did not have a well defined tool at hand. In order to obtain an initial estimate (the nominal values of the tool shown in Table 4.1), we used the “tool calibration utility” of the DX100 controller [2]. It provides an automatic method to calculate the  $X$ ,  $Y$  and  $Z$  dimensions of the mounted tool. Obviously, it is erroneous since the robot itself was not well calibrated; and hence the need to estimate tool data.

Table 4.2: 3DCAL and MotoCal “Stage 2” calibration results using 3D CompuGauge and the DH(-)(-) error model (home position and tool data) of the HP20D manipulator.

Selected parameters	Nominal values	3DCAL	MotoCal
		Estimated values	Estimated values
$\delta\theta_2$ in degrees	0.000	0.162	0.153
$\delta\theta_3$ in degrees	0.000	1.029	1.001
$\delta\theta_4$ in degrees	0.000	-0.149	-0.153
$\delta\theta_5$ in degrees	0.000	0.612	0.612
$X_{tool}$ in mm	2.549	0.032	0.041
$Y_{tool}$ in mm	0.342	0.061	0.020
$Z_{tool}$ in mm	70.079	70.425	70.588
<b>Performance Analysis: Alignment Test</b>			
$\Delta X$ in mm	7.4553	0.3012	0.2997
$\Delta Y$ in mm	0.8770	0.4624	0.4722
$\Delta Z$ in mm	1.7363	0.3889	0.4039

The performance metric that we used to validate our approach is the so called “alignment test” [5, 6]. In order to perform this alignment test, we used eight points (the vertices of a cube that covers the majority of the working envelope of the robot). Those eight points are used to find the “best” fit between the coordinates of those eight points with respect to the robot frame and measurement system frame. The fit of the alignment procedure reflects the accuracy of the robot itself; a performance standard that is widely used in the robotics community [5, 6]. Before calibration, the misfit was in the order of 7.4553, 0.8770 and 1.7363 mms along  $X$ ,  $Y$  and  $Z$  dimensions respectively. After employing our 3DCAL approach, we were able to reduce the misfit to 0.2789, 0.3039 and 0.2001 mms. According to our experiment results (shown in Table 4.1), our performance is comparable (in fact slightly better) compared to that of MotoCal.

MotoCal has similar features. It allows user interaction (which parameters to estimate). Similar to our automated/numerical parametric relevance assessment that the user may perform to complete “Stage 1”, it also provides feedback to the user regarding which parameters ought to be frozen in order to have reliable results (the so called “Acceptance Angular Stdv”) [6]. If the “Acceptance Angular Stdv” of a given parameter is above a given threshold (default value in the order of 0.04), the algorithm recommends that the parameter be omitted during the second stage of the calibration procedure. Note that this is achieved at the expense of an optimization session using available data. Based on that, in addition to  $\theta_1$ , DH parameters  $a_1$  and  $a_3$  are kept frozen during MotoCal’s second calibration round. Hence, the MotoCal’s results shown in Table 4.1 can also be referred to as “Stage 2” calibration results. Note that the same data (87 calibration points) and manipulator (HP20D) were put to use to generate the results shown in Table 4.1 using MotoCal. It is interesting to compare the estimated values. One could state that the numerical solutions obtained using MotoCal are very similar to the ones found using our 3DCAL approach. We attribute the slight boost in performance of 3DCAL to the two extra error degrees of freedom ( $a_1$  and  $a_3$ ).

We also present results for the other experiment where we elect to estimate the error parameters associated with the joint variables and tool data only (i.e., we make use of yet a further simplified model; the DH(-)(-) model). Table 4.2 shows those results. These results are very informative for two reasons: First, it appears that most of the inaccuracy is due to errors associated with the offset of joint variables (i.e., home position). We say this because only a slight deterioration in performance is observed when comparing Tables 4.1 and 4.2. Second, it allows us to make a definite comparison between the two systems (i.e., same

number of error parameters). Based on that context, we solidly affirm that our performance is indeed comparable to that of MotoCal.

Table 4.3: 3DCAL and MotoCal “Stage 2” calibration results using CMM and the DH(-) error model of the MH5-Hi manipulator.

<b>Selected parameters</b>	<b>Nominal values</b>	<b>3DCAL</b>	<b>MotoCal</b>
		<b>Estimated values</b>	<b>Estimated values</b>
$a_1$ in mm	88.000	87.864	N/A
$a_2$ in mm	310.000	310.391	310.354
$a_3$ in mm	40.000	40.003	40.064
$d_4$ in mm	305.000	305.034	304.925
$d_6$ in mm	80.000	79.824	N/A
$\delta\theta_2$ in degrees	0.000	-0.159	-0.144
$\delta\theta_3$ in degrees	0.000	0.169	0.167
$\delta\theta_4$ in degrees	0.000	0.333	0.325
$\delta\theta_5$ in degrees	0.000	0.049	0.063
$\delta\theta_6$ in degrees	0.000	-0.209	N/A
<b>Performance Analysis:</b>			
<b>Step</b>			
<i>Mean error in mm</i>	0.237	0.043	0.049
<i>Minimum error in mm</i>	0.018	0.007	0.015
<i>Maximum error in mm</i>	0.510	0.080	0.088
<i>Spread in mm</i>	0.492	0.071	0.073
<i>Standard Deviation in mm</i>	0.131	0.018	0.019
<b>Cube</b>			
<i>Mean error in mm</i>	0.819	0.185	0.186
<i>Minimum error in mm</i>	0.292	0.039	0.049
<i>Maximum error in mm</i>	1.650	0.376	0.351
<i>Spread in mm</i>	1.357	0.331	0.302
<i>Standard Deviation in mm</i>	0.419	0.099	0.090
<b>End-Effector Roll</b>			
<i>Mean error in mm</i>	0.763	0.255	0.207
<i>Minimum error in mm</i>	0.354	0.034	0.042
<i>Maximum error in mm</i>	1.196	0.524	0.430
<i>Spread in mm</i>	0.842	0.472	0.388
<i>Standard Deviation in mm</i>	0.266	0.125	0.109

Table 4.4: 3DCAL and MotoCal “Stage 2” calibration results using CMM and the DH(-)(-) error model (home position only) of the MH5-Hi manipulator.

<b>Selected parameters</b>	<b>Nominal values</b>	<b>3DCAL</b>	<b>MotoCal</b>
		<b>Estimated values</b>	<b>Estimated values</b>
$\delta\theta_2$ in degrees	0.000	-0.165	-0.131
$\delta\theta_3$ in degrees	0.000	0.218	0.208
$\delta\theta_4$ in degrees	0.000	0.354	0.342
$\delta\theta_5$ in degrees	0.000	0.081	0.053
$\delta\theta_6$ in degrees	0.000	-0.239	N/A
<b>Performance Analysis:</b>			
<b>Step</b>			
<i>Mean error in mm</i>	0.237	0.043	0.049
<i>Minimum error in mm</i>	0.018	0.010	0.010
<i>Maximum error in mm</i>	0.510	0.073	0.093
<i>Spread in mm</i>	0.492	0.063	0.083
<i>Standard Deviation in mm</i>	0.131	0.019	0.024
<b>Cube</b>			
<i>Mean error in mm</i>	0.819	0.387	0.407
<i>Minimum error in mm</i>	0.292	0.082	0.075
<i>Maximum error in mm</i>	1.650	0.671	0.666
<i>Spread in mm</i>	1.357	0.568	0.592
<i>Standard Deviation in mm</i>	0.419	0.169	0.170
<b>End-Effector/Tool Roll</b>			
<i>Mean error in mm</i>	0.763	0.304	0.249
<i>Minimum error in mm</i>	0.354	0.077	0.068
<i>Maximum error in mm</i>	1.196	0.582	0.491
<i>Spread in mm</i>	0.842	0.505	0.423
<i>Standard Deviation in mm</i>	0.266	0.135	0.123

In a final test we evaluated our TR based 3DCAL approach using a higher axes resolution robot (MH5-Hi described in Chapter 2.1) and a more accurate measuring device (CMM described in Chapter 2.3). The same technique as for the experiment with the HP20D is used to acquire the Cartesian coordinates of the calibration points used for the purpose of estimating the 10 kinematic error parameters. To be precise, a total of 71 calibration points

(non singular configurations) that cover most of the MH5–Hi dexterous workspace are utilized. Table 4.3 shows the results for “Stage 2” of the calibration procedure (estimated errors using the simplified DH(-) kinematic model of the MH5–Hi manipulator and the CMM machine). Note that, although avoidable, we took the task of completing “Stage 1” of the calibration process for sake of illustrating the effectiveness of the auto–relevance/numerical evaluation of all the error parameters associated with non–zero valued DH parameters. And as expected (based on our “conceptual approach”), our auto–relevance evaluation flagged  $\delta\theta_1$  as the only unstable error parameter. Furthermore, as also anticipated, the error parameter  $\delta\theta_6$  and the one associated with the DH parameter  $d_6$  showed negligible variances (i.e., unlike the HP20D’s case, since we made use of a well defined tool those two parameters can be calibrated.) We of course opt not to calibrate the tool data considering an accurate definition of it is provided. A similar parametric assessment is performed using MotoCal as well. In addition to  $\delta\theta_1$ , as shown in that table, MotoCal recommended the user not to perturb 3 additional parameters during “Stage 2” of the calibration procedure ( $a_1, d_6$  and  $\delta\theta_1$ ).

We made use of a different performance metric (compared to the one used to evaluate the two systems using HP20D manipulator and CompuGauge) to evaluate and compare the two calibration methods. The performance analysis labeled as “Step” in Table 4.3 is calculated by examining the difference between the measured and commanded positions (with respect to a common frame of reference) while the robot is programmed to move linearly along a dimension with fixed orientation. The “step” segment is comprised of 40 linearly spaced calibration points ranging from –200 to 200 mms along  $Y$ -axis of the robot’s base frame. The “Cube” performance analysis is computed in the same fashion except the manipulator is programmed to move at the vertices of and center of a cube (i.e.,



total of 9 calibration points). The dimensions of the cube are in the order of 200 mms along the  $X$ -axis of the robot's base frame, 600 mms along the  $Y$ -axis and 350 mms along the  $Z$ -axis. The performance/error statistics for "Roll" are calculated by systematically varying the orientation of the tool for a fixed/commanded position and examining the discrepancies among the measured values. For a well calibrated robot those values should not change as the orientation of the tool is the only variable (and not the position of the tool). The "Roll" segment consists of orientations of the tool about all three principal axes for a total of 13 calibration points (hand picked combination of  $RX_{tool} = [0^\circ, 30^\circ, 60^\circ]$ ,  $RY_{tool} = [-90^\circ, -40^\circ, 20^\circ, 0^\circ, 20^\circ, 40^\circ, 90^\circ]$  and  $RZ_{tool} = [-100^\circ, -50^\circ, 0^\circ, 50^\circ, 100^\circ, 150^\circ]$ ). Note that the "Roll" performance metric is similar to the one that we will be using to validate our 1DCAL approach (so called "Deviation" performance metric that will discuss in the next Chapter).

Our experimental results suggest that the presented approach is not only comparable but slightly better to that of MotoCal. The numerical solutions of the two systems (3DCAL and MotoCal) shown in Table 4.3 are similar for the most part except for the error parameter  $d_4$ . In terms of statistical mean error, it appears that the best results for "Step" ( $0.043 \pm 0.020$  mms) and "Cube" ( $0.185 \pm 0.099$  mms) are obtained using our approach and the one for "Roll" using MotoCal ( $0.207 \pm 0.109$  mms). We also present Table 4.4 that shows 3DCAL's and MotoCal's calibration results if one were to choose to calibrate the robot using the error parameters related to the joint variables only (i.e, calibrating using the DH(-)(-) model also referred to as calibrating the "home position"). By comparing the performance analyses presented in the two tables (Table 4.3 and Table 4.4), we affirm that most inaccuracy is due to erroneous estimate of the robot's home position as a slight deterioration in performance is noted.

To summarize the improvements in positioning accuracy, we present Table 4.5. The improvement percentages for 3DCAL and MotoCal are calculated using

$$Improvement\ percent = \frac{Before\ calibration - After\ calibration}{After\ calibration} \times 100.$$

The last column of the table shows the difference between 3DCAL and MotoCal improvement (i.e, our advancement/boost-in-performance compared to MotoCal.) As stated in the latter and as it can be clearly seen in the newly presented Table, our performance is slightly better compared to that of MotoCal for most of the cases. The only shortcoming of our 3DCAL approach is when it comes to the *End-Effector Roll* performance results for the MH5-Hi case. When interpreting the results of the experiments with the MH5-Hi, it is instructive to note that at that level of metrologic accuracy, the repeatability of the robot and measurement system are variables that can play a drastic role. Notwithstanding that, in general, it is safe to state that a comparable performance to that of MotoCal is attained. Based on the comparison of the positioning accuracy improvements achieved using DH(-) versus DH(-)(-) error models, for both robots, we re-affirm that most positioning inaccuracy is mostly due to misestimation of the home positions.

## 4.2 1DCAL Performance Analysis

In this Chapter, we present similar results as in Chapter 4.1 but for the 1DCAL approach described in Chapter 3.3. The radial measurements of 105 valid calibration points were acquired using the 1D CompuGauge measurement system described in Chapter 2.2. We used the same procedure as for our 3DCAL system to generate those points/configurations but using the MH6 manipulator. Table 4.6 summarizes the estimated values and corresponding performance upon completion of “Stage 2” of the calibration procedure. Note that we used a tool of known dimension (just like the MH5-Hi case) and hence elected to estimate

Table 4.5: Summary of the improvements in positioning accuracy using 3DCAL and MotoCal.

<b>HP20D case using DH(-) model and 3D CompuGauge.</b>			
<b>Alignment Test</b>	<b>3DCAL</b>	<b>MotoCal</b>	<b>Difference</b>
$\Delta X$	96.24%	96.15%	0.09%
$\Delta Y$	65.35%	64.46%	0.89%
$\Delta Z$	88.48%	87.52%	0.96%

<b>HP20D case using DH(-)(-) model and 3D CompuGauge.</b>			
<b>Alignment Test</b>	<b>3DCAL</b>	<b>MotoCal</b>	<b>Difference</b>
$\Delta X$	95.96%	95.98%	-0.02%
$\Delta Y$	47.27%	46.16%	1.11%
$\Delta Z$	77.60%	76.74%	0.86%

<b>MH5-Hi case using DH(-) model and CMM machine.</b>			
<b>Performance Metric</b>	<b>3DCAL</b>	<b>MotoCal</b>	<b>Difference</b>
<b>Step</b> <i>Mean error</i>	81.86%	79.32%	2.54%
<b>Cube</b> <i>Mean error</i>	77.41%	77.29%	0.12%
<b>Roll</b> <i>Mean error</i>	66.58%	72.87%	-6.29%

<b>MH5-Hi case using DH(-)(-) model and CMM machine.</b>			
<b>Performance Metric</b>	<b>3DCAL</b>	<b>MotoCal</b>	<b>Difference</b>
<b>Step</b> <i>Mean error</i>	81.86%	79.32%	2.54%
<b>Cube</b> <i>Mean error</i>	52.75%	50.31%	2.44%
<b>Roll</b> <i>Mean error</i>	60.16%	67.37%	-7.21%

the errors associated with the last link of the MH6 (i.e.  $A_6(q)$ ) as oppose to calibrating tool data as done with the HP20D case in our 3DCAL experiment.

The performance metric used here is based on the fact that given a well calibrated robot the Cartesian coordinates of a position should remain unchanged (or should change ever so slightly) as one varies the orientation of the tool frame. Hence, “Deviation” in Table 4.6 refers to the maximum discrepancy among the recorded radial measurements as we randomly varied the orientation of the tool for a given fixed position in space.

Table 4.6: 1DCAL and MotoCal “Stage 2” calibration results using 1D CompuGauge and the DH(-) error model of the MH6 manipulator.

Selected parameters	Nominal values	1DCAL	MotoCal
		Estimated values	Estimated values
$a_1$ in mm	150.000	148.140	N/A
$a_2$ in mm	614.000	614.381	614.748
$a_3$ in mm	155.000	155.281	N/A
$d_4$ in mm	640.000	640.534	N/A
$d_6$ in mm	95.000	94.855	94.827
$\delta\theta_2$ in degrees	0.000	-0.128	-0.199
$\delta\theta_3$ in degrees	0.000	0.046	-0.017
$\delta\theta_4$ in degrees	0.000	-0.117	-0.136
$\delta\theta_5$ in degrees	0.000	-0.915	-0.906
$\delta\theta_6$ in degrees	0.000	-0.837	-0.909
<b>Performance Analysis:</b>			
<i>Deviation in mm</i>	9.761	1.131	1.301

Table 4.7: 1DCAL and MotoCal “Stage 2” calibration results using 1D CompuGauge and the DH(-)(-) error model (home position) of the MH6 manipulator.

Selected parameters	Nominal values	1DCAL	MotoCal
		Estimated values	Estimated values
$\delta\theta_2$ in degrees	0.000	-0.188	-0.353
$\delta\theta_3$ in degrees	0.000	-0.083	0.074
$\delta\theta_4$ in degrees	0.000	-0.065	-0.125
$\delta\theta_5$ in degrees	0.000	-0.914	-0.907
$\delta\theta_6$ in degrees	0.000	-0.882	-0.855
<b>Performance Analysis:</b>			
<i>Deviation in mm</i>	9.761	1.372	1.434

To be precise, 34 random orientations that are bounded by the hardware limitations of measurement device and robot are chosen about all three principal axes. According to the

performance standard discussed in Chapter 2.2, if the deviation is below 2.0 mms, it is safe to assume that the robot is fairly well calibrated for most applications [5,6]. Before calibration, a deviation in the order of 9.761 mms is reported.

Based on our guidelines founded conceptually/analytically, the offset corresponding to the first joint angle ( $\delta\theta_1$ ) is only parameter that can not be calibrated in the case of the MH6 experiment. Thus, in this experiment, we are in favor of estimating all non-zero parameters except for  $\delta\theta_1$ . Note that we completed “Stage 1” for the purpose of cross validating/checking and we are pleased to make note that the auto-assessment does again conform with the “conceptual approach”. After executing “Stage 2” (calibration using the DH(-) model of the MH6 manipulator), we were able to reduce the deviation to 1.131 mms using our 1DCAL approach. The commercial system MotoCal can also calibrate a given robot with the supply of radial measurements [6]. In table 4.6, we compare our 1DCAL results and performance to that of MotoCal. Note that MotoCal’s algorithm has recommended not to consider 3 of the non-zero parameters shown in Table 4.6. Unlike the 3D case, our numerical results/solutions differ compared to the ones generated using MotoCal. This could be due to the extra 3 parameters in the case of 1DCAL. Note that in this particular scenario, our approach has outperformed MotoCal. To make a definite comparison to our best ability, we present Table 4.7 where only the joint angle offsets (i.e., calibration using the DH(-)(-) error model) are estimated. The results from both systems remain different but a comparable performance is attained.

We conclude this Chapter by presenting a similar Table like that 4.5 but with our 1D approach. Table 4.8 clearly shows that 1DCAL has outperformed MotoCal in both cases. By comparing the rows of the table, we state as a fact that indeed most of inaccuracy is due to incorrect joint variable offsets (i.e., calibration of the home position of the robot).

Table 4.8: Summary of the improvements in positioning accuracy using 1DCAL and MotoCal.

<b>Performance Metric and Error Models</b>	<b>3DCAL</b>	<b>MotoCal</b>	<b>Difference</b>
<i>Deviation using DH(-)model</i>	88.41%	86.67%	1.74%
<i>Deviation using DH(-)(-)model</i>	85.94%	85.31%	0.63%

### 4.3 Discussion

An issue that needs to be discussed when examining calibration algorithm performance is the sensitivity of the system to the number of calibration points. After experimenting, we found that as long as there are at least twice as many calibration points as calibration parameters (error degrees of freedom) both systems (using the recommended local solver over the course of the two calibration phases) converged to the same solution. It is instructive to note that at those varied positions, one needs to at least incorporate/include two different orientations (per calibration point).

Another point that needs to be discussed is how to efficiently define the solution space  $\Omega$  for the 3DCAL and 1DCAL approaches. A manipulator requires calibration for two main reasons: 1) if the accuracy of the robot (“out of the box”) is deemed inadequate (further improvement in accuracy is required by the application) or 2) if the robot has been repaired (after collision or parts replacement is performed.) Aforementioned, the dimensions of the solution space  $\Omega$  is manufacturer specific. For example, Yaskawa, Motoman Robotics Inc., manipulators are known to be fairly accurate “out of the box.” Hence if further improvement in accuracy is required a conservative restriction of the solution space is sufficient. To be specific, the angular parameter bounds of  $-e_i$  and  $\bar{e}_i$  (such as home position) may be limited to  $\pm 1^\circ$  and the length parameter bound may be

limited to  $\pm 1.0 \text{ mm}$ . robot manufacturers typically also provide a service for re-building and repairing their manipulators when needed (the second reason discussed in the latter). In this second scenario, it is first recommended that the user make use of the “pin alignment” technique to visually (with the “naked eye”) get a rough estimate of the home position. Then the user may employ the presented calibration algorithms subject to a comparatively wider solution space. That is, for this particular scenario we recommend that the angular parameter bounds of  $-e_i$  and  $\bar{e}_i$  ought to be limited to  $\pm 3^\circ$  and the length parameter bound may be limited to  $\pm 2 \text{ mm}$ . It is important to note that our recommendations here on how to set  $\Omega$  are based on Yaskawa, Motoman Robotics, Inc. manufacturing and repairing capabilities hence those bounds need to be revised accordingly as they are manipulator manufacturer and servicing quality dependent.

In addition of the specification of the solution space  $\Omega$  our 1DCAL approach requires the definition of the spherical volumetric solution space  $\mathbf{V}$ . Note that the discrepancy between the set of calculated values  $r$  and the set of measured values  $\tilde{r}$  is due to de-calibration. In fact,  $\Omega$  and  $\mathbf{V}$  are functions of the level of de-calibration. That is the reason why we recommend to set  $\Omega$  in accordance to the two scenarios discussed in the above. We recommend a more methodological approach to estimate the radius of  $\mathbf{V}$ . That is, the spherical volumetric solution space  $\mathbf{V}$  can be readily defined by computing its radius as

$$r_V = 2 \times \max_{r_V \in \mathbb{R}}(\mathbf{r} - \tilde{\mathbf{r}}),$$

where  $r$  and  $\tilde{r}$  are sets of calculated and measured values. We also propose that  $\mathbf{V}$  be centered at  $\mathbf{p}_{e0}^*$ .  $\mathbf{V}$  must include the location of the encoder and hence we believe it is wise to incorporate a safety factor (in particular since  $\mathbf{p}_{e0}^*$  is an initial guess). The largest

measurement discrepancy is multiplied by 2 in order to ensure that the actual position of the encoder is within the volumetric space  $V$ .

To compare the degree to which our hybrid optimization scheme (employed in “Stage 1” if the user chooses to make use of the “automated/numerical” approach) is successful in providing useful relevance assessment of the selected error parameters efficiently, we took on the task of implementing a similar optimization scheme. Namely, we looked at the “Multi-start” framework described in [53]. We made use of the scatter search method to generate potential starting points and employed TR for convergence purposes (i.e., we used different methods [SA versus Multi-start] to generate the starting points but employed an identical solver to analyze each initial condition). In order for that framework to convey a comparable evaluation (i.e., as indicative), a scatter search comprised of over 450 starting points is required. For example, referring back to Table 3.6, our technique (SA-TR based GS using 50 initial conditions) indicated a variance in the order of 2.263 for the error parameter  $\delta\theta_1$  upon completion of the first stage of calibration. Generating the same amount of starting points using the method described in [53] and performing a relevance evaluation on that parameter returns a variance analysis in the order of 0.981 only. Note that it also took a slightly higher number of iterations to converge for a given starting point (27 iterations on average using the same error deviation tolerance criterion). That particular parameter is obviously the most unstable for the reasons explained in Chapter 3.2 (i.e., a change in that parameter has null impact in the output of the objective function). We attribute our higher-ranking performance in this domain (i.e, relevance evaluation and convergence rate) to SA’s ability to not only globally explore the solution sub-space thoroughly but as illustrated in Figure 3.6 to output a result/starting-point that is closer to the minima of the cost function.



Using the discussed assessment methodology, one can take the advantage of the opportunity to conduct an empirical study to possibly/up-to-some-extent characterize the various objective functions associated with the DH(-) error models. For illustration purposes, Table 4.9 depicts the same results as in Table 4.3 along with the variance analysis of each error parameter (i.e., the3DCAL and MotoCal “Stage 2” calibration results using CMM and the DH(-) error model of the MH5-Hi manipulator along with the automated/numerical assessment of each parameter deemed salient). It is very interesting to note that all 50 potential starting points converged to the same numerical solution for the defined/bounded search space  $\Omega$  (i.e., the automated/numerical evaluation indicates no variance). That is likely due to the fact that after the completion of the first stage, almost all redundancy in the error model has been eliminated and the cost function associated with that model may be well-conditioned. That in turn could imply that the objective function exhibits *locally* (within  $\Omega$  a convex like characteristic. Hence the DH(-) error models could warrant a reliable calibration results as this check proves that the obtained multiple numerical solutions are consistent regardless the initial position of a widespread starting points. Note that this was found to be true for all other calibration cases as well. Also to note is that identical results were seen when employing the multi-start framework (instead of SA) to generate a different set of “casual” starting points that is significantly higher in number. Apropos MotoCal was unable to calibrate  $\delta\theta_6$  even though its own parametric assessment indicated that the parameter is stable. Hence, we believe our approach is indeed more robust compared to that of the available state-of-the-art commercial calibration system Motocal.

We also need to address the issue of processing speed. The system consumes approximately on average 271 seconds if the user opt to execute “Stage 1” of the calibration procedure

Table 4.9: 3DCAL and MotoCal “Stage 2” calibration results using CMM and the DH(-) error model of the MH5-Hi manipulator along with the automated/numerical assessment of each parameter deemed salient.

Selected parameters	Nominal values	3DCAL		MotoCal
		Estimated values	Variance	Estimated values
$a_1$ in mm	88.000	87.864	0.001	N/A
$a_2$ in mm	310.000	310.391	0.000	310.354
$a_3$ in mm	40.000	40.003	0.002	40.064
$d_4$ in mm	305.000	305.034	0.001	304.925
$d_6$ in mm	80.000	79.824	0.000	N/A
$\delta\theta_2$ in degrees	0.000	-0.159	0.000	-0.144
$\delta\theta_3$ in degrees	0.000	0.169	0.000	0.167
$\delta\theta_4$ in degrees	0.000	0.333	0.000	0.325
$\delta\theta_5$ in degrees	0.000	0.049	0.000	0.063
$\delta\theta_6$ in degrees	0.000	-0.209	0.000	N/A
<b>Performance Analysis:</b>				
<b>Step</b>				
Meanerror in mm	0.237	0.043		0.049
Minimumerror in mm	0.018	0.007		0.015
Maximumerror in mm	0.510	0.080		0.088
Spread in mm	0.492	0.071		0.073
StandardDeviation in mm	0.131	0.018		0.019
<b>Cube</b>				
Meanerror in mm	0.819	0.185		0.186
Minimumerror in mm	0.292	0.039		0.049
Maximumerror in mm	1.650	0.376		0.351
Spread in mm	1.357	0.331		0.302
StandardDeviation in mm	0.419	0.099		0.090
<b>End-Effector Roll</b>				
Meanerror in mm	0.763	0.255		0.207
Minimumerror in mm	0.354	0.034		0.042
Maximumerror in mm	1.196	0.524		0.430
Spread in mm	0.842	0.472		0.388
StandardDeviation in mm	0.266	0.125		0.109

(employing TR as local optimizers and using all 50 starting points obtained by way of SA). “Stage 2” of the calibration procedure using DH(-) and DH(-)(-) models consumes on average 7.6 and 6.8 seconds, respectively, given a single arbitrary starting point. The work is implemented in MATLAB on an HP Personal Computer (PC) with processor speed of 3.22 GHz. The obvious thing to note is that the reported temporal rates are subject to the amount of data/calibration-points used to calibrate the robot in question. Also to note is that the calculated time consumption appears to be less on average in the case of the 1DCAL calibration system (in comparison with 3DCAL). In order to accelerate the automated/numerical approach of “Stage 1”, we would like to refer the reader to a previous work of the current author [96]. In that work, we have proven that as long as processes are parallel, they can be greatly accelerated. The process related to each starting point is independent from another (i.e., parallel). And hence, those processes can be simultaneously done on a platform like the one described in [96]. We acknowledge that the scatter search method that requires the 450 processes for a good parameter assessment can also be implemented in that type of platform and hence can also be greatly accelerated (i.e., the 450 processes can be conducted simultaneously). However, we believe our method is advantageous to some degree since it ensures up to some extent a faster convergence rate (i.e., fewer number of iterations steps taken by the local solvers since initial conditions are found via SA and can be comparatively/somewhat closer to solution[s]).

In order to further boost the computationally efficiency of our 1DCAL algorithm, we considered making use of a systematic approach to reduce the dimensionality of the combined search space comprised of  $\Omega$  and  $V$ . A block diagram of this attempt is shown in Figure 4.2. The idea was to first estimate the location of the encoder and use that optimal estimate to solve for the DH error parameters. To first estimate the coordinates/location

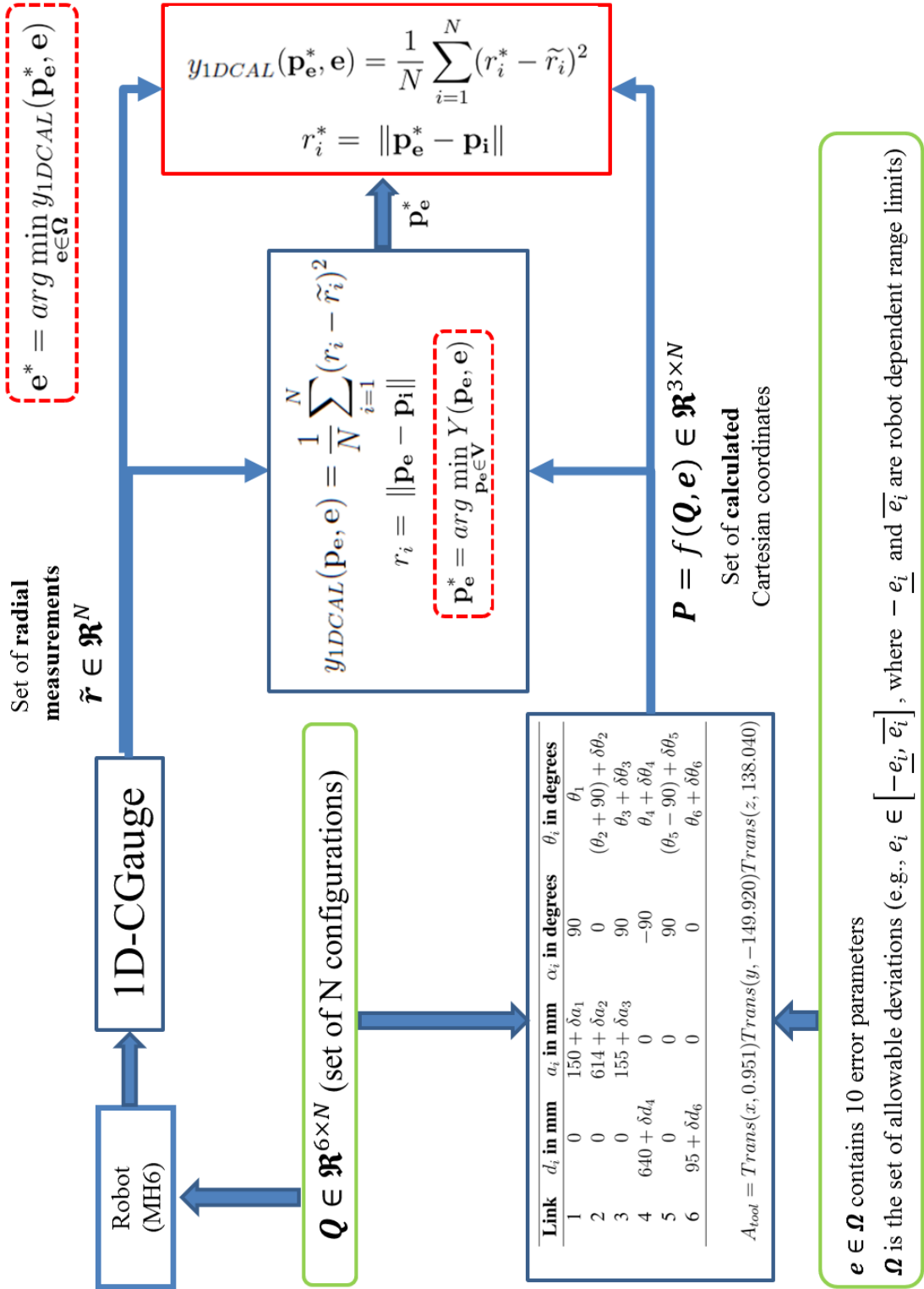


Figure 4.2: Alternate 1DCAL approach block diagram where the DH(-) model of the MH6 manipulator is used as an example.

of the fixed reference (encoder)  $\mathbf{p}_e = [\mathbf{x}_e, \mathbf{y}_e, \mathbf{z}_e]^T \in \mathbf{V}$ , we made use of the objective function  $y_{1DCAL}(\mathbf{p}_e, \mathbf{e})$  described in Eq. (3.11) of Chapter 3.3 and we minimized that function such that a minimizer

$$\mathbf{p}_e^* = \mathit{arg} \min_{\mathbf{p}_e \in \mathbf{V}} y_{1DCAL}(\mathbf{p}_e, \mathbf{e}). \quad (4.1)$$

is found. We made use of TR to solve for  $\mathbf{p}_e^*$  since it has proven to be efficient compared to LM. Note that finding the solution  $\mathbf{p}_e^*$  is a sub-procedure. As it is illustrated in Figure 4.2, We make use of that solution to minimize the objective function described in Eq. (3.11) (i.e., to minimize  $y_{1DCAL}(\mathbf{p}_e^*, \mathbf{e})$ ) so as to solve for the DH error parameters. In other words, the cost function used to estimate the error parameters can be re-written as

$$y_{1DCAL}(\mathbf{p}_e^*, \mathbf{e}) = \frac{1}{N} \sum_{i=1}^N (r_i^* - \tilde{r}_i)^2 \quad (4.2)$$

where  $r_i^*$  is the computed radial distance using Eq. (3.12) of Chapter 3.3 but differs from  $r_i$  because the minimizer  $\mathbf{p}_e^*$  obtained from Eq. (4.1) is used to carry out that calculation. In other words,  $r_i^*$  is calculated as follow

$$r_i^* = \|\mathbf{p}_e^* - \mathbf{p}_i\|.$$

In essence, we have a cyclic coordinate descent [97–99] based optimization scheme where  $\mathbf{p}_e^*$  is refined via Eq. (4.1) at each iterative step taken to minimize the cost function described in Eq. (4.2) such that a minimizer

$$\mathbf{e}^* = \mathit{arg} \min_{\mathbf{e} \in \Omega} y_{1DCAL}(\mathbf{p}_e^*, \mathbf{e})$$

is calculated. Note that TR is also put to use to solve for the error parameter vector  $\mathbf{e}^*$ . Even though the original 1DCAL and the newly discussed cyclic coordinate descent based version of it converged to the same solution, no significant improvement in processing

speed is attained. We attribute the comparable performance in speed of the original 1DCAL to the initial step of estimating the starting point  $\mathbf{p}_{e0}^*$  before optimizing over Eq. (3.11) in order to find the minimizer described in Eq. (3.13) of Chapter 3.3 at a fast/comparable rate.

We are in the process of exploring higher-quality metrology equipments (compared to CompuGauge and/or CMM) in order to evaluate the peak performance and limitations of the presented frameworks. We also believe that extending the presented simplistic approach so as to include other sources of error is another potentially fruitful area of future research (i.e., to devise non-redundant error models that take into account parameters related to elasticity of joints and links, load proportion and joint deflection, etc.). It is also our hope to investigate computationally efficient kinematic and dynamic related calibration schemes for redundant manipulators (i.e., robots that possess more than six degrees of freedom [like the Yaskawa Motoman Robotics, Inc. IA20 [100] that has 7 revolute joints]).

## CHAPTER 5

### CONCLUSION

In this dissertation, we have presented new computational efficient kinematic calibration algorithms for industrial robots. These include the 3DCAL system that uses Cartesian coordinates of the calibration points and 1DCAL approach that requires the supply of radial measurements only of the calibration points. Both methods make use of an error model that is based on the original DH convention. For both systems the calibration procedure calls for two stages. The first stage that consists of deducing the correct DH(-) model of a manipulator can be achieved either using a conceptual/analytical or an automated/numerical approach. Note that “the automated approach” does increase the burden on the user (running an optimization-scheme/Global-Search using available data) but is undoubtedly helpful in deriving the DH(-) model of the robot in question (in particular, it can greatly help if the geometry of the structure is complex and the guidelines of the “analytical approach” can no longer be extended in an obvious fashion). After completing “Stage 1” (i.e., Removing ineffectual error parameters that bring about unwanted redundancy such that to correctly construct the DH(-) model), we use TR to carry out the second stage of the calibration process. We have also introduced DH(-)(-) error model where the home positions (and tool(s) if needed) only of the robots are calibrated. Note that those two models are less complex (compared to the numerous error models discussed in Chapter 1) and hence

do not require burdensome computation. Although incomplete, in this work, we have proven that those simplified error models (DH(-) and DH(-)(-)) in conjunction with the proposed optimization scheme (TR) are capable of exceptional performance comparable to that of an existing commercial system. Using our 3DCAL, we are able to successfully calibrate the Yaskawa Motoman Robotics, Inc. 6DOF serial manipulator HP20D to a satisfactory degree for most applications (according to performance standard and metric described in Chapter 2.2) and the MH5-Hi to a higher degree of accuracy (possibly the best achievable manipulator accuracy by means of geometric/kinematic calibration). Similarly, using the presented 1DCAL approach and according to the performance standard described in Chapter 4, we have also adequately calibrated another similar robot; the MH6. Based on our experimental results, we have concluded that most of positioning inaccuracy is due to the error parameters related to joint angles (i.e., home position of robot), at least when it comes to Yaskawa Motoman Robotics, Inc. manipulators like the HP20D, MH5-Hi and MH6. And hence, the DH(-)(-) error model appears to be sufficient in improving positioning accuracy for most applications where those robots are put to use.

We make use of novel global optimization technique in our automated/numerical approach used in “Stage 1” to effectively target unstable and/or redundant error parameters. It consists of first gathering a rich set of potential starting points using SA. We then employ the local solver like TR using those starting points to carry out a GS for the solution. To our best knowledge, we have not seen that done in any other work. We found our optimization technique not only to be beneficial from a computational stand point of view (compared to other global search mechanism that call for a higher number of starting points) but to also be very well suited for the first stage in this application as it can provide an automated parametric relevance assessment to the end-user. Based on that



insight, the user can easily identify the ineffectual error parameters and construct the DH(-) error model for a given manipulator. We have conducted numerous experiments that show that the automated evaluation (variance analysis) is consistent with the guidelines that we analytically established in order to identify irrelevant parameters and to render the model non-redundant (i.e., to correctly construct the DH(-) model of the manipulator)

## BIBLIOGRAPHY

- [1] Yaskawa Motoman Robotics, Inc., *HP20D Series: Technical Specifications*, Yaskawa Motoman Robotics Inc., 2012. [Online]. Available: [www.motoman.com/datasheets/HP20D\\\_HP20D-6.pdf/](http://www.motoman.com/datasheets/HP20D\_HP20D-6.pdf/)
- [2] ———, “*DX100 Controller Manual*”, *Part Number:156431-1CD*, Yaskawa Motoman Robotics Inc., Yaskawa America, Inc., Motoman Robotics Division, Headquarters and Manufacturing Facility, 100 Automation Way, Miamisburg, OH 45342, United States, 2011. [Online]. Available: [www.motoman.com/](http://www.motoman.com/)
- [3] ———, “*FS100 Controller Manual*”, *Part Number:159550-1CD*, Yaskawa Motoman Robotics Inc., Yaskawa America, Inc., Motoman Robotics Division, Headquarters and Manufacturing Facility, 100 Automation Way, Miamisburg, OH 45342, United States, 2011. [Online]. Available: [www.motoman.com/](http://www.motoman.com/)
- [4] ———, *MH6 Series: Technical Specifications*, Yaskawa Motoman Robotics Inc., 2011. [Online]. Available: [www.motoman.com/datasheets/MH6\\\_MH6S.pdf/](http://www.motoman.com/datasheets/MH6\_MH6S.pdf/)
- [5] Dynalog Inc., *3-D COMPUGAUGE Manual (version 4.0)*, Compugauge: Dynalog Inc., Dynalog, Inc. 6001 1 N. Adams Road, Suite 125, Bloomfield Hills, MI 48304, United States., 2002. [Online]. Available: [www.dynalog-us.com/](http://www.dynalog-us.com/)

- [6] Yaskawa Motoman Robotics, Inc., *MotoSoft: "MotoCal 3.0.1 User's Manual"*, Part Number:140724-1, Revision:2, Yaskawa Motoman Robotics Inc., Yaskawa America, Inc., Motoman Robotics Division, Headquarters and Manufacturing Facility, 100 Automation Way, Miamisburg, OH 45342, United States., 2007. [Online]. Available: [www.motoman.com/](http://www.motoman.com/)
- [7] S. A. Hayati, "Robot arm geometric link parameter estimation," in *Decision and Control, 1983. The 22nd IEEE Conference on*, vol. 22. IEEE, 1983, pp. 1477–1483.
- [8] S. Hayati and M. Mirmirani, "Improving the absolute positioning accuracy of robot manipulators," *Journal of Robotic Systems*, vol. 2, no. 4, pp. 397–413, 1985.
- [9] Y. E. Corp. and S. Corp., "*MotoSimEG*", Version 3.50, Yaskawa Motoman Robotics Inc., Yaskawa America, Inc., Motoman Robotics Division, Headquarters and Manufacturing Facility, 100 Automation Way, Miamisburg, OH 45342, United States, 2009. [Online]. Available: [www.motoman.com/](http://www.motoman.com/)
- [10] S. Y. Nof, *Handbook of industrial robotics*. John Wiley & Sons, 1999, vol. 1.
- [11] B. W. Mooring, Z. S. Roth, and M. R. Driels, *Fundamentals of manipulator calibration*. Wiley New York, 1991.
- [12] A. Nubiola and I. A. Bonev, "Absolute calibration of an abb irb 1600 robot using a laser tracker," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 1, pp. 236–245, 2013.
- [13] P. G. Mikell, R. Mitchell Weis, and N. G. O. N Nagel, "Industrial robotics technology, programming and applications," 1986.

- [14] B. Horn, *Robot Vision*. The MIT Press, 1986.
- [15] A. Nubiola and I. A. Bonev, “Absolute robot calibration with a single telescoping ballbar,” *Precision Engineering*, 2014.
- [16] B. Karan and M. Vukobratović, “Calibration and accuracy of manipulation robot modelsan overview,” *Mechanism and Machine Theory*, vol. 29, no. 3, pp. 479–500, 1994.
- [17] Y. Andrew Liou, P. P. Lin, R. R. Lindeke, and H.-D. Chiang, “Tolerance specification of robot kinematic parameters using an experimental design technique the taguchi method,” *Robotics and computer-integrated manufacturing*, vol. 10, no. 3, pp. 199–207, 1993.
- [18] M. Abderrahim, A. Khamis, S. Garrido, and L. Moreno, “Accuracy and calibration issues of industrial manipulators,” *Industrial Robotics: Programming, Simulation and Applications*, 2006.
- [19] A. Elatta, L. P. Gen, F. L. Zhi, Y. Daoyuan, and L. Fei, “An overview of robot calibration,” *Information Technology Journal*, vol. 3, no. 1, pp. 74–78, 2004.
- [20] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*. Springer Verlag, 2000.
- [21] C. Gong, J. Yuan, and J. Ni, “Nongeometric error identification and compensation for robotic system by inverse calibration,” *International Journal of Machine Tools and Manufacture*, vol. 40, no. 14, pp. 2119–2137, 2000.

- [22] M. R. Driels and U. S. Pathre, "Robot calibration using an automatic theodolite," *The International Journal of Advanced Manufacturing Technology*, vol. 9, no. 2, pp. 114–125, 1994.
- [23] J. M. S. Motta, G. C. de Carvalho, and R. McMaster, "Robot calibration using a 3d vision-based measurement system with a single camera," *Robotics and Computer-Integrated Manufacturing*, vol. 17, no. 6, pp. 487–497, 2001.
- [24] R. P. Paul, *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. Richard Paul, 1981.
- [25] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons New York, 2006.
- [26] J. Robert, "Fundamentals of Robotics: Analysis and Control," *Canada, Prentice-Hall*, 1990.
- [27] J. J. Craig, *Introduction to robotics*. Addison-Wesley Reading, MA, 1989, vol. 7.
- [28] M. Brady, *Robot motion: Planning and control*. MIT press, 1982.
- [29] M. Abderrahim and A. Whittaker, "Kinematic model identification of industrial manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 16, no. 1, pp. 1–8, 2000.
- [30] X. Yang, L. Wu, J. Li, and K. Chen, "A minimal kinematic model for serial robot calibration using poe formula," *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 3, pp. 326–334, 2014.

- [31] J. Denavit, "A kinematic notation for lower-pair mechanisms based on matrices." *Trans. of the ASME. Journal of Applied Mechanics*, vol. 22, pp. 215–221, 1955.
- [32] D. Whitney, C. Lozinski, and J. M. Rourke, "Industrial robot forward calibration method and results," *Journal of dynamic systems, measurement, and control*, vol. 108, no. 1, pp. 1–8, 1986.
- [33] H. W. Stone and A. C. Sanderson, "A prototype arm signature identification system," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4. IEEE, 1987, pp. 175–182.
- [34] H. W. Stone, *Kinematic modeling, identification, and control of robotic manipulators*. Springer, 1987, vol. 29.
- [35] W. Veitschegger and C.-H. Wu, "Robot accuracy analysis based on kinematics," *Robotics and Automation, IEEE Journal of*, vol. 2, no. 3, pp. 171–179, 1986.
- [36] H. Zhuang, Z. S. Roth, and F. Hamano, "A complete and parametrically continuous kinematic model for robot manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 4, pp. 451–463, 1992.
- [37] H. Zhuang and Z. S. Roth, "Robot calibration using the cpc error model," *Robotics and computer-integrated manufacturing*, vol. 9, no. 3, pp. 227–237, 1992.
- [38] K. Okamura and F. Park, "Kinematic calibration using the product of exponentials formula," *Robotica*, vol. 14, no. 4, pp. 415–422, 1996.
- [39] R. He, Y. Zhao, S. Yang, and S. Yang, "Kinematic-parameter identification for serial-robot calibration based on poe formula," *Robotics, IEEE Transactions on*, vol. 26, no. 3, pp. 411–423, 2010.

- [40] X. Yang, L. Wu, J. Li, and K. Chen, “A minimal kinematic model for serial robot calibration using poe formula,” *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 3, pp. 326–334, 2014.
- [41] Yaskawa Motoman Robotics, Inc., “*NX100 Controller Manual*”, Part Number:149201-1, Yaskawa Motoman Robotics Inc., Yaskawa America, Inc., Motoman Robotics Division, Headquarters and Manufacturing Facility, 100 Automation Way, Miamisburg, OH 45342, United States, 2007. [Online]. Available: [www.motoman.com/](http://www.motoman.com/)
- [42] G. Du and P. Zhang, “Online robot calibration based on vision measurement,” *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 6, pp. 484–492, 2013.
- [43] S. Kirkpatrick, “Optimization by simulated annealing: Quantitative studies,” *Journal of statistical physics*, vol. 34, no. 5-6, pp. 975–986, 1984.
- [44] L. Davis, “Genetic algorithms and simulated annealing,” 1987.
- [45] P. J. Van Laarhoven and E. H. Aarts, *Simulated annealing*. Springer, 1987.
- [46] E. Aarts and J. Korst, “Simulated annealing and boltzmann machines: a stochastic approach to combinatorial optimization and neural computing,” 1988.
- [47] T. R. Tuinstra, “Automatic segmentation of small pulmonary nodules in computed tomography data using a radial basis function neural network with application to volume estimation,” Ph.D. dissertation, University of Dayton, 2008.
- [48] J. J. Moré and D. C. Sorensen, “Computing a trust region step,” *SIAM Journal on Scientific and Statistical Computing*, vol. 4, no. 3, pp. 553–572, 1983.

- [49] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, “Approximate solution of the trust region problem by minimization over two-dimensional subspaces,” *Mathematical programming*, vol. 40, no. 1-3, pp. 247–263, 1988.
- [50] T. F. Coleman and Y. Li, “An interior trust region approach for nonlinear minimization subject to bounds,” *SIAM Journal on optimization*, vol. 6, no. 2, pp. 418–445, 1996.
- [51] T. Steihaug, “The conjugate gradient method and trust regions in large scale optimization,” *SIAM Journal on Numerical Analysis*, vol. 20, no. 3, pp. 626–637, 1983.
- [52] K. M. Passino, *Biomimicry for optimization, control, and automation*. Springer, 2005.
- [53] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Martí, “Scatter search and local nlp solvers: A multistart framework for global optimization,” *INFORMS Journal on Computing*, vol. 19, no. 3, pp. 328–340, 2007.
- [54] F. Glover, “A template for scatter search and path relinking,” in *Artificial evolution*. Springer, 1998, pp. 1–51.
- [55] L. Dixon and G. Szegö, “The global optimization problem: an introduction,” *Towards global optimization*, vol. 2, pp. 1–15, 1978.
- [56] Yaskawa Motoman Robotics, Inc., *Motoman MH5: General Purpose and Handling*, Yaskawa Motoman Robotics Inc., 2011. [Online]. Available: [www.motoman.co.uk/uploads/tx\\\_catalogrobot/MH5\\\_E.pdf](http://www.motoman.co.uk/uploads/tx\_catalogrobot/MH5\_E.pdf)



- [57] K. Lau, R. Hocken, and W. Haight, "Automatic laser tracking interferometer system for robot metrology," *Precision Engineering*, vol. 8, no. 1, pp. 3–8, 1986.
- [58] A. Nubiola, M. Slamani, and I. A. Bonev, "A new method for measuring a large set of poses with a single telescoping ballbar," *Precision Engineering*, vol. 37, no. 2, pp. 451–460, 2013.
- [59] B. Mooring and S. Padavala, "The effect of kinematic model complexity on manipulator accuracy," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*. IEEE, 1989, pp. 593–598.
- [60] G. Puskorius and L. Feldkamp, "Global calibration of a robot/vision system," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4. IEEE, 1987, pp. 190–195.
- [61] Y. Meng and H. Zhuang, "Self-calibration of camera-equipped robot manipulators," *The International Journal of Robotics Research*, vol. 20, no. 11, pp. 909–921, 2001.
- [62] C. S. Gatla, R. Lumia, J. Wood, and G. Starr, "An automated method to calibrate industrial robots using a virtual closed kinematic chain," *Robotics, IEEE Transactions on*, vol. 23, no. 6, pp. 1105–1116, 2007.
- [63] F. Boochs, R. Schutze, C. Simon, F. Marzani, H. Wirth, and J. Meier, "Increasing the accuracy of untaught robot positions by means of a multi-camera system," in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*. IEEE, 2010, pp. 1–9.
- [64] J. H. Jang, S. H. Kim, and Y. K. Kwak, "Calibration of geometric and non-geometric errors of an industrial robot," *Robotica*, vol. 19, no. 3, pp. 311–321, 2001.

- [65] T.-F. Lu and G. C. Lin, "An on-line relative position and orientation error calibration methodology for workcell robot operations," *Robotics and Computer-Integrated Manufacturing*, vol. 13, no. 2, pp. 89–99, 1997.
- [66] R. P. Judd and A. B. Knasinski, "A technique to calibrate industrial robots with experimental verification," *Robotics and Automation, IEEE Transactions on*, vol. 6, no. 1, pp. 20–30, 1990.
- [67] J. Chen and L.-M. Chao, "Positioning error analysis for robot manipulators with all rotary joints," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 6, pp. 539–545, 1987.
- [68] J. F. Jarvis, "Microsurveying: towards robot accuracy," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4. IEEE, 1987, pp. 1660–1665.
- [69] H. W. Stone, A. C. Sanderson, and C. P. Neuman, "Arm signature identification," in *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol. 3. IEEE, 1986, pp. 41–48.
- [70] W. Khalil and S. Besnard, "Geometric calibration of robots with flexible joints and links," *Journal of Intelligent and Robotic systems*, vol. 34, no. 4, pp. 357–379, 2002.
- [71] G. Schiele, R. Hofmann, T. v. Diep, H. van Kunzmann, F. Wäldele, and K. Busch, "Transportable measuring instrument for testing the positional accuracy of a program-controlled appliance arm," *German Patent No. DE*, vol. 3504464, p. C1, 1986.

- [72] N. Vira and K. Lau, "An extensible ball bar for evaluation of robots' positioning performance," *Journal of robotic systems*, vol. 4, no. 6, pp. 799–814, 1987.
- [73] M. Driels, "Using passive end-point motion constraints to calibrate robot manipulators," *Journal of dynamic systems, measurement, and control*, vol. 115, no. 3, pp. 560–566, 1993.
- [74] N. Juneja and A. A. Goldenberg, "Kinematic calibration of a re-configurable robot (robotwin)," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4. IEEE, 1997, pp. 3178–3183.
- [75] H. Ota, T. Shibukawa, T. Tooyama, and M. Uchiyama, "Forward kinematic calibration and gravity compensation for parallel-mechanism-based machine tools," *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, vol. 216, no. 1, pp. 39–49, 2002.
- [76] L. Beyer and J. Wulfsberg, "Practical robot calibration with rosy," *Robotica*, vol. 22, no. 5, pp. 505–512, 2004.
- [77] H. Zhuang, L. K. Wang, and Z. S. Roth, "Error-model-based robot calibration using a modified cpc model," *Robotics and computer-integrated manufacturing*, vol. 10, no. 4, pp. 287–299, 1993.
- [78] A. Nubiola, M. Slamani, A. Joubair, and I. A. Bonev, "Comparison of two calibration methods for a small industrial robot based on an optical cmm and a laser tracker," *Robotica*, pp. 1–20, 2013.

- [79] A. Goswami, A. Quaid, and M. Peshkin, "Complete parameter identification of a robot from partial pose information," in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on.* IEEE, 1993, pp. 168–173.
- [80] L. Foulloy and R. B. Kelley, "Improving the precision of a robot," in *Robotics and Automation. Proceedings. 1984 IEEE International Conference on*, vol. 1. IEEE, 1984, pp. 62–67.
- [81] W. Veitschegger and C.-H. Wu, "A method for calibrating and compensating robot kinematic errors," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4. IEEE, 1987, pp. 39–44.
- [82] T.-W. Hsu and L. J. Everett, "Identification of the kinematic parameters of a robot manipulator for positional accuracy improvement," in *Proceedings of the Computation in Engineering Conference*, 1985, pp. 263–267.
- [83] K. Levenberg, "A method for the solution of certain problems in least squares," *Quarterly of applied mathematics*, vol. 2, pp. 164–168, 1944.
- [84] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
- [85] J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Numerical analysis.* Springer, 1978, pp. 105–116.
- [86] H. Zhuang, "Self-calibration of parallel mechanisms with a case study on stewart platforms," *Robotics and Automation, IEEE Transactions on*, vol. 13, no. 3, pp. 387–397, 1997.

- [87] J. Gentle, “Matrix algebra: Theory,” *Computations, and Applications in Statistics*, 2007.
- [88] A. Jeffrey, *Advanced engineering mathematics*. Academic Press, 2001.
- [89] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.
- [90] D. Lay, *Linear algebra and its applications*. Pearson Education, Inc., 2003.
- [91] F. V. Berghen, “Levenberg-marquardt algorithms vs trust region algorithms,” *IRIDIA, Université Libre de Bruxelles*, 2004.
- [92] J.-y. Fan, “A modified levenberg-marquardt algorithm for singular system of nonlinear equations,” *JOURNAL OF COMPUTATIONAL MATHEMATICS-INTERNATIONAL EDITION-*, vol. 21, no. 5, pp. 625–636, 2003.
- [93] J.-y. Fan and Y.-x. Yuan, “On the quadratic convergence of the levenberg-marquardt method without nonsingularity assumption,” *Computing*, vol. 74, no. 1, pp. 23–39, 2005.
- [94] G. Alici and B. Shirinzadeh, “Enhanced stiffness modeling, identification and characterization for robot manipulators,” *Robotics, IEEE Transactions on*, vol. 21, no. 4, pp. 554–564, 2005.
- [95] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [96] T. Messay, C. Chen, R. Ordóñez, and T. M. Taha, “Gpgpu acceleration of a novel calibration method for industrial robots,” in *Aerospace and Electronics Conference (NAECON), Proceedings of the 2011 IEEE National*. IEEE, 2011, pp. 124–129.

- [97] Z.-Q. Luo and P. Tseng, “On the convergence of the coordinate descent method for convex differentiable minimization,” *Journal of Optimization Theory and Applications*, vol. 72, no. 1, pp. 7–35, 1992.
- [98] A. Saha and A. Tewari, “On the nonasymptotic convergence of cyclic coordinate descent methods,” *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 576–601, 2013.
- [99] Y. Nesterov, “Efficiency of coordinate descent methods on huge-scale optimization problems,” *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.
- [100] Yaskawa Motoman Robotics, Inc., *IA20 Manipulator Manual.*, Yaskawa Motoman Robotics Inc., 2007. [Online]. Available: [www.motoman.com/motomedia/manuals/docs/151529-1.pdf](http://www.motoman.com/motomedia/manuals/docs/151529-1.pdf)

## APPENDIX A

### Geometry of the HP20D manipulator [1, 2].

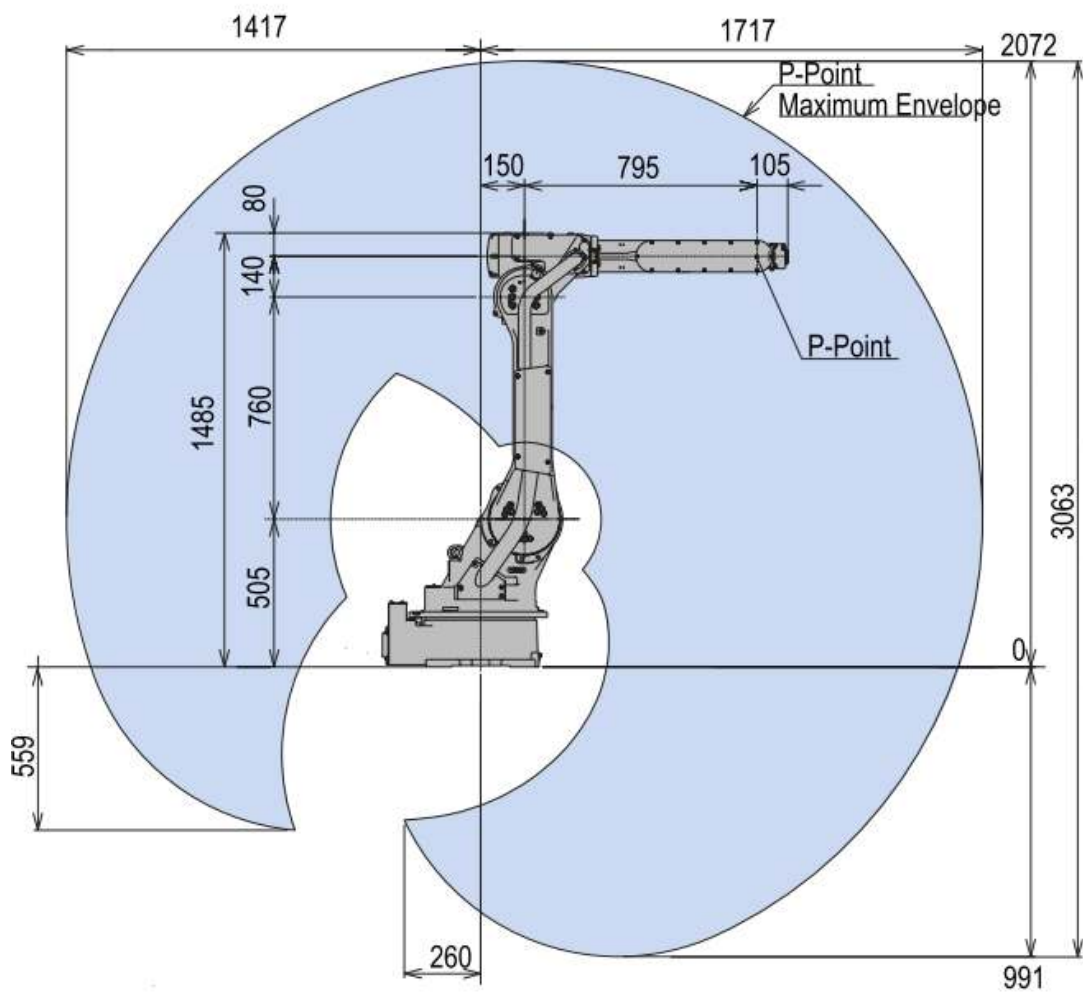


Figure A.1: Geometry of the HP20D manipulator. All dimensions are metric (mm).

## APPENDIX B

### Geometry of the MH5 manipulator [1, 3].

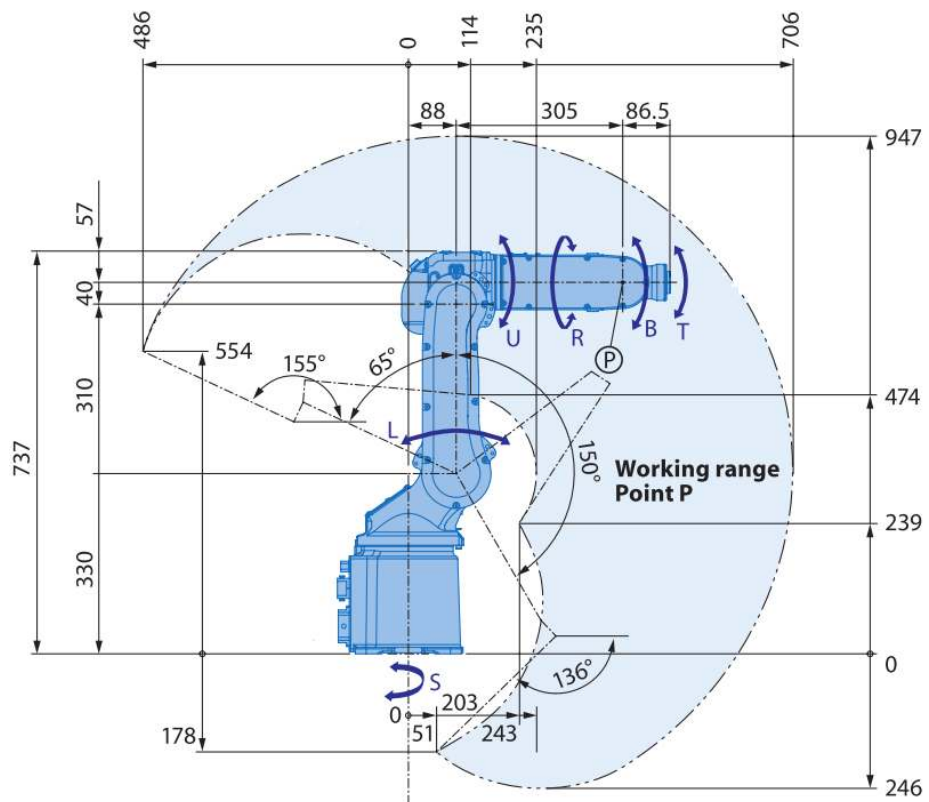


Figure B.1: Geometry of the MH5 manipulator. All dimensions are metric (mm and degrees).



## APPENDIX C

### Geometry of the MH6 manipulator [2, 4].

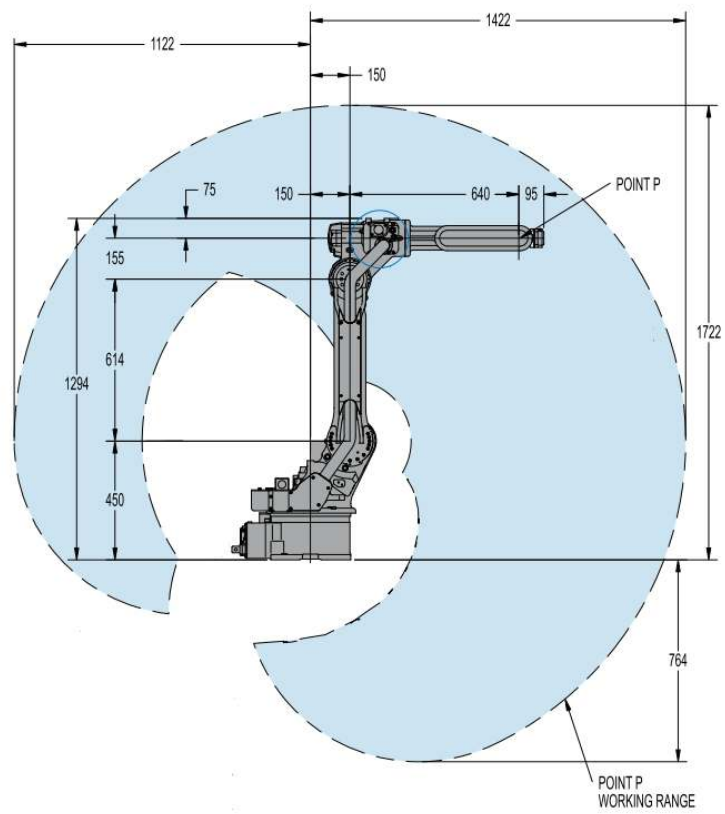


Figure C.1: Geometry of the MH6 manipulator. All dimensions are metric (mm).

## APPENDIX D

### Expressional details of homogeneous transformation matrices.

$$Rot(x, \alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rot(z, \theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Trans(z, d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Trans(x, a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_i = Rot(z, \theta_i)Trans(z, d_i)Trans(x, a_i)Rot(x, \alpha_i)$$

$$A_i = \begin{bmatrix} c(\theta_i) & -s(\theta_i)c(\alpha_i) & s(\theta_i)s(\alpha_i) & (a_i)c(\theta_i) \\ s(\theta_i) & c(\theta_i)c(\alpha_i) & -c(\theta_i)s(\alpha_i) & a_i s(\theta_i) \\ 0 & s(\alpha_i) & c(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$RPY(RX_{tool}, RY_{tool}, RZ_{tool}) =$$

$$\begin{bmatrix} c(RZ_{tool})c(RY_{tool}) & c(RZ_{tool})s(RX_{tool})s(RY_{tool}) - c(RX_{tool})s(RZ_{tool}) & & & \\ c(RY_{tool})s(RZ_{tool}) & c(RZ_{tool})c(RX_{tool}) + s(RZ_{tool})s(RX_{tool})s(RY_{tool}) & & & \dots \\ -s(RY_{tool}) & & c(RY_{tool})s(RX_{tool}) & & \\ 0 & & 0 & & \\ & & & s(RZ_{tool})s(RX_{tool}) + c(RZ_{tool})c(RX_{tool})s(RY_{tool}) & 0 \\ & \dots & & c(RX_{tool})s(RZ_{tool})s(RY_{tool}) - c(RZ_{tool})s(RX_{tool}) & 0 \\ & & & c(RX_{tool})c(RY_{tool}) & 0 \\ & & & 0 & 1 \end{bmatrix}$$

where  $c()$  and  $s()$  are the short hands of  $\cos()$  and  $\sin()$ , respectively.

# APPENDIX E

## Graphical User Interface (GUI).

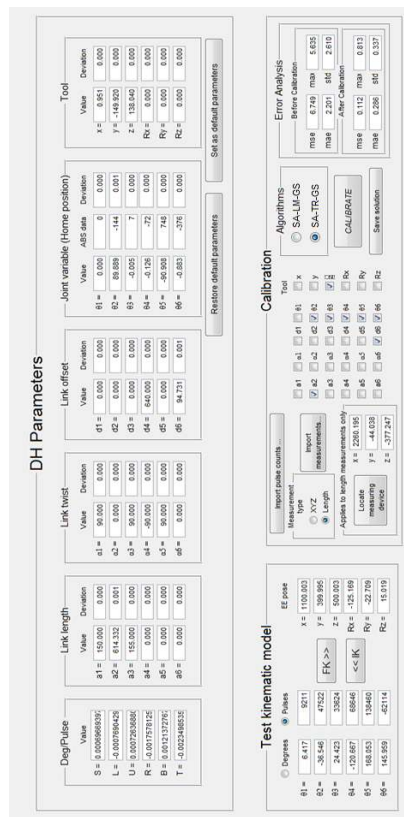


Figure E.1: Graphical User Interface (GUI) for the proposed calibration framework.