Kinematic Calibration of Six-Axis Serial Robots Using the Relative Measurement Concept

by

Nicholas W. Simpson, B.Eng.

Carleton University

A thesis submitted to

the Faculty of Graduate Studies and Research

in partial fulfillment of

the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute for

Mechanical and Aerospace Engineering

Department of

Mechanical and Aerospace Engineering

Carleton University

Ottawa, Ontario

August 26, 2004

Your file Votre référence
ISBN: 0-612-97442-1
Our file Notre référence
ISBN: 0-612-97442-1

PAGINATION ERROR.

TEXT COMPLETE.

ERREUR DE PAGINATION.

LE TEXTE EST COMPLET.

Abstract

Calibration is a necessity for those who operate industrial robots and a cost-effective, time-efficient means of accomplishing this task is highly desired. This thesis describes the development of a calibration procedure for six-axis serial robots where the distinctive feature of the procedure is that it employs the novel *Relative Measurement Concept.* The main goal of this procedure is to utilize measurements relative to the end-effector of the robot, as opposed to those provided in an absolute fixed coordinate frame. Images of a precision-ruled straight-edge, acquired along its length and focused on the graduations, yield relative measurement of the positioning errors. Robot parameter deviations can then be identified so that the nominal parameters, resident in the robot controller, can be corrected. The objectives of this thesis are to: develop the robot kinematics necessary to compute the forward and inverse kinematics for any six-axis serial robot, develop a procedure capable of using relative measurements, perform a simulation of the procedure, and validate the results through experimentation with a Thermo CRS A465 serial robot. Simulation results are very encouraging and the experimental results lead to the identification of the multiple sources of error corrupting the experimental calibration measurements.

Acknowledgements

I would like to thank my thesis supervisor, Dr. John Hayes, as he has been an indisputably positive force during this project and instrumental to its success. He has been a great teacher when called upon and an equally great collaborator when problems needed to be solved along the way. For your guidance, support, and healthy sense of humour, I thank you.

Secondly, I would like to thank my partner in this project, Andrew Fratpietro. Andrew is a very dependable and studious colleague, and more importantly a good friend. He has always been willing to listen to ideas, offer his own, and share the workload concerning all aspects of our combined project. I would also like to offer thanks to all of the elite members of graduate student office 3284ME for providing a real sense of community and belonging. Lastly, I would like to thank Mandy Feldman, a woman of great character and beauty, as she has constantly encouraged and supported my work as she has strived and laboured with her own.

Finally, this project would not have been possible without the financial support provided by NSERC, Materials and Manufacturing Ontario, for which Carole Champion was the project liaison, and the Department of Mechanical and Aerospace Engineering of Carleton University.

Dedicated to my family,

for their continual love and encouragement.

List of Symbols

α	rotation about the x axis, Denavit-Hartenberg joint twist parameter
β	rotation about the y axis
Δ	difference
$\Delta\Theta$	stacked vector of parameter deviations
γ	Euler angle
η	angle from a transcendental function
θ	rotation about the z axis, Denavit-Hartenberg joint variable
σ	standard deviation
ϕ	Euler angle
ψ	Euler angle
a	axis, Denavit-Hartenberg link length parameter
$\mathbf{a}$	unit vector
a'	axis
$\mathbf{a}'$	unit vector
a_x	component of $\mathbf{a}$ in the x direction, homogenized component
a_y, a_z	components of $\mathbf{a}$, in the y and z directions
A_i	actual points
A_{p_i}	actual points projected on to the image plane
$\mathbf{A}$	general matrix
b_y	homogenized component
B	origin of the base frame
c	arbitrary constant
c_z	homogenized component

C	calibration index
C_i	controller points
C_{p_i}	controller points projected on to the image plane
C_{u_i}	updated controller points projected on to the image plane
d	Denavit-Hartenberg link offset parameter
$\mathbf{d}$	vector directed at the origin of F_{noa} from the origin of F_{xyz}
d_x, d_y, d_z	components of $\mathbf{d}$ in the x, y, and z directions
f_1, f_2, f_3	simplifying functions of Pieper's method
F_0	base frame
F_A, F_B, F_C	intermediate coordinate frames
F_i	frame i
F_{noa}	coordinate frame consisting of the n, o, and a unit vectors
$F_{noa'}$	coordinate frame consisting of the n', o', and a' unit vectors
F_N	frame N
F_W	wrist centre-point frame
F_{xyz}	coordinate frame consisting of the x, y, and z unit vectors
g_1, g_2, g_3	simplifying functions of Pieper's method
G_i	graduation points
i	index variable
$\mathbf{i}$	unit vector
$\mathbf{I}$	identity matrix
$\mathbf{j}$	unit vector
$\mathbf{J}$	Jacobian matrix
$\mathbf{J_a}$	Jacobian matrix with respect to link lengths
$\mathbf{J_d}$	Jacobian matrix with respect to link offsets
$\mathbf{J}_\alpha$	Jacobian matrix with respect to joint twists

$\mathbf{J}_\theta$	Jacobian matrix with respect to joint variables
k_1, k_2, k_3, k_4	simplifying functions of Pieper's method
$\mathbf{k}$	unit vector
m	number of measurements
n	axis, number of modelled kinematic parameters
n'	axis
$\mathbf{n}$	unit vector
$\mathbf{n}'$	unit vector
n_x, n_y, n_z	components of $\mathbf{n}$ in the x, y, and z directions
N	total number of frames
o	axis
o'	axis
$\mathbf{o}$	unit vector
$\mathbf{o}'$	unit vector
o_x, o_y, o_z	components of $\mathbf{o}$ in the x, y, and z directions
O_i	origin corresponding to F_i
O_{noa}	origin corresponding to F_{noa}
O_{xyz}	origin corresponding to F_{xyz}
$\mathbf{p}$	vector terminating at P
$\mathbf{p}_M$	measured point
${}^{noa}\mathbf{p}$	$\mathbf{p}$ expressed with respect to F_{noa}
${}^{xyz}\mathbf{p}$	$\mathbf{p}$ expressed with respect to F_{xyz}
${}^0\mathbf{p}_4$	position of the wrist centre-point described relative to the base frame
${}^3\mathbf{p}_4$	position of the wrist centre-point described relative to F_3
$\Delta\mathbf{p}_D$	ruler alignment deviation
$\Delta\mathbf{p}_E$	absolute error

$\Delta \mathbf{p}_R$	relative error
p_n, p_o, p_a	components of $\mathbf{p}$ in the n, o, and a directions
p_x, p_y, p_z	components of $\mathbf{p}$ in the x, y, and z directions
P	general point
r	magnitude of a vector, number of variables
$\mathbf{R}$	rotation matrix
${}^{xyz}\mathbf{R}_{noa}$	rotation matrix describing F_{noa} with respect to F_{xyz}
S	origin of the sensor frame
$\mathbf{S}$	diagonal matrix of singular values
t_n, t_o, t_a	translational components in the n, o, and a directions
t_x, t_y, t_z	translational components in the x, y, and z directions
T	transpose
$\mathbf{T}$	transformation matrix
$\mathbf{T}_{Rotation}$	elementary rotation transformation
$\mathbf{T}_{Translation}$	elementary translation transformation
${}^{i}\mathbf{T}_{i+1}$	general transformation matrix that describes F_{i+1} relative to F_i
${}^{noa}\mathbf{T}_{noa'}$	transformation matrix that describes $F_{noa'}$ relative to F_{noa}
${}^{T_F}\mathbf{T}_{T_T}$	transformation matrix that describes T_T relative to T_F
${}^{xyz}\mathbf{T}_{noa}$	transformation matrix that describes F_{noa} relative to F_{xyz}
${}^{0}\mathbf{T}_{T_T}$	transformation matrix that describes T_T relative to the base frame
${}^{0}\mathbf{T}_{W}$	transformation matrix that describes W relative to the base frame
${}^{4}\mathbf{T}_{6}\vert_{\theta_4=0}$	transformation describing the orientation of the of the robot's wrist
T_F	tool flange centre-point
T_T	tool tip centre-point
u	single parameter in a polynomial expression
u_i	single parameter in a polynomial expression corresponding to θ_i

$\mathbf{u}$	arbitrary vector
$\mathbf{U}$	matrix of left singular values
$\mathbf{v}$	arbitrary vector
$\mathbf{V}$	matrix of right singular values
w	homogenizing coordinate
W	wrist centre-point
x	axis
x_A, x_B, x_C	x axis of F_A, F_B, F_C
x_i	x axis of F_i, denominator of $\tan^{-1}$ function, independent variable
$\mathbf{x}$	general vector
y	axis
y_A, y_B, y_C	y axis of F_A, F_B, F_C
y_i	y axis of F_i, numerator of $\tan^{-1}$ function, dependent variable
z	axis, z-component in Pieper's method
z_A, z_B, z_C	z axis of F_A, F_B, F_C
z_i	z axis of F_i

Contents

List of Figures

List of Tables

Claim of Originality

Certain aspects of the kinematic calibration of six-axis serial robots are presented herein for the first time. The following contributions are of particular interest:

1. The implementation of Pieper's solution to the inverse kinematic problem in Matlab.
2. A closed-form solution and implementation in Matlab for the inverse kinematics of wrist-partitioned serial robots using DH parameters.
3. A simulation of a kinematic calibration procedure, that uses absolute measurements, for six-axis serial robots.
4. A simulation of a kinematic calibration procedure, that uses relative measurements, for six-axis serial robots.
5. An automated data acquisition system that is used to capture numerous images of a measurement artifact.
6. An error analysis of the experiment designed to validate the *Relative Measurement Concept.*

Parts of these results have appeared in three refereed publications and technical reports [1, 2, 3].

Chapter 1

Introduction

Errors in end-effector position and orientation, which afflict all forms of robots, are the primary indication of their performance and can dictate their possible applications. Over the past four decades, since their introduction into the industrial world, the performance characteristics of robots have been greatly advanced. However, the need for precisely constructed complex parts has also grown substantially over the same time period. To ensure that robot routines, that are generated through off-line programming, do not suffer needlessly from deviations in robot geometry generated during construction, some form of calibration must be performed. This thesis presents a calibration method that uses relative measurements for robot kinematic parameter deviation identification. Computer simulations and experimental results were pursued to develop and validate the *Relative Measurement Concept* (RMC). The requirements for the new calibration system, which employs the RMC are that it must perform as well as those based on absolute measurements, successfully estimate all observable parameters, be of minimal additional cost to the user, and not require the removal of the robot from its workspace.

1.1 Motivation

Serial robots can be represented as a kinematic chain of links and joints. In between two successive links is a joint that provides some form of relative motion capability. If the first joint is commanded to a specific value, all the subsequent links and joints of the chain are affected. The Thermo CRS A465, illustrated in Figure 1.1 was the robot acquired for the experimental research of this thesis. It is a six degree-of-freedom serial robot. The robot is rigidly mounted to a surface through its base and different tools can be mounted to the robot's tool flange. The end-effector, rigidly attached to the tool flange, is the last link in the kinematic chain and represents the means by which the robot can affect its environment. This particular robot has six revolute joints. The three distal axes all intersect at a point and this configuration is commonly referred to as *wrist-partitioned.*

To achieve some desired pose, which places the end-effector of the robot in a specific position with a specific orientation, each controllable joint variable must be commanded

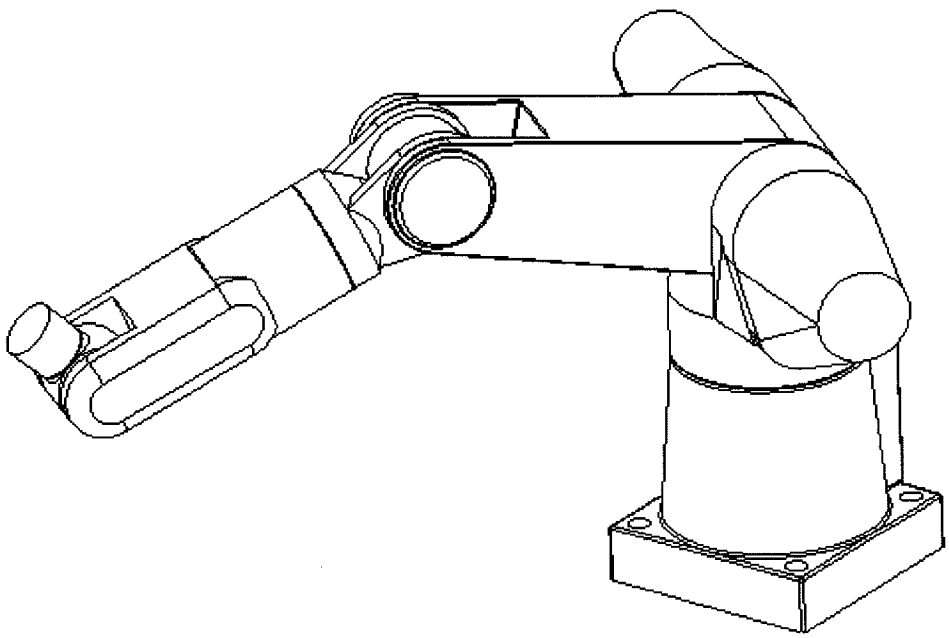

Figure 1.1: The Thermo CRS A465.

to the proper value. Key information concerning the geometry of the robot, and various ways to use this information, must be stored in the robot controller. The robot controller is a computer system that accompanies and interfaces with the robot. A geometric model, which fully describes all the links and interconnections of the robot, is stored in its memory along with the means to compute how each joint must move to execute a motion.

To predict the end-effector position and orientation, given the joint angles, the forward kinematics of the robot are computed. To go to a specified position and orientation, the inverse kinematics are computed to obtain the necessary joint angles. Finally, to move the end-effector at a specified speed the robot Jacobian must be computed and continuously updated. The Jacobian relates the linear and angular velocity of the end-effector to the joint rates. All of these hinge on an accurate kinematic model of the robot. Inaccuracies in this model result in end-effector positioning and orientation errors as well as deviations in trajectory and path planning. Thus, the kinematic model stored in the controller is of great importance to the robot's potential as a useful machine.

For a serial robot, any deviation between the nominal geometry of the robot and the actual geometry defined in its manufacture, will propagate that error to the end-effector. Geometric errors, which are the primary source of endpoint positioning error, accounting for up to 95 % [4], can be identified through kinematic calibration. A calibration technique for identifying these errors is the subject of this thesis. In general, a calibration procedure entails acquiring measurements of the end-effector, comparing these measurements of the robot's pose to the predicted position and orientation, and relating these errors to the kinematic parameter deviations.

1.2 Objectives

The first objective of this thesis was to design and develop a calibration procedure, capable of identifying the deviations from the nominal robot geometry based on the *Relative Measurement Concept* (RMC), which is discussed in Chapter 3. This calibration procedure was developed by creating a simulation, where parameter deviations were specified at the beginning of the program and the goal was to successfully identify these deviations [1, 3, 5]. A number of simulation components were needed, such as the capability to compute the forward kinematics, inverse kinematics, and the Identification Jacobian matrix. The Denavit-Hartenberg (DH) and the Modified Denavit-Hartenberg (MDH) parameterizations were used to model the Thermo CRS A465 and the KUKA KR 15/2 six-axis serial robots. The capability to substitute the geometry for other similar wrist-partitioned serial robots is also possible. The *Singular Value Decomposition* (SVD) was employed to obtain the pseudo-inverse of the Identification Jacobian matrix, so an approximate solution for the parameter deviations could be computed in a least-squares sense. To ensure the proper function of the program components and to assist in the development of the RMC calibration procedure, a companion simulation, based on absolute measurements, was also created.

The second objective of this thesis was to validate the procedure with an appropriate experimental setup. A calibration system, constructed in conjunction with a concurrent project, consisting of a *Charge-coupled Device* (CCD) camera and a precision-ruled straight edge, was utilized in this attempt at validation. A set of measurements from a preliminary experiment with the KR 15/2 was first analyzed. Multiple sets of measurements were then acquired with the A465.

1.3 Advances in the Calibration of Serial Robots

Historically, calibration has been an expensive and labour-intensive task. Efforts over the past twenty years have been focused on reducing costs and automating the calibration process. The calibration of serial robots, due to their prominence in the manufacturing sector and widespread use, has been a well-documented technical problem with many researchers devoted to its solution. A survey of the more recent advances in serial robot calibration was performed to explore different concepts and determine the current trends of the research sector [2]. The focus of the survey was the proposed techniques for improving calibration in the past ten years. The findings are summarized in what follows.

1.3.1 Calibration Methods

A wide variety of new calibration methods have been developed in the past decade. These include inverse calibration methods and kinematic calibration methods. With inverse calibration, the goal is to estimate the end-effector error for the entire workspace of the robot by taking a discrete number of measurements. This error can then be added to the end-effector position, as determined by the nominal parameters, to place it at the desired pose. The objective of kinematic calibration methods is to identify the deviations in the nominal kinematic parameters of the geometric model of the robot. These methods generally fall into three categories: endpoint constraints, laser-based methods, and more recently, camera-based methods.

Inverse Calibration

With inverse calibration, no attempt is made at identifying the geometric model of the robot, rather a model for the end-effector error is generated. Correction functions are generated by taking a discrete number of measurements in a portion of the workspace.

The functions are then applied over a larger portion of the workspace to predict the end-effector errors. Doria *et al.* [6] proposed a method in which spline functions, which are piecewise polynomials, could predict end-point error based on a given joint angle set and the nominal robot parameters. The end-effector, an attached sphere, was measured by a fixture with three perpendicular sliders. The robot was mounted onto a solid plate with this fixture. The Cartesian error would then be used to adjust the position determined with the nominal parameters.

Zupancic [7] obtained measurements by use of a plate attachment, mounted to the tool flange, with specifically-placed grooves. A stationary measurement device, consisting of two linear displacement transducers, were spaced such that they would measure the top surface and the bottom surface of the groove. This yielded measurements in two directions. Error distribution functions were then generated in accordance with inverse calibration.

Park *et al.* [8] applied an inverse kinematic procedure to a robot operation that only required two poses. By measuring these two poses with an apparatus that consisted of dial gauges and a tooling ball, the Cartesian error was eliminated by appropriately moving the end-effector.

The results of inverse calibration are relatively easy to implement with robot controller software. As the tool frame can be defined by the user, and the predicted errors could be incorporated as a look-up table or a mathematical function, this frame can be re-defined for each specific pose. However, the displacements representing these predicted errors are still afflicted by the parameter deviations. Thus, it is a valid method and would reduce inaccuracy in the robot poses, but to achieve the best results the kinematic parameters deviations should be identified.

Endpoint Constraints

Constraining the endpoint usually entails affixing the end-effector to an external structure. This constraint forms a closed-loop kinematic structure. As long as sufficient mobility remains, the joints can still be actuated. Thus, computation of the forward kinematics for different joint angle sets achieve the same pose of the end-effector. Another way to constrain the end-effector would be to mate it with a planar surface. Relative motion with respect to the plane would still be possible, but for each pose of the robot, geometric constraints are placed on the entries of the general transformation matrix for the robot.

An external sensing device, with several sensed degrees-of-freedom, was developed by Khoshzaban *et al.* [9] to calibrate heavy machines with hydraulically-powered serial manipulators. The device was fixed to the base of the machine and then to the end-effector of the manipulator, closing the kinematic loop. For smaller industrial robots, Omodei *et al.* [10] constructed a measurement artifact in which the end-effector could be inserted into numerous holes.

Tang and Liu [4] compared three methods based on flat surfaces. With a flat surface aligned with the base coordinate system of the robot, a suitable pose, where the end-effector was made to have perfect surface contact with the flat surface, could be achieved. The three methods involved different apparatus. With the first, a solid block with a flat surface was mounted on the tool flange of the robot. For the second, a set of linear displacement transducers, symmetrically arranged, were mounted on the end-effector and the third involved a laser displacement sensor and a stepped pyramid measurement artifact. In Ikits and Hollerbach [11], a touch probe was used in conjunction with a fixed plane described by the general equation of a plane in Cartesian space. Khalil *et al.* [12] also evaluated this method through simulation.

Custom-built sensing devices and fixtures are an innovative way in which to potentially provide a great number of measurements from one contact position. They also make use

of the built-in joint encoders of the robot. The device proposed by Khoshzaban *et al.* [9] allows a certain freedom in the movement of the end-effector, due to the degrees-of-freedom of the sensing device, but the measurement phase is inherently a manual process. The sensing device is subject to kinematic errors and the resolution of the joint encoders limit its potential for other robots, but fine accuracy is not necessary for heavy machines such as excavators and log-loaders. The measurement artifact of Omodei *et al.* [10] requires that each pose be taught to the robot controller. The planar constraint methods suffer from this disadvantage as well, making them somewhat tedious to set up. Once the teaching process is completed, the process could be automated. However, if the surface or measurement artifact is moved, all of the poses must be re-taught. A method that requires very limited setup and space is highly desired.

Theodolites and Laser-based Methods

Theodolites have been widely-used for the calibration of serial robots. They essentially provide the horizontal and elevation angles of the line of sight from the base to a mounted target. They do not provide distance information. However, two or three of these devices can be used to triangulate a position. As multiple targets can be mounted to the end-effector, the full pose of the robot could be measured by mounting three targets.

In their development of an automatic theodolite, Driels and Pathre [13] incorporated a camera into the design of a theodolite. But the automatic theodolite was an additional 2 degree-of-freedom sensing device that needed calibration itself. As with theodolites, this device reported the horizontal and elevation angles with respect to its own coordinate system. However, it was automated with stepper motors that would focus the camera onto the reference target mounted on the robot.

Sultan and Wager [14] used two theodolites to separately determine the location and orientation of a robot's joint axes. Abderrahim and Whittaker [15] performed a similar

calibration technique, termed circle-point analysis, but with a laser tracking device. Both of these methods required the actuation of one joint at a time. The path of the end-effector was a circle in Cartesian space that could be measured by their respective systems with a target mounted on the end-effector. However, with the method of Sultan and Wager [14], the joint transmission errors could be profiled.

Omodei *et al.* [16] calibrated a SCARA class robot with a unique laser and mirror measurement system. A plane mirror and a concave mirror were attached to the end-effector and it was commanded to traverse a path parallel to the direction of a laser beam. The beam was reflected onto a screen which was monitored. A deviation in the incident location yielded a measurement of the orientation of the end-effector.

A 3D laser tracking system and three cat's eye laser targets were used by Drouet *et al.* [17] to calibrate the geometric and elastic errors of a patient positioning system. Due to the wide range of loads expected in its service, the elastic errors were more prominent. However, the calibration was performed as with industrial manipulators but more measurements were required to accommodate the elastic error model.

The use of theodolites is inherently a manual operation, however, methods based on this measurement device are well-established in the literature. The automation of these devices could be quite beneficial. But, their accuracy would depend on the quality of their components, and again the geometry stored in the controller would be different from the machine's actual geometry. Lasers are a similar device, but can also provide the position vector to the target. Mirrors, or metallic surfaces, can be used to reflect the beam as performed by Omodei *et al.* [16]. In their setup, a camera recorded the position of the spot on a screen. Although an interesting setup, the robot does not necessarily follow a path parallel to the laser beam, which could affect the results. Cat's eye laser targets reflect the beam back to the source to register the measurement. Laser tracking systems are based on this principle, only they are automated. Unfortunately, the cost of such a

system would be relatively high.

Camera-based Methods

Camera-based calibration methods rely on *Charge-coupled Device* (CCD) cameras that generate a pixelated image of the reference object. One, or two cameras in a stereo-vision setup, can be incorporated into the calibration system. Analysis of the images reveals the error of the pose. If a camera is mounted as the end-effector of the robot, the extrinsic properties define its relation to the tool flange. Intrinsic properties govern its focal length and the scale factor. Other considerations when using CCD cameras are, for example, perspective distortion, radial distortion, and pin-cushioning.

Zhuang *et al.* [18] developed a one-stage procedure for the calibration of a camera and a robot. As with many other researchers, a PUMA 560 robot was used along with a calibration board. The calibration board is a glass plate with an array of dots painted on the surface. The board was mounted on a *Coordinate Measurement Machine* (CMM), so that it could be displaced in known quantities. This established the world coordinates of the dots. Images of the calibration board are taken from various robot configurations and the transformation between the camera frame and the world frame can be identified. Zhuang *et al.* [19], used this equipment on a SCARA arm in a similar calibration procedure. Given the world coordinates of the calibration board and acquiring its image coordinates with the CCD camera, the intrinsic and extrinsic properties of the camera are identified. Essentially, these properties determine the pose of the camera, the end-effector, in the world coordinates.

A single CCD camera was used by Motta *et al.* [20] to calibrate two robots: an ABB IRB-2400 and a PUMA 500. The camera was mounted to the tool flange and acquired images of a calibration board placed at a distance of 0.60 m to 1.00 m from the robot. The camera parameters were obtained through use of the radial alignment constraint method.

Rousseau *et al.* [21] used a unique measurement artifact in their camera-based calibration method. The target consisted of a number of spheres that were accurately positioned in three-dimensional space. When an image of the sphere array was taken, the pose of the camera could be determined as the spheres would be in unique pose. One of the spheres was oversized to ensure that all the spheres could be identified. The coordinates of the circles in the image plane were mapped to the physical coordinates of the spheres to identify the camera pose.

Meng and Zhuang [22] developed a self-calibration method for camera-robot systems. Self-calibration is a process in which the parameters of the system are determined without a known scale factor. The rotational aspect of the camera-tool flange transformation can be successfully identified but the translational component is only known up to a scale factor. A set of optimal configurations are chosen that have the same scale factor and then a length standard is incorporated.

1.3.2 Calibration Optimization

Many new and innovative methods to perform calibration were developed in the last decade. However, some researchers have developed ways in which to improve the calibration process. These include creating new models to describe robot geometry, simulation techniques to identify sufficient measurement poses and suitable equipment, indexing of the calibration methods to classify them according to measurement potential, and the optimization of robot configuration for measurement purposes.

Zhuang *et al.* [23], Okamura and Park [24], and Chen *et al.* [25] proposed new models in which the geometry of a robot could be described. It is desired that geometric models be minimal, complete, and parameterically continuous. A model is minimal and complete when the minimum number of parameters is used to fully describe the geometry of the robot. It is parameterically continuous when small deviations in robot geometry result

in small errors of the end-effector. The DH parameterization suffers from singularities when neighbouring axes are nearly-parallel. Zhuang *et al.* [23] modified their previously developed complete and parameterically continuous model to bring it closer to the widely-used DH model. Okamura and Park [24] and Chen *et al.* [25] utilize the product-of-exponentials model, which is based on screw theory.

Adopted by most of the research community, Zak *et al.* [26] emphasized the need for simulation of calibration experiments before an actual attempt was made. Hollerbach and Wampler [27] created a calibration index so that a particular calibration method could be judged based on the measurements that are performed. Essentially, all calibration methods can be thought of as closed-loop kinematic chains, where the measurement closes the loop. The generation of optimal robot configurations for calibration was researched by Zhuang *et al.* [28]. By using an observability index, the condition number of the Jacobian matrix, which relates linear and angular velocities of the end-effector to the joint rates, the effect of the measurement uncertainties could be minimized.

1.3.3 Distance Measure Method and the RMC

A calibration method, devised by Gong *et al.* [29] and compared to other methods by Khalil *et al.* [12], uses a measurement device that is a combination of a laser displacement sensor and an optical sensor. It is based on configuring the robot in a pose that places the end-effector at a known distance relative to another pose. This is similar to the RMC, but there are significant differences.

In their calibration method, a measurement artifact was constructed where a number of holes were drilled in an aluminum plate. The centre-to-centre distance between each hole was measured by a CMM. A CMM typically has an accuracy of ± 0.01 mm. The optical sensor, mounted on the tool flange of the robot and requiring hand-to-sensor calibration, placed the hole's centre such that it was coincident with the image centre in the image

plane. The vertical distance from the end-effector to the plate was provided by the laser displacement sensor. This sort of positioning was most likely accomplished by teaching. Though it was stated that the measurement process was automated, it does not exploit the fact that the robot moves relative to its current position. By precisely positioning the robot at each of the holes, where the distance between all the other holes is known, the error in this length can be computed. This method has an index value of $C = 1$ according to Hollerbach and Wampler [27]. This index essentially means that for one measurement with their system, one equation is generated. For the system based on the RMC, $C = 2$, which will be expanded to $C = 3$, and possibly to $C = 6$, which corresponds to full pose measurement and the maximum value of the index. This means that a factor of six fewer measurements would be needed.

With the RMC, the relative nature of the movements, prone to error derived from the inaccuracy of the robot model and not its repeatability, is utilized. With an attached CCD camera, which does not require hand-to-sensor calibration, the end-effector is moved in increments along the length of a precision-ruled straight edge. By not having to perform hand-to-sensor calibration, the error chain is shorter and errors generated in the camera calibration would not propagate to the robot calibration. The ruler is commercially available and fabricated according to certain standards governing flatness and tolerances. It has an accuracy of approximately ± 0.003 mm. The cost of this ruler is approximately \$560.00, compared to \$50,000.00 for a CMM. Images are taken at each of the increments. The shifts in the images, when compared to a reference image, yields the difference data. An equivalence to the method offered by Gong *et al.* [29] would be teaching all the robot poses so the camera was centred on the intersection of the edge of the ruler and each graduation of the ruler. This is a highly arduous task given the repeatability of the robot.

The measurements acquired using the RMC are not reduced to a single dimension as in the distance measurement method. Errors in two coordinate directions are currently

measured, hopefully to be expanded to a third with a laser displacement sensor. Our goal is to measure relative displacement errors in all three coordinate directions and substitute them for absolute measurements, not reduce them to one dimension.

1.4 Thesis Overview

In this chapter, the problem of geometric errors, that afflict all manner of robots, was introduced. The source of these errors were discussed along with the need for the calibration of robots. Advances and innovations in the calibration of serial robots, and matters relating to calibration, over the last decade, were reviewed and it was found that most rely on absolute measurement positioning sensors or camera-based techniques. Camera-based methods, like the one incorporating the RMC, were found to be the most cost-effective and promising.

Presented in Chapter 2 is the theoretical background needed for the geometric modelling of robots, the computation of the forward and inverse kinematics, and the derivation of the Identification Jacobian matrix. The fundamentals of robot kinematics, leading to homogeneous transformation matrices, are discussed, followed by the description of the DH and MDH parameters, which includes the assignment of coordinate frames and the robot parameters. Brief descriptions of the Thermo CRS A465 and the KUKA KR 15/2 are given along with their associated DH and MDH parameters and coordinate frame assignments. Two methods for the computation of the inverse kinematics for six degree-of-freedom serial robots are derived: Pieper's solution and a closed-form solution that uses the DH parameters. Finally, the derivation of the Identification Jacobian is reported along with its pertinence to calibration.

Chapter 3 serves to introduce the RMC and describes a conventional calibration method, based on absolute measurements, used as a reference model. The fundamentals of robot

kinematics, explored in Chapter 2, are used in the development of both of these methods. The SVD matrix factorization method is presented as it is used to compute the pseudo-inverse of the Identification Jacobian matrix in both methods. The method in which the RMC measurements are utilized in a calibration scheme is discussed as well as the benefits of the experimental system.

Chapter 4 presents the simulation results for the absolute and relative cases. The implementation of the essential components of the simulations, pose generation, measurement acquisition, the incorporation of noise, parameter identification, and convergence, are described. Results from both cases are offered along with particular issues that would influence the experimental trials.

In Chapter 5, the setups for both experiments, based on the KUKA KR 15/2 and the Thermo CRS A465, are described. In the case of the A465, an automated camera-based measurement acquisition system was developed. Several observations are made concerning the experimental data and the influences of measurement noise. The results of the kinematic calibration procedure, with this data, are reported.

Chapter 6 summarizes the achievements made in this thesis. Several conclusions are offered from the results of the simulations and preliminary experiment towards a kinematic calibration procedure based on relative measurements. Finally, recommendations for future work and experimental testing are suggested.

Chapter 2

Robot Kinematics

The fundamental principles of robot kinematics, which allow one to represent a physical robotic manipulator as a geometric model, will be discussed in this chapter. The simulations created to perform the kinematic calibration, which utilize absolute and relative measurements, employ these principles to compute the position and orientation of the end-effector (forward kinematics), the corresponding joint angles necessary to attain a specific pose (inverse kinematics), and the Identification Jacobian matrix used for kinematic parameter deviation identification (differential kinematics).

Robots are an arrangement of links connected together with joints which allow relative motion between the links. Joints can either be rotational, where the relative motion consists of a rotation about an axis defined at the joint, or prismatic, where the relative motion consists of a translation parallel to the longitudinal axis of the joint, or combinations involving these basic joints. Spherical, cylindrical, screw, and planar joints can be modelled as combinations of prismatic and revolute joints. A series of rigid links connected together via joints is modelled as a kinematic chain. Within this chapter, the mathematical concepts required to geometrically model robots as kinematic chains will be presented. This includes the computation of the forward kinematics, inverse kinematics

and the Identification Jacobian.

2.1 Forward Kinematics

The computation of the forward, or direct, kinematics for a robot is the process in which the position and orientation of the end-effector, the last link of the kinematic chain, is calculated given the variable joint inputs and geometric model. However, the representation of points, vectors, and transformations using matrix operations must first be discussed as well as the parameterization conventions.

2.1.1 Transformations

A transformation, or mapping, is a matrix operation that transforms the coordinates of a point in one frame into coordinates for the same point expressed in another frame. Attached to each joint of a robot, along with its base and end-effector, is a coordinate frame and so it is of key importance to be able to represent points in these frames and express them relative to any desired frame. Matrices and vectors will be used to describe points, orientations, translations, rotations, and general transformations.

A point, P, as seen in Figure 2.1, has the coordinates p_x, p_y, and p_z relative to reference

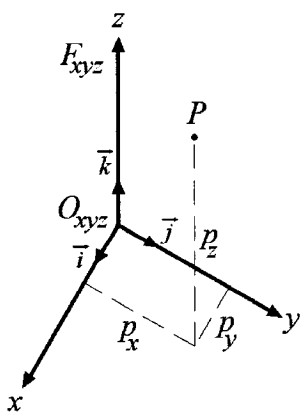

Figure 2.1: Representation of a point in a reference frame.

frame F_{xyz}. A reference frame, for the purposes of this thesis, is a stationary coordinate system in which all other constructs, such as points, vectors, or frames themselves are described. F_{xyz} is defined by origin O_{xyz} and three orthogonal unit vectors: **i**, **j**, and **k**, which are directed along the x, y, and z axes. Thus, the orthogonal component-wise expression for point P is:

$$P = p_x\mathbf{i} + p_y\mathbf{j} + p_z\mathbf{k}. \tag{2.1}$$

This expression of point P can also be represented using vector form:

$$\mathbf{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}. \tag{2.2}$$

In this form, the entries of the rows represent coordinates in the three linearly independent orthonormal basis directions.

A vector is used to describe the distance between two points in space as well as a specific direction. The point P and the origin of reference frame F_{xyz} can be used in the assignment of a position vector directed towards and terminated at point P while originating from O_{xyz}, as seen in Figure 2.2. The coordinates of O_{xyz} are $(0, 0, 0)^T$. Thus,

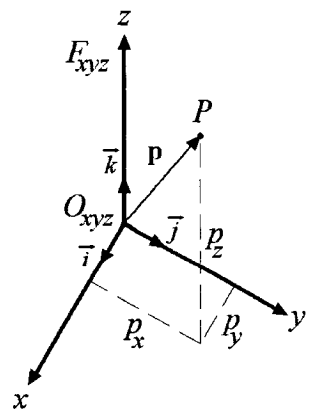

Figure 2.2: Representation of a vector in a reference frame.

the expression for the position vector $\mathbf{p}$ is:

$$\begin{aligned} \mathbf{p} &= (p_x - 0)\mathbf{i} + (p_y - 0)\mathbf{j} + (p_z - 0)\mathbf{k} \\ &= p_x\mathbf{i} + p_y\mathbf{j} + p_z\mathbf{k}. \end{aligned} \tag{2.3}$$

A vector with a left-hand superscript, such as ${}^{xyz}\mathbf{p}$, means the vector is described in that specific coordinate frame. Thus, the vector $\mathbf{p}$ of Equation 2.3, could have been written ${}^{xyz}\mathbf{p}$. Note that the expression obtained in Equation 2.3 is identical to that in Equation 2.1. The vector form of this expression is identical to that of Equation 2.2.

The means to use homogeneous transforms will now be discussed. A 3×3 rotation matrix,

$$\mathbf{R} = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}, \tag{2.4}$$

can be created by concatenating three unit vectors, $\mathbf{n}$, $\mathbf{o}$, and $\mathbf{a}$. With respect to Figure 2.3, the n direction of F_{noa} is expressed in the x, y, and z coordinate directions of the reference frame, F_{xyz}, in the first column of Equation 2.4. Thus, to express points defined in F_{noa}

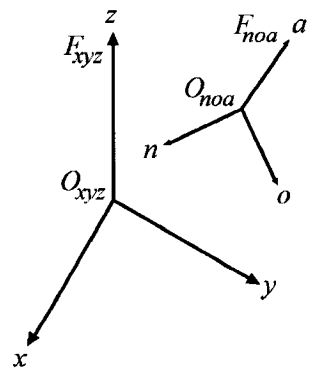

Figure 2.3: Representation of a coordinate frame relative to a reference frame.

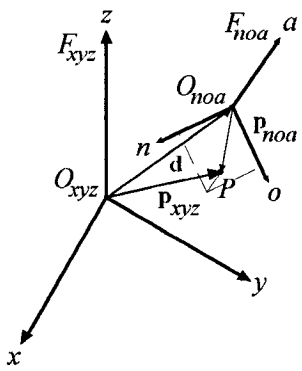

Figure 2.4: Point P expressed in frames F_{xyz} and F_{noa}.

in the coordinates of F_{xyz}:

$$^{xyz}\mathbf{p} = {}^{xyz}\mathbf{R}_{noa}\,{}^{noa}\mathbf{p} + \mathbf{d}. \tag{2.5}$$

The coordinates of vector $^{xyz}\mathbf{p}$, expressed with respect to F_{xyz}, are computed by first aligning F_{noa} with F_{xyz} through the rotation $^{xyz}\mathbf{R}_{noa}$, applying this to $^{noa}\mathbf{p}$, and then translating them by the distance between the origins of the frames, $\mathbf{d}$. This can be seen in Figure 2.4.

However, this operation, related in Equation 2.5, is not a linear transformation. The operation is not distributive under the two properties of vector addition, $\mathbf{T}(\mathbf{u}+\mathbf{v}) = \mathbf{T}(\mathbf{u})+\mathbf{T}(\mathbf{v})$, or scalar multiplication, $\mathbf{T}(c\mathbf{u}) = c\mathbf{T}(\mathbf{u})$, where $\mathbf{T}$ represents a transformation matrix [30]. A convenient matrix multiplication approach, that combines rotations and translations into one operation, is desired. This is achieved through the use of homogeneous coordinates. An additional row is included in the description of positions:

$$\mathbf{p} = \begin{pmatrix} a_x \\ b_y \\ c_z \\ w \end{pmatrix}, \tag{2.6}$$

where p_x, p_y, and p_z are related to a_x, b_y, and c_z by:

$$p_x = \frac{a_x}{w}, \quad p_y = \frac{b_y}{w}, \quad p_z = \frac{c_z}{w}. \tag{2.7}$$

The homogenizing coordinate, w, is used to define directional and positional vectors. A vector which has a homogenizing coordinate equal to one yields the Cartesian difference in position between two points. A vector whose homogenizing coordinate is equal to zero is used to indicate a direction with respect to the reference frame. When $w = 0$, the vector represents the point at infinity through which all parallel vectors pass.

The means to algebraically represent coordinate frames, using homogeneous coordinates, has now been established. With respect to a reference frame, another coordinate frame can be described by amalgamating four vectors, three directional vectors and one positional, into one matrix. The orientation and position of this new frame, the working frame, relative to the reference frame is determined by the entries within this matrix. In general, to transform points described in F_{noa} to their counterparts in F_{xyz}, shown in Figure 2.3, the matrix:

$$^{xyz}\mathbf{T}_{noa} = \begin{bmatrix} n_x & o_x & a_x & d_x \\ n_y & o_y & a_y & d_y \\ n_z & o_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.8}$$

is used, where the first three columns consist of entries that define the directions of the three mutually orthogonal unit basis vectors of F_{noa} relative to the reference frame F_{xyz} and the entries of the last column define the position of the origin of F_{noa}, the working

frame, with respect to F_{xyz}. Thus, the relation replacing Equation 2.5 is:

$$\left\{\begin{array}{c} p_x \\ p_y \\ p_z \\ \hline 1 \end{array}\right\} = \left[\begin{array}{ccc|c} & \mathbf{R}_{3\times3} & & \mathbf{d}_{3\times1} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \left\{\begin{array}{c} p_n \\ p_o \\ p_a \\ \hline 1 \end{array}\right\}. \tag{2.9}$$

where the structure of the transformation matrix is illustrated.

Unless otherwise stated, the transformations used in this thesis are coordinate transformations, as opposed to geometric transformations. A coordinate transformation can be thought of as moving the coordinate system relative to a stationary object while a geometric transformation is used to displace the object relative to a stationary coordinate system. A geometric transformation is the inverse of the corresponding coordinate transformation. The transformation operator is denoted $\mathbf{T}$, where the left superscript indicates a reference frame and the right subscript indicates the working frame described relative to that reference frame. With reference to Figure 2.4, the vector $\mathbf{p}_{noa}$ is transformed into vector $\mathbf{p}_{xyz}$ via the transformation ${}^{xyz}\mathbf{T}_{noa}$ and computed by:

$$\mathbf{p}_{xyz} = {}^{xyz}\mathbf{T}_{noa}\ \mathbf{p}_{noa}. \tag{2.10}$$

If F_{noa} originated at the same point as the reference frame and its coordinate directions were the same as well, then the matrix would be a 4×4 identity matrix,

$${}^{xyz}\mathbf{T}_{noa} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.11}$$

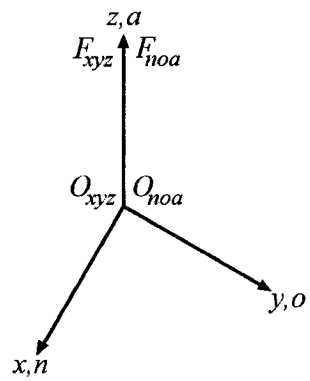

Figure 2.5: Representation of a coordinate frame that is coincident relative to a reference frame.

A visual representation of this can be seen in Figure 2.5. This can be thought of as a starting point in the construction of a complex transformation from a series of elementary transformations.

There are two elemental displacements: translations and rotations. A translation is a linear displacement in a specified direction and a rotation is an angular displacement about a specified axis. With translations, the origin of a frame is displaced by a fixed amount while the orientation of the frame remains constant. Consider the three frames, F_{xyz}, F_{noa}, and $F_{noa'}$ of Figure 2.6. The translation between F_{noa} and $F_{noa'}$ can be described with a

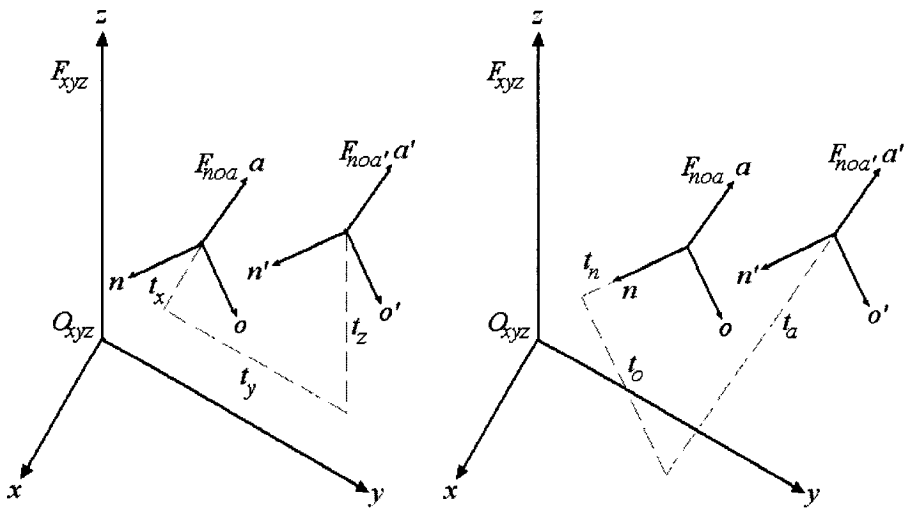

Figure 2.6: The two interpretations of a translation of frame F_{noa}.

transformation matrix. F_{noa} has already been described relative to F_{xyz} with ${}^{xyz}\mathbf{T}_{noa}$. The translation to $F_{noa'}$ can be given relative to F_{xyz} or with respect to F_{noa}. Interpretations of both of these cases can be seen in Figure 2.6.

The effect of specifying the translation relative to the reference or working frame is to either *pre-* or *post*-multiply the transformation in the overall series of transformations. With reference to Equation 2.10, the translational transform would appear immediately before or after ${}^{xyz}\mathbf{T}_{noa}$. Pre-multiplication is for transformations defined relative to the reference frame while post-multiplication is reserved for transformations defined with respect to working frame. The translation transformation matrix, with respect to the F_{xyz} reference frame, is:

$$\mathbf{T}_{Translation}(t_x, t_y, t_z) = {}^{noa}\mathbf{T}_{noa'} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.12}$$

The operation will only affect the positional elements, p_x, p_y, and p_z of the general transformation matrix of Equation 2.8 when multiplied. The rotation portion is unaffected by such a transformation.

For rotational transformations, an angular displacement is specified about a particular axis. The right-hand-rule convention is used for positive angular displacements. The minimum impact of a rotation is that it will change the direction of two axes of the working frame. This occurs when the angle is specified about one of the axes of the working frame. If the angle is specified about an axis of the reference frame, where the working frame is arbitrarily oriented relative to the reference frame, then all three directions for the axes of the working frame will change. To identify the rotation transformation matrix, consider the two frames, F_{xyz} and F_{noa}, which originate at the same point but F_{noa} has been rotated

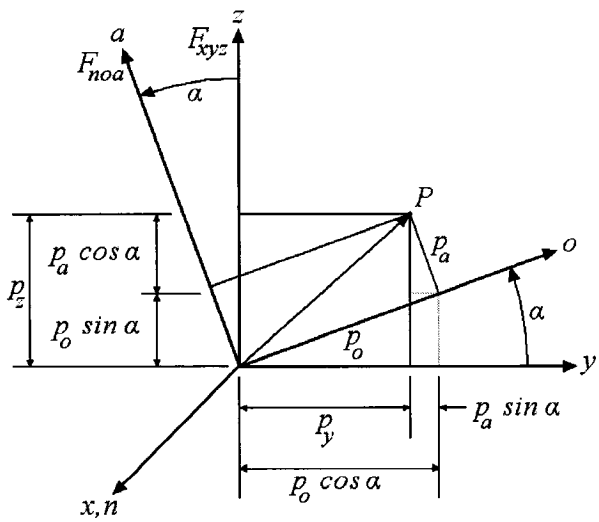

Figure 2.7: Two frames with the same origin, F_{xyz} and F_{noa}, where frame F_{noa} has been rotated by an angle α about the x axis.

by an angle, α, about the x axis, or n axis, as depicted in Figure 2.7. A point, P, exists in the yz plane, as well as the oa plane, and is described relative to both frames with the appropriate distances. The expressions that relate the position of the point in the reference frame to those in the current frame are:

$$p_x = p_n, \tag{2.13}$$

$$p_y = p_o \cos\alpha - p_a \sin\alpha, \tag{2.14}$$

$$p_z = p_o \sin\alpha + p_a \cos\alpha. \tag{2.15}$$

By arranging the expressions of Equations 2.13 to 2.15 into matrix form and including the homogenizing coordinates,

$$\begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} p_n \\ p_o \\ p_a \\ 1 \end{pmatrix}. \tag{2.16}$$

Recall that the two frames share the same origin, thus the translational portion is $(0, 0, 0, 1)^T$.

The transformation matrices for rotations about the x, y, and z axes by angles of α, β, and θ, respectively, can be computed with:

$$\mathbf{T}_{Rotation}(x,\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.17}$$

$$\mathbf{T}_{Rotation}(y,\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.18}$$

$$\mathbf{T}_{Rotation}(z,\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.19}$$

The elementary displacements are depicted in Figure 2.8.

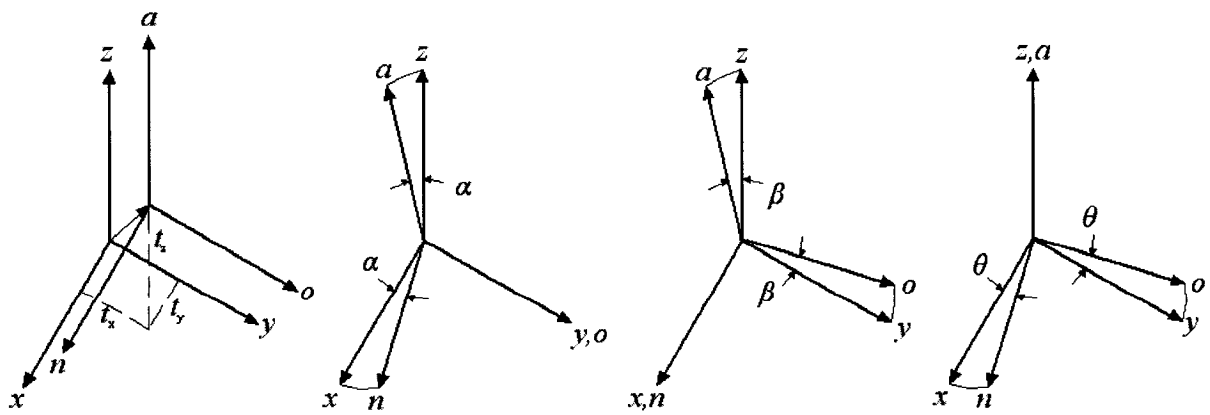

Figure 2.8: The elementary transformations.

The elementary transformations presented in this chapter are used to construct more complex transformations based on various conventions for the geometric modelling of robots. The translations and rotations used to describe robots configurations, from the base, to the joints, and finally to the end-effector, are dependent upon the geometry of the robot and the joint angles provided for a specific pose. These conventions will be explored next.

2.1.2 Denavit-Hartenberg Parameters

The standard model used to algebraically represent the kinematic geometry of a robot was proposed by Denavit and Hartenberg [31]. Under their convention, a coordinate frame is attached to each joint of the robot, from which coordinate transformation matrices are derived, which are then used to compute the kinematics and dynamics of the system. Thus, there are two main tasks; assigning the coordinate frames and then characterizing the associated robot parameters used in the matrix operations.

The general procedure in which coordinate frames are assigned is as follows:

(i) Identify all the joint axes of the robot. All joints are represented by a z axis. For rotary joints, it indicates the axis of rotation. For prismatic joints, it indicates the direction of translation.

(ii) Identify the common perpendiculars between neighbouring joint axes. The origin of the i^{th} frame lies on the $(i+1)^{th}$ axis at its intersection with the common normal between the i^{th} and $(i+1)^{th}$ axes. The x_i axis is directed along this common normal. In general, there are three cases that can arise in terms of its direction.

A. The neighbouring joint axes are skew lines. In this case, a unique common normal exists between the axes and the direction of this line is the direction of the x axis for the current coordinate frame.

B. The neighbouring joint axes intersect. In this case, the x axis will be assigned in the direction perpendicular to the plane that contains the two z axes of the neighbouring joint axes.

C. The neighbouring joint axes are parallel. In this case, there are an infinite number of common normals. The common normal co-planar with the previous x axis will be chosen to represent the x_i axis.

(iii) In general, assign the z_i axis to be directed along the joint axis $i+1$.

(iv) Assign the x_i axis to be the relevant direction specified in the three cases of (ii).

(v) Assign the y_i axis to complete the right-hand rule.

Note that the assignment of the base reference frame, F_0, is not included in these steps. Any convenient location will suffice, and this is usually chosen to be somewhere in the base of the robot and defines the world coordinate system. This means that the frame assignments are not unique. Moreover, the positive directions of rotations and translations associated with joint variables are arbitrary; however, convenient choices should be made. For rotations, the positive direction is taken as that defined by the right-hand-rule for axes and for translations, the positive direction is along the length of the actuator towards its distal end.

Once the coordinate frames have been assigned, the Denavit-Hartenberg (DH) parameters must be extracted. Each of these parameters may be thought of corresponding to an elementary transformation. The orientation and position of each frame, relative to the previous one, can be obtained through the multiplication of the elementary transformation matrices. A set of four DH parameters exists for each joint of the robot. The four parameters comprise: a controlled joint variable, either an angular displacement in the case of revolute joints or a linear displacement in the case of prismatic joints, and three

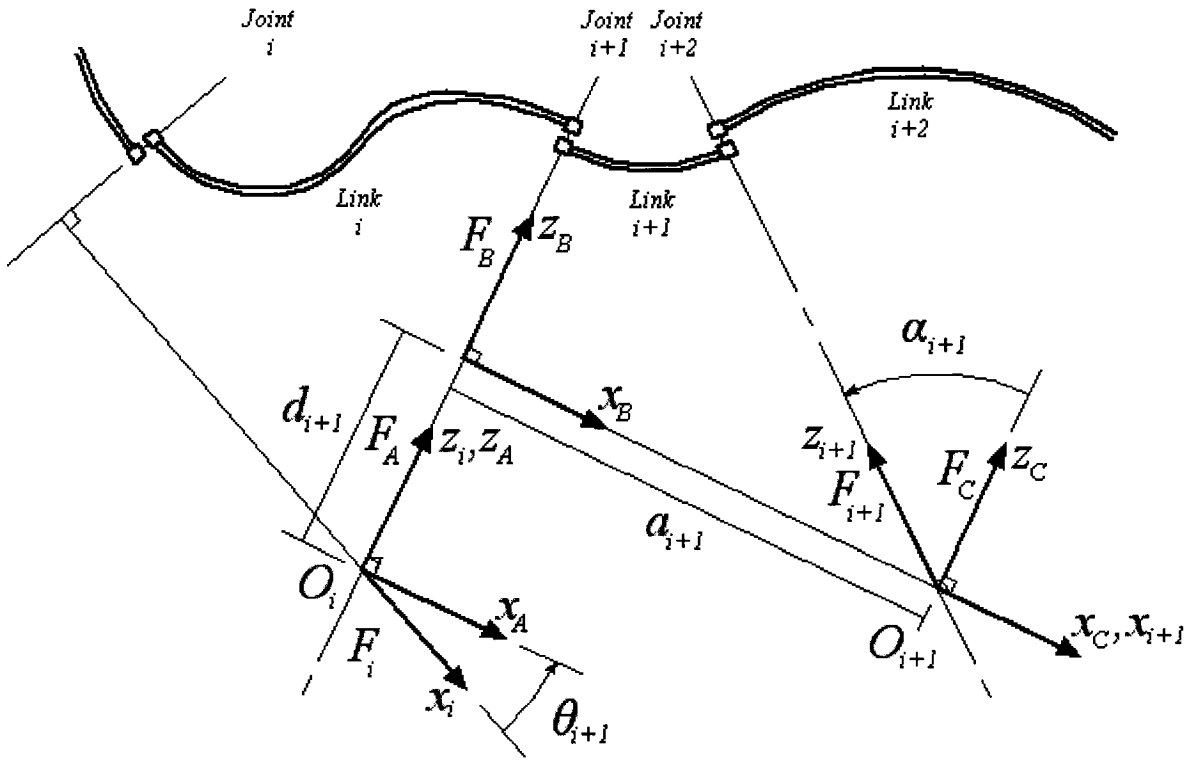

Figure 2.9: The Denavit-Hartenberg transformation.

constant parameters, based on the geometry of the link and the interconnection to the next link. The joint angle, θ, and the link offset, d, defined in the list below, are the parameters associated with the interconnection. The link length, a, and the link twist, α, are associated with the geometry of the link. As a and α are always constants, they represent a fixed relation between two successive joints.

With reference to Figure 2.9, the four independent DH parameters and their relations between adjacent frames, F_i and F_{i+1}, are defined as follows:

(i) θ_{i+1} is the angle of rotation about the z_i axis. The x_i axis is rotated by the angle θ_{i+1} to form an intermediate coordinate frame, F_A, whose x_A axis is parallel to the x_{i+1} axis and z_A axis coincident with the z_i axis.

(ii) d_{i+1} is a translation along the z_A axis. The second intermediate coordinate frame,

F_B, is obtained by translating the origin of F_A along the z_A axis a distance of d_{i+1}. The x_B axis of this coordinate frame is coincident with the x_{i+1} axis while the z_B axis remains coincident with the z_i axis.

(iii) a_{i+1} is a translation along the x_B axis. The third intermediate coordinate frame, F_C, is defined by translating the origin of the F_B along the x_B axis a distance of a_{i+1}. The origin of this frame is coincident with that of the final frame's origin, O_{i+1}. The x_C axis of this frame is coincident with the x_{i+1} axis and originates at the same point. The z_C axis of this frame is parallel to the z_i axis.

(iv) α_{i+1} is the angle of rotation about the x_C axis. The final frame is achieved by rotating the z_C axis with respect to the x_C axis by an angle of α_{i+1} to align itself with the z_{i+1} axis. Now, F_{i+1} is fully defined, with respect to F_i.

There are three intermediate frames so generated. These frames are important in the respect that, in general, the $(i+1)^{th}$ frame is always defined relative to the $(i)^{th}$ frame, and not the base reference frame. Thus, after an elementary displacement, in which an intermediate frame is generated, the next frame after that will be defined relative to that intermediate frame, and so on. Thus, all the transformation matrices involved in this procedure are post-multiplied, starting with the rotation of θ_{i+1} about z_i, denoted $\mathbf{T}_{Rotation}(z_i, \theta_{i+1})$. The expression, involving the elementary displacements, for the general DH transformation is thus obtained as:

$$^{i}\mathbf{T}_{i+1} = \mathbf{T}_{Rotation}(z_i, \theta_{i+1})\mathbf{T}_{Translation}(0, 0, d_{i+1})\mathbf{T}_{Translation}(a_{i+1}, 0, 0)\mathbf{T}_{Rotation}(x_C, \alpha_{i+1}). \tag{2.20}$$

The resulting general transformation matrix,

$$ {}^{i}\mathbf{T}_{i+1} = \begin{bmatrix} \cos\theta_{i+1} & -\sin\theta_{i+1}\cos\alpha_{i+1} & \sin\theta_{i+1}\sin\alpha_{i+1} & a_{i+1}\cos\theta_{i+1} \\ \sin\theta_{i+1} & \cos\theta_{i+1}\cos\alpha_{i+1} & -\cos\theta_{i+1}\sin\alpha_{i+1} & a_{i+1}\sin\theta_{i+1} \\ 0 & \sin\alpha_{i+1} & \cos\alpha_{i+1} & d_{i+1} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.21} $$

maps points in the $(i+1)^{th}$ frame to those in the i^{th} frame, and is denoted ${}^{i}\mathbf{T}_{i+1}$.

In general, for a series of N frames, the overall transformation between the first frame, F_0, and the last, F_N, would be the matrix multiplication of all the DH transformation matrices:

$$ {}^{0}\mathbf{T}_{N} = {}^{0}\mathbf{T}_{1}\ {}^{1}\mathbf{T}_{2} \cdots {}^{N-2}\mathbf{T}_{N-1}\ {}^{N-1}\mathbf{T}_{N}. \tag{2.22} $$

Thus, for a six degree-of-freedom robot, the overall robot transform would be:

$$ {}^{0}\mathbf{T}_{6} = {}^{0}\mathbf{T}_{1}\ {}^{1}\mathbf{T}_{2}\ {}^{2}\mathbf{T}_{3}\ {}^{3}\mathbf{T}_{4}\ {}^{4}\mathbf{T}_{5}\ {}^{5}\mathbf{T}_{6}. \tag{2.23} $$

This overall robot transformation relates points expressed in the tool flange reference frame to those of the same points expressed in the world coordinate frame. Equation 2.22 represents the forward kinematics of an N degree-of-freedom serially-connected robot. The DH parameterization model was not the only one pursued however, as the inverse kinematic problem could not be easily solved using this convention. The Modified Denavit-Hartenberg (MDH) parameterization, outlined in [32], allowed the use of Pieper's solution to the inverse kinematic problem and this convention will be discussed next.

2.1.3 Modified Denavit-Hartenberg Parameters

The MDH parameterization is very similar to the DH parameterization, but, as its name implies, there is a slight modification present. The four parameters, θ, d, a, and α are still used and they represent the same geometric quantities. However, there are important changes to the assignment of coordinate frames and to the order of elementary transformations accompanied by each parameter.

The assignment of coordinate frames follows the same rules as outlined in Section 2.1.2, with two exceptions. First, the origin of the i^{th} frame is located on the i^{th} joint axis. Second, the base frame, F_0, is taken into account. F_0 is assigned such that it mirrors F_1 when θ_1 is zero. The other important modification is that the order of the elementary transformations, corresponding to the robot parameters, is different. As was reported in Section 2.1.1, the order of transformations, pre-multiplication or post-multiplication, is significant. With reference to Figure 2.10, the expression for the MDH transformation is:

$$ {}^{i}\mathbf{T}_{i+1} = \mathbf{T}_{Rotation}(x_i, \alpha_i)\mathbf{T}_{Translation}(a_i, 0, 0)\mathbf{T}_{Rotation}(z_B, \theta_{i+1})\mathbf{T}_{Translation}(0, 0, d_{i+1}). \tag{2.24} $$

As seen in Equation 2.24, the indices for the a and α parameters are one increment behind the d and θ parameters. As stated previously, the four parameters represent the same basic operations, however, they are associated with the mechanical design of the robot. As the coordinate frames under the MDH parameterization are assigned under slightly different rules, the expressions for the transformations between frames involve different geometric quantities. In the case of the MDH parameterization, the parameters used in one transformation can be related to the geometry of a particular link and the interconnection between it and the previous link. In the DH model, the geometry of a link and the interconnection between it and the next link is utilized in a transformation

between frames. The MDH transformation matrix is:

$$
{}^{i}T_{i+1} = \begin{bmatrix} \cos\theta_{i+1} & -\sin\theta_{i+1} & 0 & a_i \\ \sin\theta_{i+1}\cos\alpha_i & \cos\theta_{i+1}\cos\alpha_i & -\sin\alpha_i & -d_{i+1}\sin\alpha_i \\ \sin\theta_{i+1}\sin\alpha_i & \cos\theta_{i+1}\sin\alpha_i & \cos\alpha_i & d_{i+1}\cos\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.25}
$$

With reference to Figure 2.10, the procedure for transforming the coordinates in the $(i+1)^{th}$ frame to those in the i^{th} frame is as follows:

(i) α_i is the angle of rotation about the x_i axis. The z_i axis is rotated by the angle α_i to form an intermediate coordinate frame, F_A, whose x_A axis is coincident to the x_i axis and z_A axis parallel with the z_{i+1} axis.

(ii) a_i is a translation along the x_A axis. The second intermediate coordinate frame, F_B,

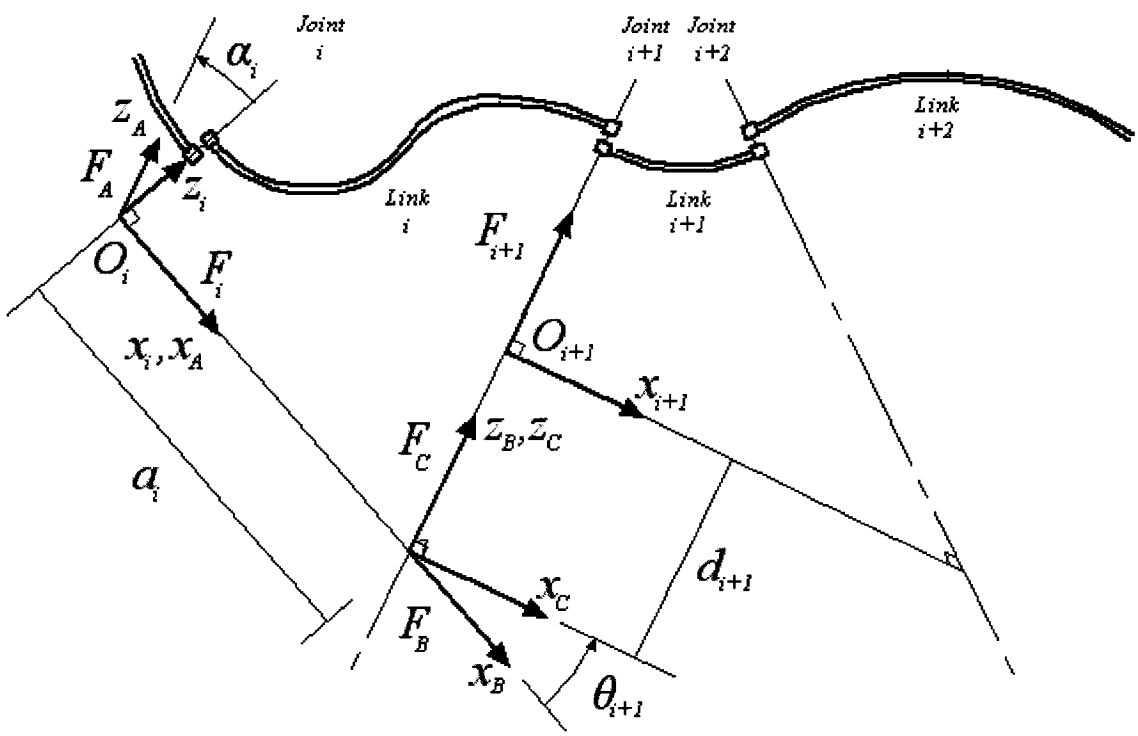

Figure 2.10: The Modified Denavit-Hartenberg transformation.

is obtained by translating the origin of F_A along the x_A axis a distance of a_i. The x_B axis of this coordinate frame is coincident with the x_i axis while the z_B axis is now coincident with the z_{i+1} axis.

(iii) θ_{i+1} is the angle of rotation about the z_B axis. The third intermediate coordinate frame, F_C, is defined by rotating the z_B axis of F_B by an angle of θ_{i+1}. The x_C axis of this frame is now parallel to the x_{i+1} axis and the z_C axis is coincident with the z_{i+1} axis.

(iv) d_{i+1} is a translation along the z_C axis. The final frame is achieved by translating a distance of d_{i+1} along the z_C axis. The $(i+1)^{th}$ frame is now fully defined with respect to the i^{th} frame.

The coordinate frames and robot parameters were assigned using both the DH and MDH models for two robots: the KUKA KR 15/2 and the Thermo CRS A465. A brief introduction will now be given regarding these two six-axis serial robots as well as the corresponding coordinate frames and robot parameters.

2.1.4 Forward Kinematics of the Thermo CRS A465

The Thermo CRS A465 six-axis serial robot is a relatively small manipulator and is reasonably precise, which is typical of small serial robots. It has a maximum reach of 711 mm and a maximum payload of 2 kg. It also has a stated repeatability of $\pm$ 50 μm. The manipulator has six degrees-of-freedom, which are provided by six actuated revolute joints. The axes of the last three of these joints intersect at the wrist-centre. Such an architecture is typically called *spherical wrist*, or *wrist-partitioned*. Figure 2.11 illustrates the A465 and the experiment setup.

The forward kinematics will be analyzed using the DH and MDH parameterizations. The DH frame assignments for the A465 are depicted in Figure 2.12, where the y_i axes

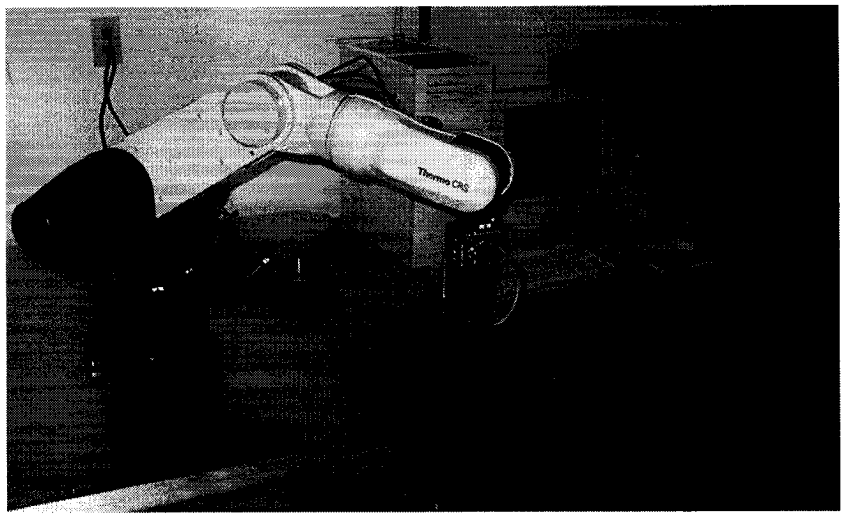

Figure 2.11: The Thermo CRS A465 experiment setup.

are not shown as they simply complete the right hand-rule for the x_i and z_i axes, and the parameters are listed in Table 2.1.

As all the joints for this robot are revolute, the i^{th} joint variable is the joint angle θ_{i+1}. For the second and third joint angles of the A465, an offset of $\frac{\pi}{2}$ exists in the values reported by the controller and therefore $\frac{\pi}{2}$ must be added to both of these joint angles. This is due to differences in the zero positions for these joints of the kinematic model resident in the controller versus those of the DH convention. Under the DH model, the forward kinematics relates the pose of the tool flange centre-point with respect to the base reference frame. An additional transformation, a translation, ${}^{6}\mathbf{T}_{T_T}$, can be included so

Table 2.1: The Thermo CRS A465 DH parameters.

$i+1$	θ_{i+1}	d_{i+1} (m)	a_{i+1} (m)	α_{i+1} (rad)
1	θ_1	0.330	0.000	$\frac{\pi}{2}$
2	θ_2	0.000	0.305	0
3	θ_3	0.000	0.000	$\frac{\pi}{2}$
4	θ_4	0.330	0.000	$-\frac{\pi}{2}$
5	θ_5	0.000	0.000	$\frac{\pi}{2}$
6	θ_6	0.076	0.000	0

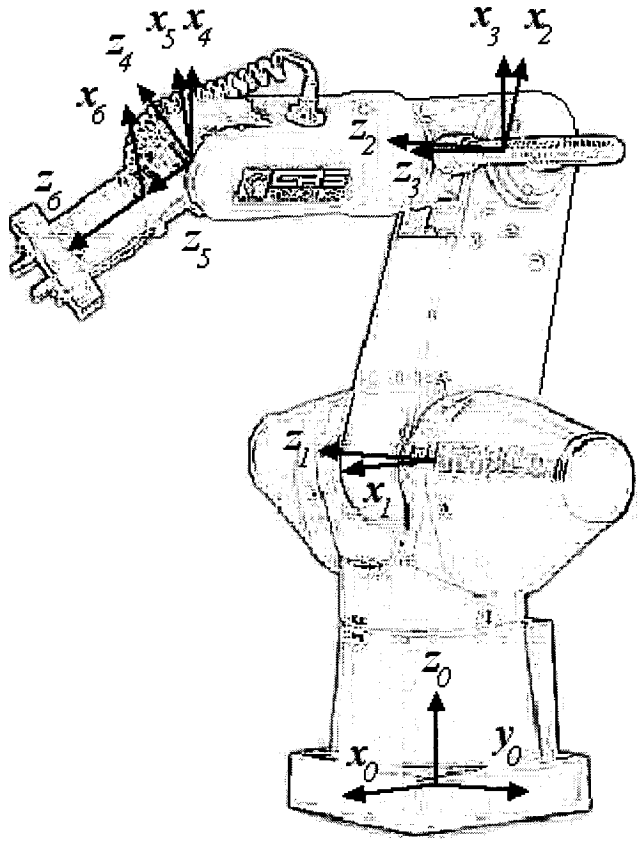

Figure 2.12: The Denavit-Hartenberg coordinate frames for the Thermo CRS A465.

that the last frame originates at the tool tip centre-point. Thus, the forward kinematics equation is:

$$^{0}\mathbf{T}_{T_T} = ^{0}\mathbf{T}_1 \; ^{1}\mathbf{T}_2 \; ^{2}\mathbf{T}_3 \; ^{3}\mathbf{T}_4 \; ^{4}\mathbf{T}_5 \; ^{5}\mathbf{T}_6 \; ^{6}\mathbf{T}_{T_T}. \tag{2.26}$$

The MDH frame assignments are depicted in Figure 2.13 and the parameters in Table 2.2. Note that the last three coordinate frames originate at the wrist-centre, the intersection of the fourth, fifth and sixth axes, in this representation. Also, in this model points described with respect to the wrist-centre are related to those in F_0, whose origin is at the intersection of the first and second joint axes. In order to incorporate the base reference frame and the tool flange centre-point in the model, two additional transformations can be included. These two transformations, $^{B}\mathbf{T}_0$ for the base and $^{6}\mathbf{T}_{T_F}$ for the tool

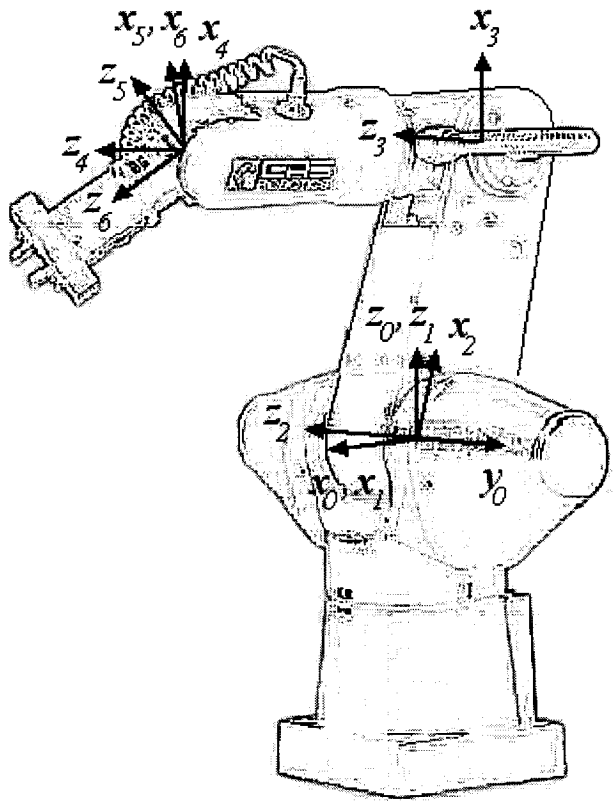

Figure 2.13: The Modified Denavit-Hartenberg coordinate frames for the Thermo CRS A465.

flange, can be employed in conjunction with the tool tip transformation, ${}^{T_F}\mathbf{T}_{T_T}$, so that points described with respect to the tool tip can be transformed. This relation is:

$$ {}^{B}\mathbf{T}_{T_T} = {}^{B}\mathbf{T}_0 \, {}^{0}\mathbf{T}_1 \, {}^{1}\mathbf{T}_2 \, {}^{2}\mathbf{T}_3 \, {}^{3}\mathbf{T}_4 \, {}^{4}\mathbf{T}_5 \, {}^{5}\mathbf{T}_6 \, {}^{6}\mathbf{T}_{T_F} \, {}^{T_F}\mathbf{T}_{T_T}. \tag{2.27} $$

Table 2.2: The Thermo CRS A465 MDH parameters.

$i+1$	θ_i	d_{i+1} (m)	a_i (m)	α_i (rad)
1	θ_1	0.000	0.000	0
2	θ_2	0.000	0.000	$\frac{\pi}{2}$
3	θ_3	0.000	0.305	0
4	θ_4	0.330	0.000	$\frac{\pi}{2}$
5	θ_5	0.000	0.000	$-\frac{\pi}{2}$
6	θ_6	0.000	0.000	$\frac{\pi}{2}$

As is implied, a relation for any useful point can be created provided the geometric information is available.

2.1.5 Forward Kinematics of the KUKA KR 15/2

The KUKA KR 15/2 is a larger industrial robot capable of carrying heavier loads. It has a repeatability of ± 0.1 mm and with a maximum payload 15 kg. The KR 15/2 was used in a preliminary experiment at the Mining University of Leöben [33] where the RMC was conceived. As access to this robot was not possible, a new robot, the Thermo CRS A465, was procured. However, during the initial stages of the project, the availability of a robot for experimentation was uncertain, thus all the simulations were programmed using the geometry of the KR 15/2. The original experimental setup is pictured in Figure 2.14.

The DH frame assignments for the KR 15/2 are depicted in Figure 2.15 and the DH parameters are listed in Table 2.3. Each of the joint angles reported by the controller must be multiplied by -1 and the third joint angle must have $\frac{\pi}{2}$ subtracted from it before the multiplication. The KR 15/2 has a noticeably larger workspace, as evidenced by the link lengths and offsets, and a slightly more complicated geometry than the A465. As this is an industrial robot that must accommodate relatively more massive equipment and objects, it has a more rigid construction. As with the A465, under this representation, points described with respect to the tool flange centre-point are related to their counterparts

Table 2.3: The KUKA KR 15/2 DH parameters.

$i+1$	θ_{i+1}	d_{i+1} (m)	a_{i+1} (m)	α_{i+1} (rad)
1	θ_1	0.675	0.300	$\frac{\pi}{2}$
2	θ_2	0.000	0.650	0
3	θ_3	0.000	0.155	$\frac{\pi}{2}$
4	θ_4	0.600	0.000	$-\frac{\pi}{2}$
5	θ_5	0.000	0.000	$\frac{\pi}{2}$
6	θ_6	0.140	0.000	0

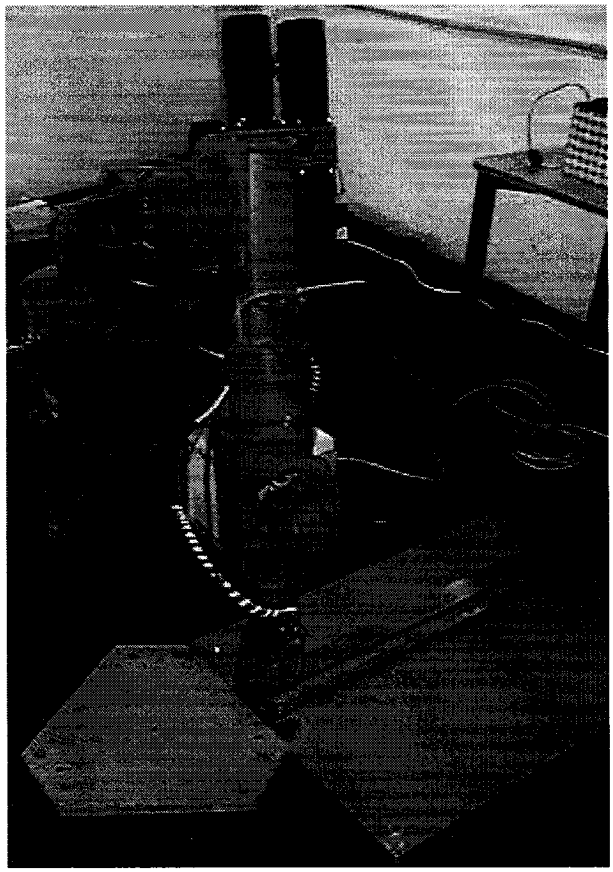

Figure 2.14: The KUKA KR 15/2 experimentation setup.

in the base reference frame. A tool tip transformation can be included as before with Equation 2.26.

The MDH frame assignments for the KR 15/2 are illustrated in Figure 2.16 and the MDH parameters are presented in Table 2.4. As with the A465, the last three coordinate frames originate at the wrist-centre. F_0 is located at the intersection of the first joint axis with the common normal between the first and second joint axes. Again, if it is desired to relate points with respect to the tool tip centre-point to their counterparts in the base frame, Equation 2.27 can be employed.

The forward kinematic problem of six-axis serial robots is easily solved employing matrix multiplication. This allows the computation of the pose of the tool flange, or tool tip, given a set of joint angles and the appropriate robot parameters. However, for

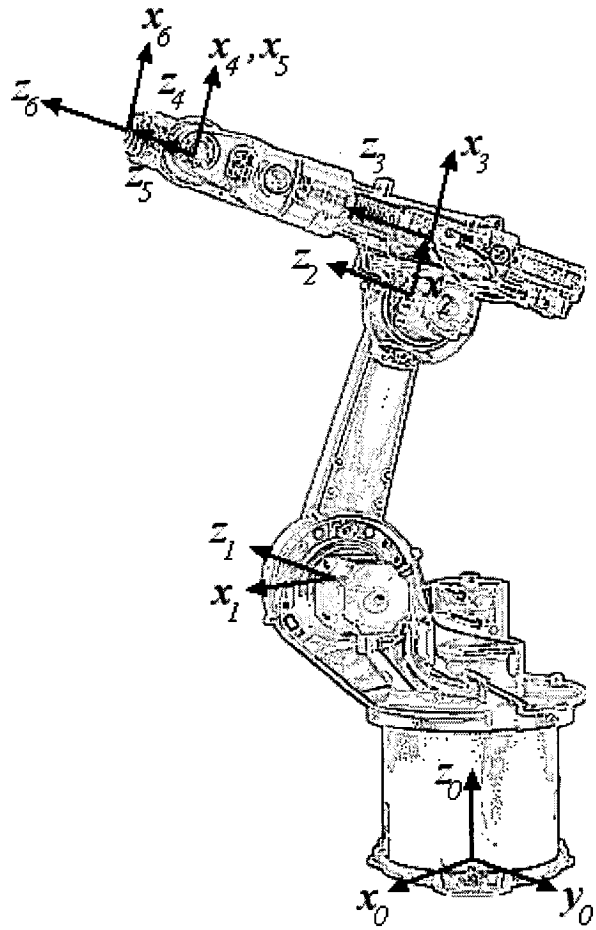

Figure 2.15: The Denavit-Hartenberg coordinate frames for the KUKA KR 15/2.

simulation purposes it is additionally required to generate a set of joint angles necessary to achieve some specified pose. This requires a solution to the inverse kinematic problem.

Table 2.4: The KUKA KR 15/2 MDH parameters.

$i+1$	θ_i	d_{i+1} (m)	a_i (m)	α_i (rad)
1	θ_1	0.000	0.000	0
2	θ_2	0.000	0.300	$\frac{\pi}{2}$
3	θ_3	0.000	0.650	0
4	θ_4	0.600	0.155	$\frac{\pi}{2}$
5	θ_5	0.000	0.000	$-\frac{\pi}{2}$
6	θ_6	0.000	0.000	$\frac{\pi}{2}$

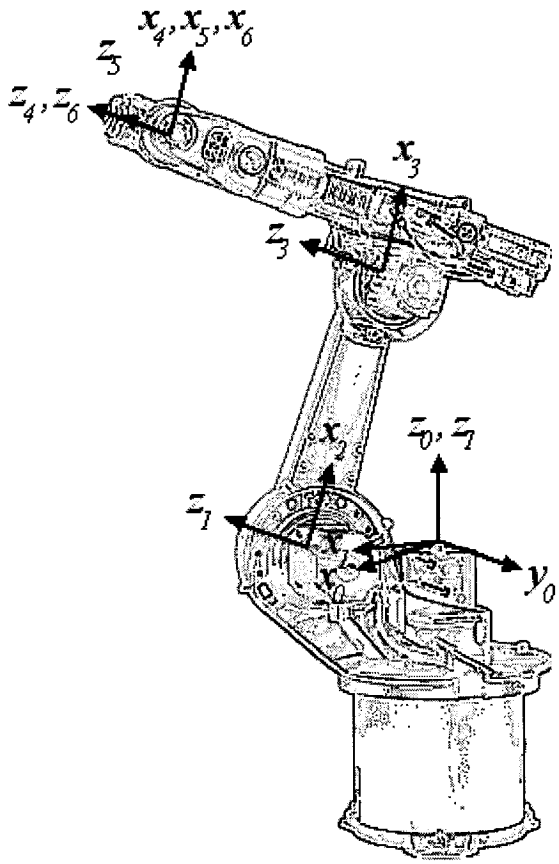

Figure 2.16: The Modified Denavit-Hartenberg coordinate frames for the KUKA KR 15/2.

Two such solutions are described in the following.

2.2 Inverse Kinematics

Inverse kinematics is the process by which a set of joint variables are computed that allow the end-effector to attain a prescribed pose. This is required, and utilized in the simulations, when the end-effector of the robot is commanded to move to a point within its workspace while maintaining a constant orientation. Essentially, the only difference between the computed forward kinematics for these two poses would be the fourth column, the translation vector. From the 4×4 matrix, given the numerical value of the elements,

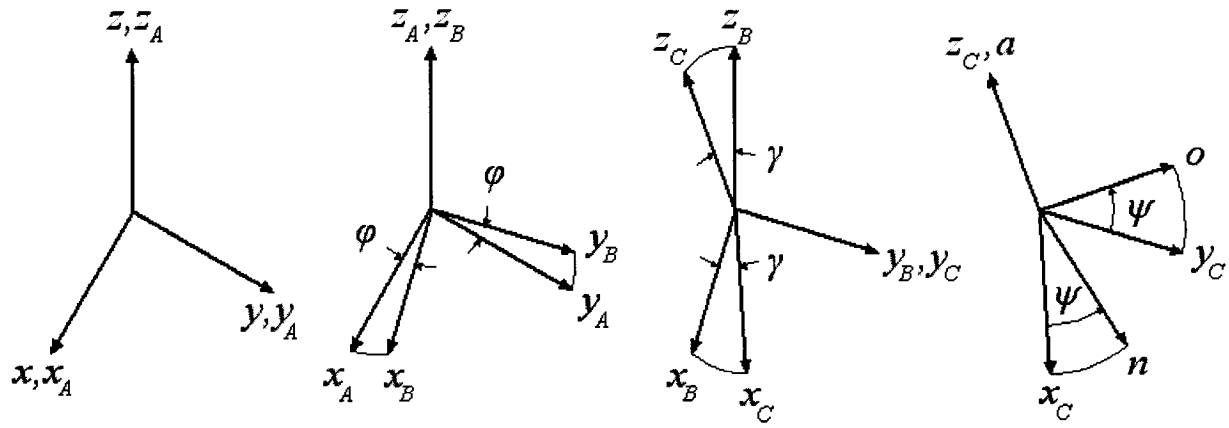

Figure 2.17: Euler angles used in the transformation from F_{xyz} to F_{noa}.

the joint angles can be computed using the geometric constraints on the elements. Each of the elements of this matrix correspond to a kinematic equation that depends on the joint variables and the robot parameters. Two solutions to this problem will be presented: Pieper's solution [34], which utilizes the MDH parameters, and a closed-form solution involving the DH parameterization. In order to develop Pieper's solution, a brief description of Euler angles will be presented.

2.2.1 Euler Angles

Euler angles are a means to represent orientation. There are twelve Euler angle conventions and twelve fixed angle conventions ??. Euler angles are defined with respect to the revolved intermediate frames, and therefore are post-multiplied, while the fixed angle representations are expressed with respect to a stationary reference frame and are pre-multiplied. In all of these conventions, three rotations are performed. As the two robots of interest both conform to the *Z-Y-Z* Euler angle representation, only it will be discussed.

Consider the frame, F_A, which has the same orientation and origin as the base reference frame, F_{xyz}, in the leftmost image of Figure 2.17. It is now desired to achieve a new orientation having the Euler angles ϕ, γ, and ψ. These angles are expressed relative to

axes z_A, y_B, and z_C, respectively, of the intermediate frames, in this specific order. The resulting transformation matrix describing coordinate frame F_{noa} can be computed by concatenating the three rotations:

$$^{xyz}\mathbf{T}_{noa} = \mathbf{T}_{Rotation}(z_A, \phi)\ \mathbf{T}_{Rotation}(y_B, \gamma)\ \mathbf{T}_{Rotation}(z_C, \psi). \tag{2.28}$$

A visual representation of this series of transformations can be viewed in the three right images of Figure 2.17. The transformation matrix for the *Z-Y-Z* convention is:

$$^{xyz}\mathbf{T}_{noa} = \begin{bmatrix} \cos\phi\cos\gamma\cos\psi - \sin\phi\sin\psi & -\cos\phi\cos\gamma\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\gamma & 0 \\ \sin\phi\cos\gamma\cos\psi + \cos\phi\sin\psi & -\sin\phi\cos\gamma\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\gamma & 0 \\ -\sin\gamma\cos\psi & \sin\gamma\sin\psi & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.29}$$

The rotation sub-matrix of this general transformation matrix has nine elements, but is only dependent on three variables. There are six constraints on orthonormal rotation matrices. Each column vector has a magnitude of one and so this provides three constraint equations. As these vectors are mutually orthogonal, the dot products of the column vectors account for the other three constraint equations. Thus, any orientation is dependent on just three variables.

Given a transformation matrix, such as in Equation 2.8, where the elements have numerical values, the Euler angles can be extracted according to [32]:

$$\begin{aligned} \gamma &= atan2(\sqrt{n_z^2 + o_z^2}, a_z) \\ \phi &= atan2(\frac{a_y}{\sin\gamma}, \frac{a_x}{\sin\gamma}) \\ \psi &= atan2(\frac{o_z}{\sin\gamma}, -\frac{n_z}{\sin\gamma}). \end{aligned} \tag{2.30}$$

This solution degenerates when γ equals zero or $\frac{\pi}{2}$. In these cases, ϕ equals zero and ψ can be computed by $atan2(-o_x, n_x)$ and $atan2(o_x, -n_x)$, respectively. The $atan2$

function computes the inverse tangent of $\frac{y}{x}$, but uses their respective signs to determine the quadrant of the calculated angle.

Based on this approach, the joint variables that strictly affect the orientation of the end-effector can now be identified in an inverse kinematic solution. For wrist-partitioned serial robots, the first three controlled joint variables, θ_1, θ_2, and θ_3, are used to position the wrist-centre. The last three, θ_4, θ_5, and θ_6, comprising the spherical wrist, define the orientation of the end-effector. An established method for the solution of the inverse kinematic problem for wrist-partitioned robots with six revolute joints will now be described.

2.2.2 Pieper's Method

Pieper's solution [34] applies to manipulators which have six degrees-of-freedom where three consecutive coordinate frames have a common origin, and thus three consecutive axes that intersect. This class of robot, as previously stated, is known as wrist-partitioned. As noted with the forward kinematics of the A465 and the KR 15/2, under the MDH parameterization this stipulation is satisfied. However, under the DH model, the tool flange dimension is included in the last transformation for both robots so that the last three coordinate frames do not originate at a common point. This is computationally inconvenient, hence the MDH model is used.

Pieper's method begins by expressing the origin of the three intersecting axes, the wrist-centre, in the base reference frame coordinates. This is accomplished by:

$$ {}^0\mathbf{p}_4 = {}^0\mathbf{T}_1 \; {}^1\mathbf{T}_2 \; {}^2\mathbf{T}_3 \; {}^3\mathbf{p}_4. \tag{2.31} $$

The position of the origin of the fourth frame with respect to the third, ${}^3\mathbf{p}_4$, is simply the fourth column of the fourth transformation matrix, ${}^3\mathbf{T}_4$. If this substitution is made,

then:

$$ {}^0\mathbf{p}_4 = {}^0\mathbf{T}_1 \; {}^1\mathbf{T}_2 \; {}^2\mathbf{T}_3 \begin{pmatrix} a_3 \\ -d_4 \sin\alpha_3 \\ d_4 \cos\alpha_3 \\ 1 \end{pmatrix}. \tag{2.32} $$

Applying the third transformation matrix, ${}^2\mathbf{T}_3$, to the ${}^3\mathbf{p}_4$ position vector yields:

$$ {}^0\mathbf{p}_4 = {}^0\mathbf{T}_1 \; {}^1\mathbf{T}_2 \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ 1 \end{pmatrix}, \tag{2.33} $$

where the resulting expressions for f_1, f_2, and f_3 are:

$$ \begin{aligned} f_1 &= a_3 \cos\theta_3 + d_4 \sin\alpha_3 \sin\theta_3 + a_2, \\ f_2 &= a_3 \cos\alpha_2 \sin\theta_3 - d_4 \sin\alpha_3 \cos\alpha_2 \cos\theta_3 - d_4 \sin\alpha_2 \cos\alpha_3 - d_3 \sin\alpha_2, \\ f_3 &= a_3 \sin\alpha_2 \sin\theta_3 - d_4 \sin\alpha_3 \sin\alpha_2 \cos\theta_3 + d_4 \cos\alpha_2 \cos\alpha_3 + d_3 \cos\alpha_2. \end{aligned} \tag{2.34} $$

Applying the remaining two transformations, ${}^0T_1$ and ${}^1T_2$, yields an expression for ${}^0\mathbf{p}_4$ which involves three new terms, g_1, g_2, and g_3, based on f_1, f_2, and f_3, and the MDH parameters associated with the two transformations. The final expression for ${}^0\mathbf{p}_4$ is:

$$ {}^0\mathbf{p}_4 = \begin{pmatrix} \cos\theta_1 g_1 - \sin\theta_1 g_2 \\ \sin\theta_1 g_1 + \cos\theta_1 g_2 \\ g_3 \end{pmatrix}, \tag{2.35} $$

where the expressions for g_1, g_2 and g_3 are:

$$\begin{aligned} g_1 &= \cos\theta_2 f_1 - \sin\theta_2 f_2 + a_1, \\ g_2 &= \sin\theta_2 \cos\alpha_1 f_1 + \cos\theta_2 \cos\alpha_1 f_2 - \sin\alpha_1 f_3 - d_2 \sin\alpha_1, \\ g_3 &= \sin\theta_2 \sin\alpha_1 f_1 + \cos\theta_2 \sin\alpha_1 f_2 + \cos\alpha_1 f_3 + d_2 \cos\alpha_1. \end{aligned} \tag{2.36}$$

An expression for the squared magnitude of ${}^0\mathbf{p}_4$ is now required and simplifies to:

$$r^2 = g_1^2 + g_2^2 + g_3^2. \tag{2.37}$$

This result is due to the application of the trigonometric identity:

$$\cos^2\theta + \sin^2\theta = 1. \tag{2.38}$$

Through substitution and simplification, an expression for r^2 can obtained:

$$r^2 = f_1^2 + f_2^2 + f_3^2 + a_1^2 + d_2^2 + 2d_2 f_3 + 2a_1(f_1 \cos\theta_2 - f_2 \sin\theta_2). \tag{2.39}$$

By introducing four new terms, k_1, k_2, k_3, and k_4, and rewriting Equation 2.39, a system of two equations can be produced when combined with the z component of Equation 2.35. This system is:

$$r^2 = (k_1 \cos\theta_2 + k_2 \sin\theta_2)2a_1 + k_3, \tag{2.40}$$

$$z = (k_1 \sin\theta_2 - k_2 \cos\theta_2) \sin\alpha_1 + k_4, \tag{2.41}$$

where k_1, k_2, k_3, and k_4 are defined as

$$k_1 = f_1,$$

$$\begin{aligned} k_2 &= -f_2, \\ k_3 &= f_1^2 + f_2^2 + f_3^2 + a_1^2 + d_2^2 + 2d_2 f_3, \\ k_4 &= f_3 \cos\alpha_1 + d_2 \cos\alpha_1. \end{aligned} \tag{2.42}$$

This system of equations is dependent on θ_2 and θ_3, but the dependence on θ_2 is of a simple form.

Now, there are three possible cases for this system of equations. Two cases can be attributed to the two possible ways the dependence on θ_2 can be eliminated in a simple fashion, while the third accounts for the general case where a trigonometric identity is used to eliminate the dependence. The equation used to solve for θ_3 in the third case is undefined when the stipulations of either of the first two cases are satisfied.

Case 1, $a_1 = 0$: When a_1 equals zero, Equation 2.40 simplifies to the solution of k_3 with r^2 known. A transcendental equation, involving $\cos\theta_3$ and $\sin\theta_3$, results from the expansion of this equation. Geometric identities, seen in Equation 2.44, are used to simplify the expression in terms of a single variable. The resulting quadratic equation can then be solved. As multiple roots exist, a decision regarding the appropriate solution must be made.

Case 2, $\sin\alpha_1 = 0$: When $\sin\alpha_1$ equals zero, the same procedure is followed except the solution of k_4 is performed with z known. For both of these cases, once a solution for θ_3 has been computed, θ_2 is solved by using the other available, either Equation 2.40 or 2.41, and θ_1 is obtained through the use of either the first or second element of Equation 2.35.

Case 3, $a_1 \neq 0$ and $\sin\alpha_1 \neq 0$: For the final case, $\sin\theta_2$ and $\cos\theta_2$ are eliminated from

Equation 2.40 and 2.41, using the trigonometric identity of Equation 2.38 to produce:

$$\frac{(r-k_3)^2}{4a_1^2} + \frac{(z-k_4)^2}{\sin^2\alpha_1} = k_1^2 + k_2^2, \tag{2.43}$$

which is, upon the substitution of the two geometric identities:

$$\begin{aligned} \cos\eta &= \tfrac{1-u^2}{1+u^2} \\ \sin\eta &= \tfrac{2u}{1+u^2}. \end{aligned} \tag{2.44}$$

a 4^{th} order polynomial. The roots of this polynomial are obtained and an acceptable solution chosen. As before, the first two joint angles, θ_2 and θ_1, can be attained through the solution of one of Equations 2.40 and 2.41, and then 2.35, respectively.

The last three joint angles follow the *Z-Y-Z* Euler angle convention and are solved using the method presented in Section 2.2.1. Recall that three rotations are performed with Euler angles and that the last three coordinate frames under the MDH model intersect at a common point. Thus, the last three transformations, which define the orientation of the end-effector, are just rotations. As the first three joint angles are known at this point, the orientation of the fourth frame can be computed when θ_4 equals zero. The difference between this orientation and the specified orientation of the given pose is strictly due to the three rotations, θ_4, θ_5, and θ_6:

$${}^4\mathbf{T}_6|_{\theta_4=0} = {}^0\mathbf{T}_4^{-1}|_{\theta_4=0}\ {}^0\mathbf{T}_6. \tag{2.45}$$

Pieper's method yields 32 possible solutions. To solve for θ_3 one must obtain the roots of a 4^{th} order polynomial and two 2^{nd} order polynomials for θ_2 and θ_1. There are also two solution sets for the last three joint angles, θ_4, θ_5, and θ_6, when using Euler angles due to the square root of Equation 2.30. Thus, there are 32 possible outcomes. However, some

of the roots may be complex conjugate pairs which are immediately discounted.

Obviously, some comparison must be made to select one of the 32 possible solutions. In the simulation, the end-effector is commanded to move in small increments along the length of a straight ruler, the difference between the two joint angle sets of the adjacent poses should be minimal. Thus, when solving for a joint angle, whichever solution is closest to the same joint angle of the previous pose is selected.

As Pieper's solution to the inverse kinematic problem can only be applied to robots which have three consecutive coordinate frames, each originating at the same point, the use of the MDH representational scheme was required. The solution, for all three cases, can be found in Appendix A. However, a solution was required for the use of the DH model, as it was not desired to switch between parameterizations within the simulation and the final results of the simulation were more successful with the DH parameters. A closed form solution, based on the systematic isolation of the joint variables will now be developed.

2.2.3 Closed Form Solution Using DH Parameters

The general robot transformation matrix for both the A465 and the KR 15/2 under the DH model relates the position and orientation of the tool flange centre-point to the base reference frame. Since this point is described, as opposed to the wrist-centre point, its position is dependent upon all six joint variables. As both robots are wrist-partitioned, it is possible to solve for θ_1, θ_2, and θ_3 by eliminating this dependency on θ_4, θ_5, and θ_6. By translating back to the wrist-centre, which is easily accomplished with the inclusion of an additional post-multiplied transformation using the tool flange geometry, the translational component of the general matrix is then dependent upon the first three joint variables, θ_1, θ_2, and θ_3. The rotational component is unaffected and still represents the orientation of the tool flange. As both a_6 and α_6 are nominally zero in both cases, no influence is

provided by these parameters. Thus, a translation of $-d_6$ relative to the final frame, F_6, generates a new frame, F_W, that originates at the wrist-centre point.

In general, the transformation matrix that relates points defined in F_W to the base reference frames is:

$$ {}^0\mathbf{T}_W = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.46} $$

The general matrix is obtained by the matrix multiplication of the DH transformations between frames plus the additional translation by:

$$ {}^0\mathbf{T}_W = {}^0\mathbf{T}_1 \; {}^1\mathbf{T}_2 \; {}^2\mathbf{T}_3 \; {}^3\mathbf{T}_4 \; {}^4\mathbf{T}_5 \; {}^5\mathbf{T}_6 \; {}^6\mathbf{T}_W. \tag{2.47} $$

By pre-multiplying both sides of Equation 2.47 by the inverse of ${}^0\mathbf{T}_1$ and inspecting the elements of the resultant matrices, a suitable equation can be found that isolates the first joint variable [35]. The resulting matrices correspond to ${}^1\mathbf{T}_W$. After solving for θ_1, another pre-multiplication is performed and a solution for θ_2 is attempted. This matrix corresponds to ${}^2\mathbf{T}_W$. Once a solution for θ_2 is obtained, θ_3 is found in the same manner. The final three joint angles, θ_4, θ_5, and θ_6, can be solved for with one more inverse pre-multiplication, using different elements for each. To solve for the joint angles, the trigonometric identity of Equation 2.38 and the geometric identities of Equation 2.44 have to be used. Again, as polynomial expressions result in as many as 32 roots, comparisons must be made to select the appropriate solution set.

The complete solution for the inverse kinematics for the A465 is located in Appendix B. The solution for KR 15/2 follows the same solution and involves the same elements. The only difference is that it has a slightly more complicated geometry, which affects the

complexity of the terms.

The general form of the forward kinematics is found analytically and then the pre-multiplication steps are performed. The six joint angles for the A465 can be computed with Equations 2.48- 2.60. The first angle is determined by:

$$\theta_1 = atan2(p_y, p_x). \tag{2.48}$$

To obtain the second joint angle, a quadratic equation must be solved. The roots of this quadratic expression,

$$\begin{aligned}
&(2\sin\theta_1 p_y a_2 + \cos^2\theta_1 p_x^2 + d_1^2 + 2\cos\theta_1 p_x \sin\theta_1 p_y + a_2^2 + 2\cos\theta_1 p_x a_2 \\
&+p_z^2 + p_y^2 - 2p_z d_1 - p_y^2\cos^2\theta_1 - d_4^2)u_2^2 + (-4a_2 p_z + 4a_2 d_1)u_2 + \cos^2\theta_1 p_x^2 \\
&+d_1^2 + 2\cos\theta_1 p_x \sin\theta_1 p_y + a_2^2 - 2\cos\theta_1 p_x a_2 + p_z^2 + p_y^2 - 2p_z d_1 \\
&-p_y^2\cos^2\theta_1 - d_4^2 - 2\sin\theta_1 p_y a_2 = 0,
\end{aligned} \tag{2.49}$$

are substituted into:

$$\theta_2 = 2\tan^{-1}(u_2), \tag{2.50}$$

to obtain θ_2. u_2 corresponds to the single variable substitution for $\cos\theta_2$ and $\sin\theta_2$ with Equation 2.44 for the second joint angle.

The third joint angle is obtained with:

$$\theta_3 = \tan^{-1}\left(\frac{\cos\theta_2\cos\theta_1 p_x + \cos\theta_2\sin\theta_1 p_y + \sin\theta_2 p_z - \sin\theta_2 d_1 - a_2}{\sin\theta_2\cos\theta_1 p_x + \sin\theta_2\sin\theta_1 p_y - \cos\theta_2 p_z + \cos\theta_2 d_1}\right). \tag{2.51}$$

The fourth joint angle is obtained from:

$$\theta_4 = atan2\left(y_4, x_4\right), \tag{2.52}$$

where the variables y_4 and x_4 are:

$$\begin{aligned} y_4 &= \sin\theta_1 a_x - \cos\theta_1 a_y, && (2.53) \\ x_4 &= \cos\theta_3\cos\theta_2\cos\theta_1 a_x + \cos\theta_3\cos\theta_2\sin\theta_1 a_y + \cos\theta_3\sin\theta_2 a_z - \\ &\quad \sin\theta_3\sin\theta_2\cos\theta_1 a_x - \sin\theta_3\sin\theta_2\sin\theta_1 a_y + \sin\theta_3\cos\theta_2 a_z. && (2.54) \end{aligned}$$

The fifth angle is:

$$\theta_5 = atan2\left(y_5, x_5\right), \tag{2.55}$$

where the terms y_5 and x_5 are defined as:

$$\begin{aligned} y_5 &= \cos\theta_4\cos\theta_3\cos\theta_2\cos\theta_1 a_x + \cos\theta_4\cos\theta_3\cos\theta_2\sin\theta_1 a_y + \cos\theta_4\cos\theta_3\sin\theta_2 a_z - \\ &\quad \cos\theta_4\sin\theta_3\sin\theta_2\cos\theta_1 a_x - \cos\theta_4\sin\theta_3\sin\theta_2\sin\theta_1 a_y + \cos\theta_4\sin\theta_3\cos\theta_2 a_z + \\ &\quad \sin\theta_4\sin\theta_1 a_x - \sin\theta_4\cos\theta_1 a_y, && (2.56) \\ x_5 &= \sin\theta_3\cos\theta_2\cos\theta_1 a_x + \sin\theta_3\cos\theta_2\sin\theta_1 a_y + \sin\theta_3\sin\theta_2 a_z + \\ &\quad \cos\theta_3\sin\theta_2\cos\theta_1 a_x + \cos\theta_3\sin\theta_2\sin\theta_1 a_y - \cos\theta_3\cos\theta_2 a_z. && (2.57) \end{aligned}$$

Finally, the sixth joint angle is:

$$\theta_6 = atan2\left(y_6, x_6\right), \tag{2.58}$$

for which y_6 and x_6 are given as:

$$\begin{aligned} y_6 &= \sin\theta_3\cos\theta_2\cos\theta_1 o_x + \sin\theta_3\cos\theta_2\sin\theta_1 o_y + \sin\theta_3\sin\theta_2 o_z \\ &\quad \cos\theta_3\sin\theta_2\cos\theta_1 o_x + \cos\theta_3\sin\theta_2\sin\theta_1 o_y - \cos\theta_3\cos\theta_2 o_z, && (2.59) \\ x_6 &= \sin\theta_3\cos\theta_2\cos\theta_1 n_x + \sin\theta_3\cos\theta_2\sin\theta_1 n_y + \sin\theta_3\sin\theta_2 n_z \\ &\quad \cos\theta_3\sin\theta_2\cos\theta_1 n_x + \cos\theta_3\sin\theta_2\sin\theta_1 n_y - \cos\theta_3\cos\theta_2 n_z. && (2.60) \end{aligned}$$

The first three joint angles for the KR 15/2 can be computed with Equations 2.61-2.63. The solutions for the last three mirror those of the A465. Thus, θ_4, θ_5, and θ_6 can be obtained with Equations 2.52-2.60. For the KUKA geometry, the first joint angle is:

$$\theta_1 = atan2(p_y, p_x). \tag{2.61}$$

Both the second and third joint angles require the solution of quadratic equations. In accordance with the solution of θ_2 for the A465, the roots of these equations,

$$\begin{aligned}&(-2a_2a_1 + p_z^2 + d_1^2 + a_2^2 - a_3^2 - d_4^2 + 2\sin\theta_1 p_y a_2 + 2\cos\theta_1 p_x a_2 + p_y^2 + a_1^2 + \\ &2\cos\theta_1 p_x \sin\theta_1 p_y - p_y^2\cos^2\theta_1 - 2p_z d_1 + \cos^2\theta_1 p_x^2 - 2\cos\theta_1 p_x a_1 - 2\sin\theta_1 p_y a_1)u_2^2 + \\ &(-4a_2p_z + 4a_2d_1)u_2 - d_4^2 - a_3^2 + \cos^2\theta_1 p_x^2 - p_y^2\cos^2\theta_1 - 2\sin\theta_1 p_y a_1 - 2\cos\theta_1 p_x a_1 + \\ &p_y^2 + a_2^2 + d_1^2 + a_1^2 + 2\cos\theta_1 p_x \sin\theta_1 p_y + 2a_2a_1 - 2p_z d_1 - 2\cos\theta_1 p_x a_2 - 2\sin\theta_1 p_y a_2 + \\ &p_z^2 = 0\end{aligned} \tag{2.62}$$

and

$$\begin{aligned}&(\cos\theta_2 \sin\theta_1 p_y + \cos\theta_2 \cos\theta_1 p_x + \sin\theta_2 p_z - \cos\theta_2 a_1 - a_2 - \sin\theta_2 d_1 a_3)u_3^2 - 2d_4u_3 + \\ &\cos\theta_2 \cos\theta_1 p_x - a_3 + \cos\theta_2 \sin\theta_1 p_y - \cos\theta_2 a_1 + \sin\theta_2 p_z - \sin\theta_2 d_1 - a_2 = 0,\end{aligned} \tag{2.63}$$

are substituted into:

$$\theta_2 = 2\tan^{-1}(u_2) \tag{2.64}$$

and

$$\theta_3 = 2\tan^{-1}(u_3), \tag{2.65}$$

respectively.

With the equations provided, a direct solution for the joint variables for the A465 and

the KR 15/2 can be computed. As long as a reference pose is specified and the difference between sequential poses is suitably small, an inverse kinematic solution can be achieved. The final aspect of robot kinematics to be discussed, before the development of the RMC, is the Identification Jacobian. This form of the Jacobian relates velocities in Cartesian space to the joint rates of a robot.

2.3 Identification Jacobian

Thus far, the computation of the position and orientation, based on the geometry of the robot and the joint variables, has been presented along with the inverse process of computing the joint variables necessary for a specific pose. The next tool necessary for the calibration procedure is to be able to determine the relation between small variances in end-effector position to the unavoidable variations in the nominal robot geometry. These two quantities are related by the Identification Jacobian, which is the same as the robot Jacobian, but with columns corresponding to the other robot parameters. The robot Jacobian relates the linear and angular velocities of the end-effector to the joint rates, while the Identification Jacobian relates variations in robot geometry to the end-effector position under the assumption that the differences are small compared to the nominal robot parameter values. In this section, the Identification Jacobian will be defined for the purpose of robot calibration.

A Jacobian, in the mathematical sense, is a multi-dimensional form of the derivative represented in a matrix. It is a time-varying linear transformation. Given a set of functions, each dependent on a common set of independent parameters, a matrix can be constructed where the elements are the derivatives of a specific function with respect to a specific variable. The rows correspond to the functions while the columns are associated with the

variables. So, considering a set of q functions,

$$\begin{aligned} y_1 &= f(x_1, x_2, \ldots, x_r) \\ y_2 &= f(x_1, x_2, \ldots, x_r) \\ &\vdots \qquad\qquad \vdots \\ y_q &= f(x_1, x_2, \ldots, x_r), \end{aligned} \tag{2.66}$$

each dependent on a set of r variables, the differentials of y_i, as functions of the x_i, can be calculated with chain rule:

$$\begin{aligned} \delta y_1 &= \tfrac{\partial y_1}{\partial x_1}\delta x_1 + \tfrac{\partial y_1}{\partial x_2}\delta x_2 + \cdots + \tfrac{\partial y_1}{\partial x_r}\delta x_r \\ \delta y_2 &= \tfrac{\partial y_2}{\partial x_1}\delta x_1 + \tfrac{\partial y_2}{\partial x_2}\delta x_2 + \cdots + \tfrac{\partial y_2}{\partial x_r}\delta x_r \\ &\vdots \qquad\qquad\qquad\qquad \vdots \\ \delta y_q &= \tfrac{\partial y_q}{\partial x_1}\delta x_1 + \tfrac{\partial y_q}{\partial x_2}\delta x_2 + \cdots + \tfrac{\partial y_q}{\partial x_r}\delta x_r. \end{aligned} \tag{2.67}$$

When this set of equations is converted to vector-matrix form,

$$\begin{pmatrix} \delta y_1 \\ \delta y_2 \\ \vdots \\ \delta y_q \end{pmatrix} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_r} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_r} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_q}{\partial x_1} & \frac{\partial y_q}{\partial x_2} & \cdots & \frac{\partial y_q}{\partial x_r} \end{bmatrix} \begin{pmatrix} \delta x_1 \\ \delta x_2 \\ \vdots \\ \delta x_r \end{pmatrix}, \tag{2.68}$$

the Jacobian matrix appears. The compact form of this relation is:

$$\delta \mathbf{y} = \mathbf{J}\delta \mathbf{x}, \tag{2.69}$$

where $\mathbf{J}$ is the Jacobian.

In terms of calibration, the Identification Jacobian is used to identify variations from the nominal robot parameters. By their very nature, they are relatively small variations

compared to the nominal parameters, thus the Jacobian can be used in this manner. There are four types of robot parameters: θ, d, a, and α. As stated previously, they correspond to the joint angle, link offset, link length and link twist, respectively.

Some form of measurement must be used in a calibration procedure to identify the variations in these parameters. In the case of the RMC, only those measurements that relate to position will be considered. The RMC is novel form because endpoint errors are expressed relative to an arbitrary reference point, rather than the base frame origin. p_x, p_y, and p_z, of the general robot transform, which are functions of the joint variables and robot parameters, are of key importance. An Identification Jacobian relates the measured differences in end-effector position, δp_x, δp_y, and δp_z, to errors between the nominal robot geometry and the actual robot geometry. It is a *pose-dependent* linear transformation in that the value of its elements change according to the robot's joint angles. However, the elements of the Jacobian also depend on the nominal values for d, a, and α. The three kinematic equations used to calculate p_x, p_y, and p_z are differentiated with respect to each modelled kinematic error, n, forming a matrix of size $3 \times (N \times n)$, where N is the number of degrees-of-freedom. The Identification Jacobian:

$$[\mathbf{J}] = \begin{bmatrix} \mathbf{J}_{\theta_{1..n}} & \mathbf{J}_{d_{1..n}} & \mathbf{J}_{a_{1..n}} & \mathbf{J}_{\alpha_{1..n}} \end{bmatrix}, \tag{2.70}$$

can be partitioned into four distinct sub-Jacobians, each correlating to one type of robot

parameter. Each of the sub-matrices are defined by:

$$\mathbf{J}_\theta = \begin{bmatrix} \frac{\partial p_x}{\partial \theta_1} & \frac{\partial p_x}{\partial \theta_2} & \cdots & \frac{\partial p_x}{\partial \theta_n} \\ \frac{\partial p_y}{\partial \theta_1} & \frac{\partial p_y}{\partial \theta_2} & \cdots & \frac{\partial p_y}{\partial \theta_n} \\ \frac{\partial p_z}{\partial \theta_1} & \frac{\partial p_z}{\partial \theta_2} & \cdots & \frac{\partial p_z}{\partial \theta_n} \end{bmatrix}, \quad \mathbf{J}_d = \begin{bmatrix} \frac{\partial p_x}{\partial d_1} & \frac{\partial p_x}{\partial d_2} & \cdots & \frac{\partial p_x}{\partial d_n} \\ \frac{\partial p_y}{\partial d_1} & \frac{\partial p_y}{\partial d_2} & \cdots & \frac{\partial p_y}{\partial d_n} \\ \frac{\partial p_z}{\partial d_1} & \frac{\partial p_z}{\partial d_2} & \cdots & \frac{\partial p_z}{\partial d_n} \end{bmatrix},$$
$$\mathbf{J}_a = \begin{bmatrix} \frac{\partial p_x}{\partial a_1} & \frac{\partial p_x}{\partial a_2} & \cdots & \frac{\partial p_x}{\partial a_n} \\ \frac{\partial p_y}{\partial a_1} & \frac{\partial p_y}{\partial a_2} & \cdots & \frac{\partial p_y}{\partial a_n} \\ \frac{\partial p_z}{\partial a_1} & \frac{\partial p_z}{\partial a_2} & \cdots & \frac{\partial p_z}{\partial a_n} \end{bmatrix}, \quad \mathbf{J}_\alpha = \begin{bmatrix} \frac{\partial p_x}{\partial \alpha_1} & \frac{\partial p_x}{\partial \alpha_2} & \cdots & \frac{\partial p_x}{\partial \alpha_n} \\ \frac{\partial p_y}{\partial \alpha_1} & \frac{\partial p_x}{\partial \alpha_2} & \cdots & \frac{\partial p_y}{\partial \alpha_n} \\ \frac{\partial p_z}{\partial \alpha_1} & \frac{\partial p_x}{\partial \alpha_2} & \cdots & \frac{\partial p_z}{\partial \alpha_n} \end{bmatrix}. \tag{2.71}$$

The elements of $\mathbf{J}_\theta$ are identical to those that appear in the robot Jacobian relating linear velocity. Thus, the equation that relates end-effector position errors to errors in the kinematic model of the robot is:

$$\begin{pmatrix} \delta p_x \\ \delta p_y \\ \delta p_z \end{pmatrix} = \begin{bmatrix} \\ \mathbf{J}_{3\times(N\times n)} \\ \\ \end{bmatrix} \begin{pmatrix} \delta\theta_1 \\ \delta\theta_2 \\ \vdots \\ \delta\theta_n \\ \delta d_1 \\ \delta d_2 \\ \vdots \\ \delta d_n \\ \delta a_1 \\ \delta a_2 \\ \vdots \\ \delta a_n \\ \delta\alpha_1 \\ \delta\alpha_2 \\ \vdots \\ \delta\alpha_n \end{pmatrix}. \tag{2.72}$$

Notice that if Equation 2.72 is divided by δt, the left-hand side becomes the linear velocities in terms of the three Cartesian directions. The Identification Jacobian remains unchanged and the vector of small parameter displacements becomes the time rate of change of the robot parameters. As only the joint variables typically can change state, the rest of these terms vanish in a differential sense. Because of the assumption that the parameter deviations are small compared to the parameters themselves, we can use this differential relation to solve for the approximate deviations. The resulting equation relates linear velocities of the end-effector to the joint rates and parameter deviations. Three additional rows are required to relate the three linear and angular velocity components of the end-effector to the joint rates.

Chapter 3

Kinematic Calibration

Despite today's current level of technology and precision in manufacturing processes, deviations between the nominal geometry of a final product and actual finished product always exist. In the real world, small discrepancies in a robot's geometry exist due to manufacturing inaccuracies. This fact specifically constitutes the need for calibration. In the off-line programming of robots, they are commanded to various poses to achieve different tasks in an automated procedure. If errors exist in the geometric model of the robot stored in the controller and the real manipulator, they will compound to negatively impact the performance of these tasks.

The *accuracy* of a robot is a measure of the difference between a commanded pose and the robot's actual position and orientation. Accuracy governs the tasks that can be performed with *off-line programmed* configurations. A robot's *repeatability* is the performance index that governs *taught* configurations. Repeatability is a measure of how well the robot can return to a taught configuration. Taught configurations are robot poses where the joint angles, which are as precise as the joint encoders allow, are saved in the controller's memory. Thus, repeatability is the limit to accuracy. A robot can only be calibrated to such a level where the end-effector errors will not exceed the stated repeatability of the

machine.

In the kinematic calibration of a serial robot [36], the objective is to identify the deviations in the nominal kinematic parameters that result in the actual robot parameters. These identified deviations are then used to update the stored parameters resident in the controller to improve the accuracy of the robot, potentially up to the limit of its repeatability, for off-line programming purposes. In this chapter, a conventional kinematic calibration method, based on absolute measurements and utilizing the *Singular Value Decomposition* (SVD), will be discussed and a definition of the RMC will be provided. The differences between the two will be stated along with the benefits of the novel RMC calibration method [1, 5].

3.1 Conventional Calibration Method

In the conventional calibration method, it is required that the measurements of the end-effector be expressed in the base coordinate frame of the robot. With this method, only the position of the end-effector is used as opposed to its orientation, or both. To satisfy this requirement, a position measurement sensor, as seen in Figure 3.1, must be able to recognize the location of the base of the robot and the location of the end-effector in Cartesian coordinates. The coordinates of both of these points are expressed in the sensor's coordinate system, and the calculated difference between them comprises the measurement of the robot's end-effector relative to the base.

In general, the position error for a number of end-effector positions, as computed via the difference in the predicted location through the forward kinematics and the provided measurements, is used with the pseudo-inverse of the Identification Jacobian to estimate the parameter deviations. Assuming that the robot was constructed in an ideal world, the

position of the end-effector expressed in the base coordinate, ${}^{B}\mathbf{p}_{C}$, would be:

$$ {}^{B}\mathbf{p}_{C} = f(\theta, d, a, \alpha), \tag{3.1} $$

and strictly dependent on the nominal robot parameters. Equation 3.1 corresponds to the computation of the forward kinematics with the nominal robot parameters by the controller. The parameters, θ, d, a, and α, are the DH parameters (or MDH parameters), as defined in Chapter 2. The stacked vector Θ consists of all the robot parameters in the same order as in Equation 2.72, so Equation 3.1 can also be stated as:

$$ {}^{B}\mathbf{p}_{C} = f(\Theta). \tag{3.2} $$

As deviations exist between the nominal parameters and the actual parameters, they are represented as $\Delta\theta$, Δd, Δa, and $\Delta\alpha$. The two parameters relating to the interconnection between links, θ and d, are the controlled variables for revolute and prismatic joints,

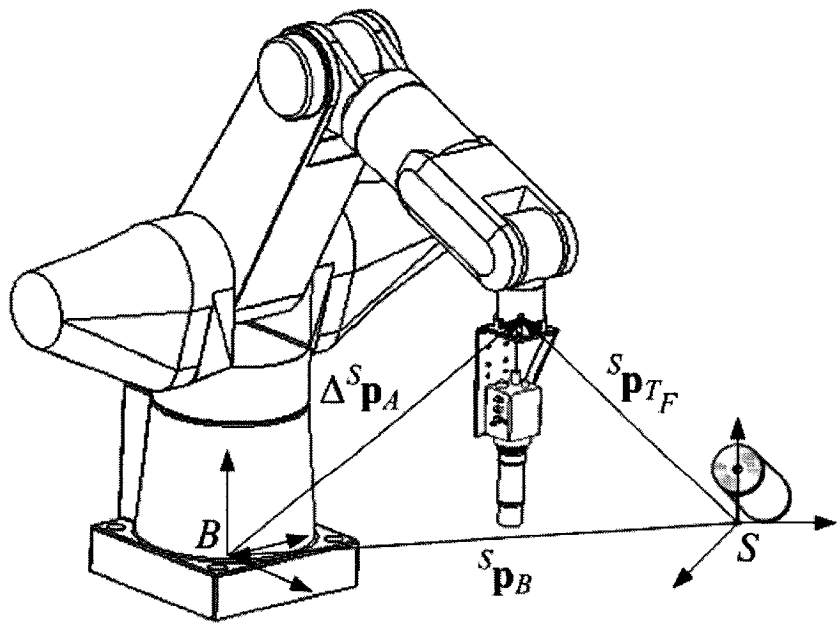

Figure 3.1: Measurements provided by an absolute position measurement system.

respectively. For a revolute joint, the joint angle, θ, is controlled while the link offset, d, is a fixed quantity. The opposite situation exists for prismatic joints. The deviation in the controlled variable can be thought of as bias error, a consistent offset from a desired value. The deviations in the remaining three parameters are fixed displacements, linear and angular, between the nominal and actual case. So, the position of the end-effector, ${}^{B}\mathbf{p}_{A}$, for the actual case, is a function of the nominal robot parameters with the deviations present,

$$ {}^{B}\mathbf{p}_{A} = f(\theta + \Delta\theta, d + \Delta d, a + \Delta a, \alpha + \Delta\alpha). \tag{3.3} $$

The difference between these two positions is the measured error:

$$ \Delta^{B}\mathbf{p}_{E} = {}^{B}\mathbf{p}_{A} - {}^{B}\mathbf{p}_{C}. \tag{3.4} $$

The vector, $\Delta^{B}\mathbf{p}_{E}$, is invariably the same vector despite being expressed in a different coordinate system. However, the components of the vector change when it is described with respect to different coordinate systems. Thus, the components of $\Delta^{B}\mathbf{p}_{E}$ are expressed with respect to frame B. In order to identify the parameter deviations, Equation 2.72 is used provided there are a sufficient number of m measurements. There are four types of kinematic parameters and six degrees-of-freedom, thus 24 parameters to be identified. The compact form is:

$$ (\Delta^{B}\mathbf{p}_{E})_{(3\times m)\times 1} = [\mathbf{J}]_{(3\times m)\times 24}(\Delta\boldsymbol{\Theta})_{24\times 1}, \tag{3.5} $$

where $\Delta\Theta$ is the concatenated vector of the robot parameter deviations.

As the Jacobian is non-square, and thus non-invertible in general, the solution for its pseudo-inverse must be approximated with an appropriate method. The SVD was chosen as it provides additional useful information. The parameter deviations can then be

estimated using:

$$\Delta\Theta = \mathbf{J}^{+}\Delta^{B}\mathbf{p}_{E}, \tag{3.6}$$

where $\mathbf{J}^{+}$ is the pseudo-inverse of $\mathbf{J}$.

This approach was used to generate data for the absolute measurement simulation. Each measurement of the end-effector position provides three equations in the twenty-four parameters. Thus, a suitable number of measurements had to be taken to create an over-determined set of equations. A general rule-of-thumb in this regard is three times the number of parameters to be identified [37].

The absolute simulation served as a means to test program elements and was used extensively in the development of the RMC simulation. The results of the absolute simulation, as well as the different programmed elements, will be discussed in the next chapter. An essential component to the calibration procedure is the SVD. This matrix factorization method will now be described.

3.1.1 Singular Value Decomposition

As noted in the conventional calibration scheme, the pseudo-inverse of the Identification Jacobian is required to identify the parameter deviations. The SVD is a powerful matrix factorization method that can be used in the solution of over-determined systems of linear equations in a least-squares sense [37, 30]. It can be further analyzed to determine which estimates are unreliable and thus states specifically the number of trustworthy estimates that approximately satisfy the equations in a least-squares sense.

In the reduced form of the SVD of an $m \times n$ matrix $\mathbf{J}$, as opposed to the full SVD, three matrices are produced: $\mathbf{U}$, an $m \times n$ column-orthogonal matrix; $\mathbf{S}$, an $n \times n$ diagonal matrix with entries that comprise the singular values; and the transpose of $\mathbf{V}$, an $n \times n$

orthogonal matrix. Any matrix $\mathbf{J}$ can be decomposed:

$$\mathbf{J} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T, \tag{3.7}$$

where $\mathbf{S}$ appears as:

$$\mathbf{S} = \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & s_n \end{bmatrix}, \tag{3.8}$$

and the s_i represent the singular values.

The pseudo-inverse of $\mathbf{J}$ can be obtained by rearranging Equation 3.7. As $\mathbf{V}$ is a column-orthogonal square matrix, its inverse is simply its transpose. $\mathbf{U}$ is also column-orthogonal, but is rectangular. For the $m \times n$ matrix $\mathbf{U}$, with orthonormal columns, then $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, where $\mathbf{I}$ is an $n \times n$ identity matrix. The dot products of the like numbered columns result in a 1 on the diagonal and all other entries, as the columns are mutually orthogonal, are zero. If $\mathbf{S}$ is of full rank its inverse, $\mathbf{S}^{-1}$, is the diagonal matrix with elements that are the reciprocal of those in $\mathbf{S}$. So, the final version of the parameter estimation relation can be obtained from Equation 3.5 by multiplying the left-hand side with $\mathbf{U}^T$, then $\mathbf{S}^{-1}$, and finally $\mathbf{V}$. The result is:

$$\Delta\boldsymbol{\Theta} = \mathbf{V} \begin{bmatrix} \frac{1}{s_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{s_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \frac{1}{s_n} \end{bmatrix} \mathbf{U}^T \Delta^B \mathbf{p}_E. \tag{3.9}$$

One concern of this matrix factorization method is that it may compute a singular value that is close to the numerical precision of the computer. In this case, allowing $\frac{1}{s_i}$,

where the s_i is very small, to continue through the program would corrupt any results, the matrix entry is set to zero. If this action is not taken, the inversion of a singular value of a magnitude of 10^{-15} would produce a solution dominated by round-off error.

When this is done, the influence of the linear combination of free parameters causing the system to be ill-conditioned is removed. A check of the column in $\mathbf{V}$ corresponding to the column in $\mathbf{S}$ for the zeroed singular value reveals which estimated parameters are not to be trusted. The columns of $\mathbf{V}$ corresponding to $\frac{1}{s_i} = 0$ form an orthonormal basis for the nullspace of $\mathbf{J}$. In general, the nullspace for the matrix $\mathbf{A}$ is the set of all solutions where $\mathbf{Ax} = 0$. The number of zeroed singular values indicates the number of untrustworthy parameter estimates which occupy the nullspace.

3.2 Relative Measurement Concept

The RMC is a new approach to measurement acquisition in that it uses relative measurements, which are not expressed with respect to the base frame of the robot. Images of a precision-ruled straight edge are taken by a *Charge-coupled Device* (CCD) camera rigidly mounted to the end-effector of the robot. The CCD camera is focused on the graduated edge of the ruler when acquiring images, as illustrated in Figure 3.2. Through the implementation of digital image processing algorithms, the analysis of a series of images reveals the error in their corresponding robot poses. The images of the ruler are a measure of the actual location of the end-effector but the error is defined relative to a reference pose and not the base coordinate system.

With reference to Figure 3.2, consider a ruler placed at any location in the workspace of a serial robot. The CCD camera is positioned such that the surface of the ruler is within the focal length of the lens and the camera is perpendicular with respect to the ruled surface. Assume that the line of sight for the camera is aligned with the z direction of the

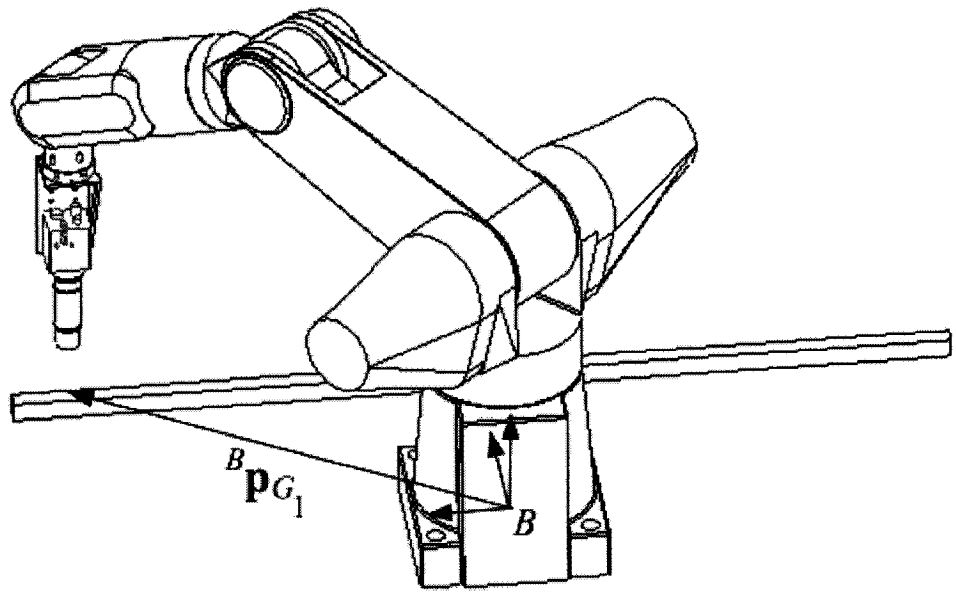

Figure 3.2: The RMC setup.

tool frame, or parallel to this direction. Although this is not the case due to inaccuracies in the mounting bracket, we assume the error is negligibly small.

The edge of the ruler appears as a horizontal line in the image while the graduations are vertical. The coordinates of the intersection of the ruler edge and the first graduation, in the image coordinate system, is used as a reference point for all the other images. The first graduation is situated at a fixed distance from the world coordinate system, as are all the other graduations, but the coordinates of the vector describing this point, ${}^{B}\mathbf{p}_{G_1}$, shown in Figure 3.2, are unknown.

Once the first image has been taken with the end-effector suitably posed, the robot is commanded to move a specified increment along the length of the ruler. The increment is expressed in the tool frame as it is properly aligned with the ruler. With reference to Figure 3.3, the second image is a fixed distance away from the first. In the ideal world, both of these images would be identical. However, due to the presence of kinematic errors, repeatability effects, and thermal expansion effects, shifts in the location of the edge of the ruler and graduations in the image are observed. The ruler edge either shifts upwards

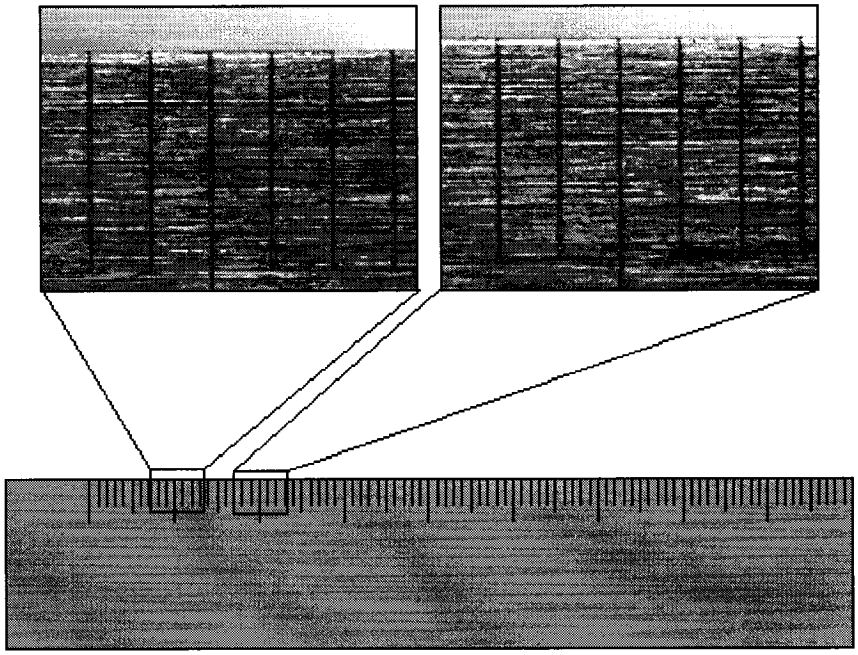

Figure 3.3: Measurements obtained utilizing the RMC.

or downwards while the graduations shift left and right in the image. This means that the end-effector is lower or higher and to the right or to the left of where it should have gone, respectively. Pictured in Figure 3.4 are the first and second images from an actual measurement set where these shifts can be recognized.

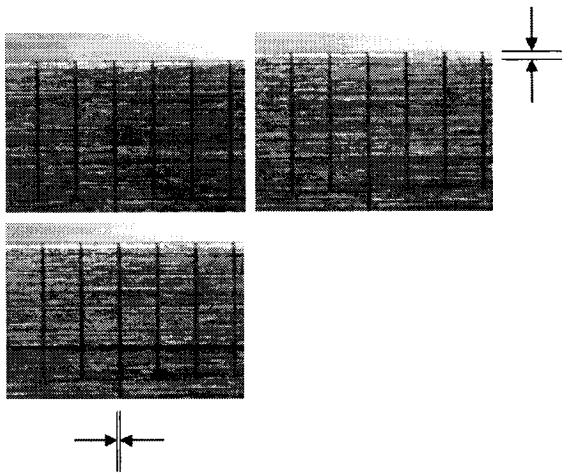

Figure 3.4: Shifts in end-effector position recognized in the comparison of sequential images.

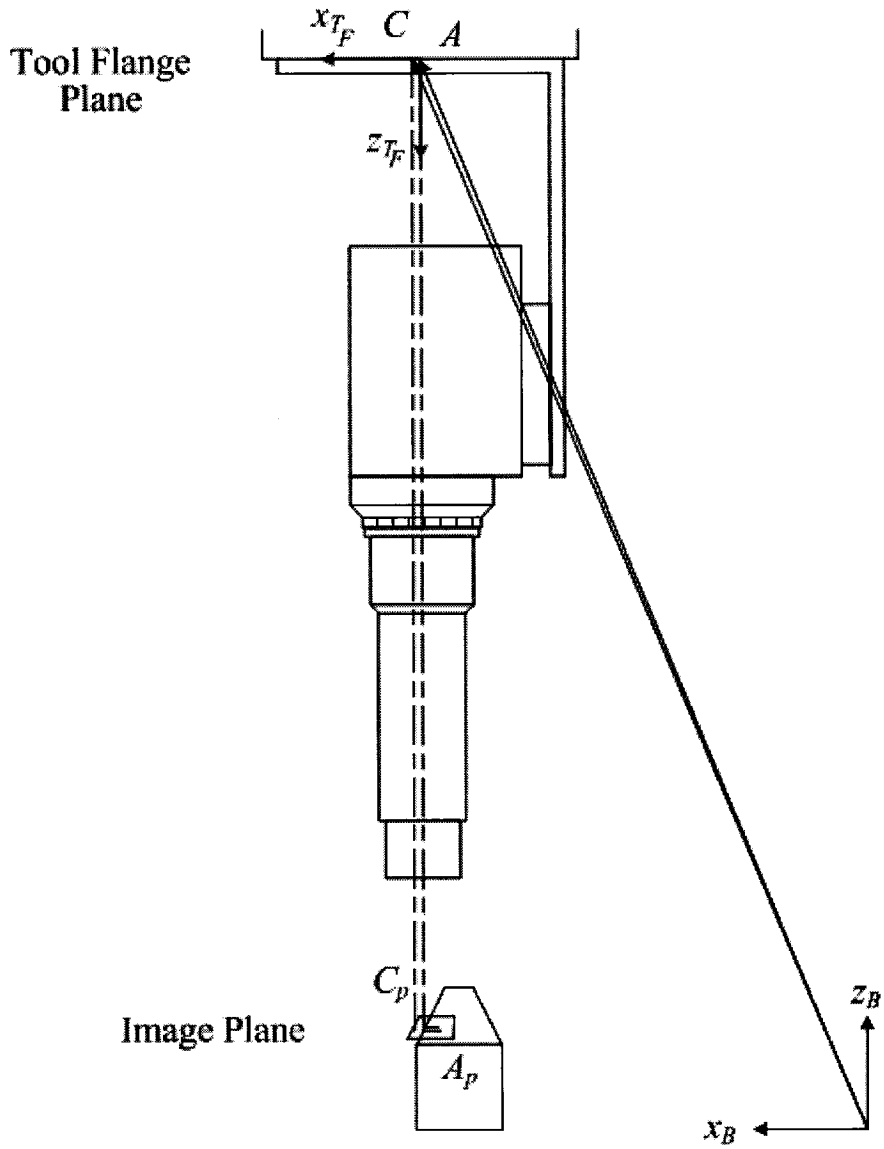

Figure 3.5: Projection of the actual and controller points onto the image plane.

The computation of the forward kinematics using the nominal parameters reveals the *controller points*, $C = \{C_1, C_2, ..., C_m\}$, whereas the *actual points*, $A = \{A_1, A_2, ..., A_m\}$, are determined with the inclusion of the parameter deviations. These two sets of points terminate at the tool flange centre-point for the predicted and actual cases. They are projected onto to the image plane to become $C_p = \{C_{p_1}, C_{p_2}, ..., C_{p_m}\}$ and $A_p = \{A_{p_1}, A_{p_2}, ..., A_{p_m}\}$. The set of points, $G = \{G_1, G_2, ..., G_m\}$, represent the intersections between the ruler edge

and the graduations. The projection of A and C onto the image plane can be seen in Figure 3.5.

Ideally, the projected controller points, C_p, should form a line parallel to the graduation points. Hence, their image coordinates vary from image to image, but remain a fixed distance from the ruler edge and the graduation. However, the projected actual points, A_p, should have the same image coordinates in every image. This is because the camera is aligned with the z direction of the tool frame and the two are assumed to be parallel in space.

Considering the first two images, the relative error, ${}^{T_F}\mathbf{p}_R$, is the difference between the image coordinates of G_1, in the reference image, and the image coordinates of G_2 in the second image. As the image plane is assumed parallel to the tool flange plane, this quantity is expressed with respect to the tool frame, as opposed to the base frame. The relative error, $\Delta^{T_F}\mathbf{p}_R$, is the difference between the displacement of the end-effector that should of occurred, from A_{p_1} to a point that is a distance of $\overline{G_1G_2}$ away, and where it actually went, from A_{p_1} to A_{p_2}. This can be seen in the comparison of the first two images and the ruled surface in Figure 3.6.

While the placement of the ruler in the workspace is arbitrary, some way in which to transform the measurement points into the base coordinate system is necessary. This is achieved by rotating the measurements with the Euler angles of the first pose. These Euler angles do contain error, but the first pose is a taught pose which establishes the tool frame necessary to perform the measurement acquisition. The only other way to obtain this information would be to measure the orientation with an external apparatus, and to an order of magnitude better than the installed equipment. For the purposes of the RMC calibration system, the Euler angles of the first pose, which are accurate up to the repeatability of the robot, must be accepted. Thus, $\Delta^{T_F}\mathbf{p}_R$ can be transformed into $\Delta^{B}\mathbf{p}_R$ with the transformation ${}^{B}\mathbf{T}_{T_F}$. As stated before, the vector itself remains invariant but

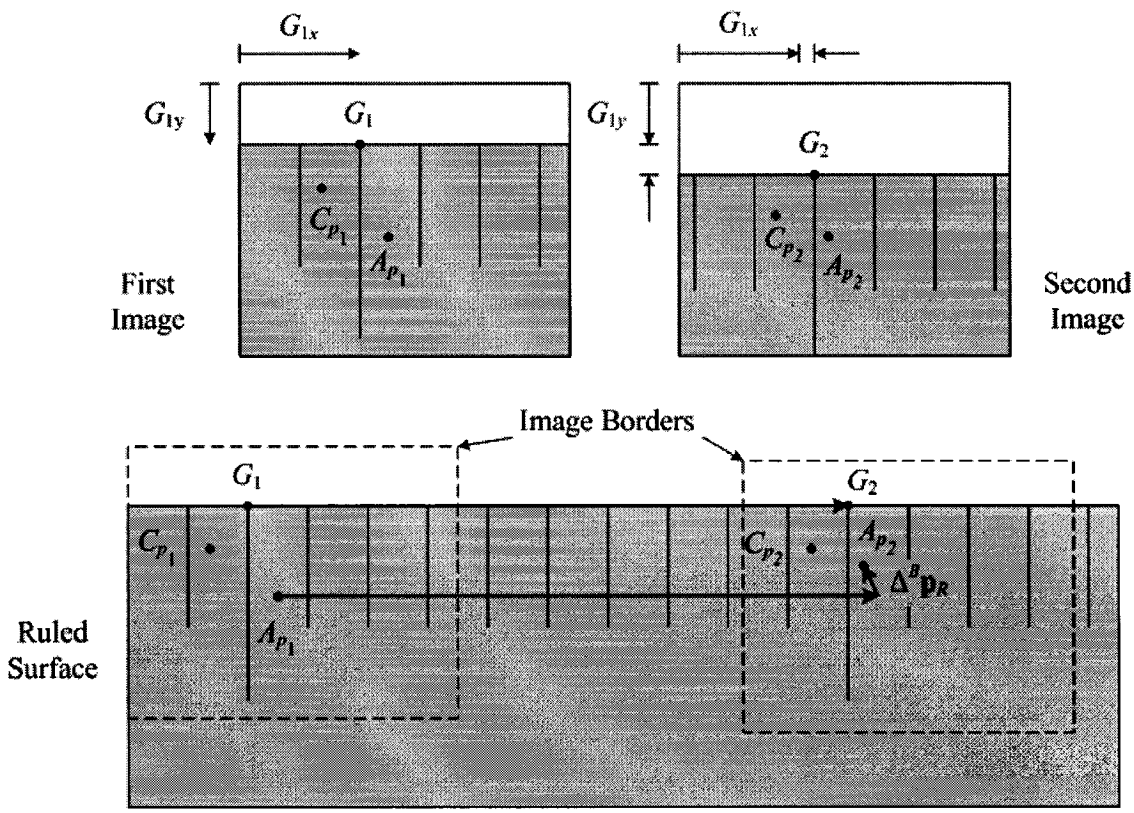

Figure 3.6: Relative error obtained in comparison of two images.

its components change to reflect the description in another coordinate system.

To acquire a series of m measurements, a corresponding number of images would be taken for m increments along the length of the ruler. The error is computed through the comparison of all the images with the reference position, G_1, as recorded in the first image and does not correspond to the total error of the end-effector's position, $\Delta^B\mathbf{p}_E$, but rather a significant component of the error, $\Delta^B\mathbf{p}_R$. The total error, $\Delta^B\mathbf{p}_E$, can be computed with direct measurements of the end-effector position, ${}^B\mathbf{p}_A$, and is expressed with respect to the base coordinate system in the conventional calibration scheme. With this scheme two sets of points are available. A stacked error vector, of size $(3 \times m) \times 1$, would be substituted into Equation 3.6 to estimate the parameter deviations.

With the RMC, the relative error vector is substituted. Only one set of points is

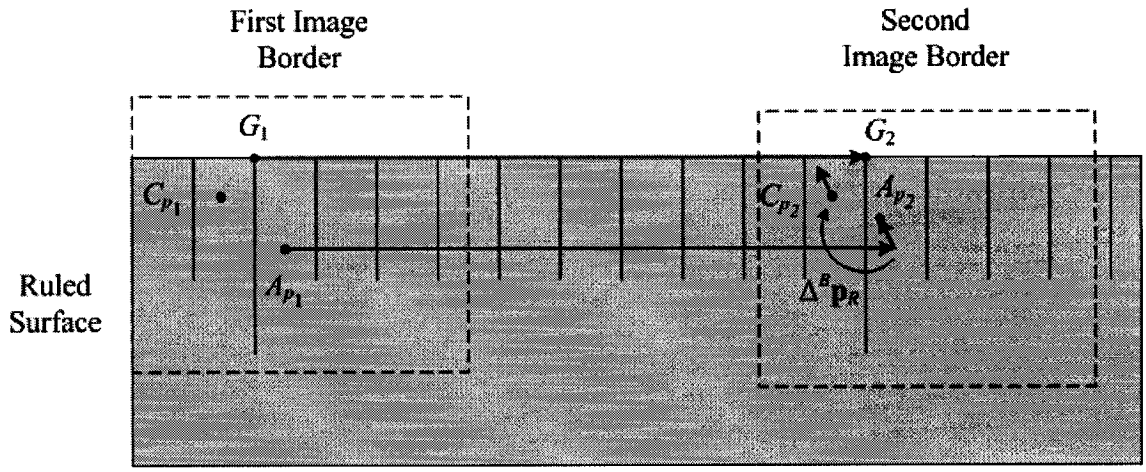

Figure 3.7: Relative error substituted as total error.

available, the controller points. The relative error is the output of the image analysis. The locations of the actual points are measured relative to the first one but their Cartesian coordinates, in the base frame, are unknown. Essentially, by performing the substitution in the first iteration, it is being assumed that the relative error acts at the controller points, and is the total error. This can be seen in Figure 3.7.

Consider again the three sets of points, C_p, A_p, and G, of Figure 3.6, along with the method of taking measurements along the length of the ruler. If the robot was perfectly constructed and the ruler perfectly aligned, these three sets of points would be identical. Introduce the parameter deviations and the situation of Figure 3.6 results. The total error is considered to consist of two components: the relative error, $\Delta^B\mathbf{p}_R$, which changes, and a component due to the offset between the predicted and actual position of the tool flange for the first pose, $\Delta^B\mathbf{p}_D$.

Ideally, $\Delta^B\mathbf{p}_D$ does not change from pose to pose. In reality, as the robot cannot be commanded to move perfectly along the length of the ruler $\Delta\mathbf{p}_D$ is not a constant vector. The lines $\overline{C_{p_1}C_{p_m}}$ and $\overline{G_1G_m}$ are not parallel as assumed. The effect of this imperfect situation can be made quite small. The length of the ruler used in the current setup is 80 cm, and its edge at the start and end locations can be placed to within a couple of pixels

of each other when comparing their associated images. Thus, the small error generated by this situation can be either ignored or some compensation could be performed as the errors are linear nature in the image plane. The parameter estimation is:

$$\Delta\Theta = \mathbf{J}^{+}(\Delta^{B}\mathbf{p}_{R} + \Delta^{B}\mathbf{p}_{D}). \tag{3.10}$$

The pseudo-inverse of the Identification Jacobian is a linear transformation. So, Equation 3.10 can be re-written as:

$$\Delta\Theta = \mathbf{J}^{+}\Delta^{B}\mathbf{p}_{R} + \mathbf{J}^{+}\Delta^{B}\mathbf{p}_{D}. \tag{3.11}$$

As $\Delta^{B}\mathbf{p}_{D} \rightarrow 0$, the latter component of the equation, $\mathbf{J}^{+}\Delta^{B}\mathbf{p}_{D}$, also tends towards zero. However, if $\Delta^{B}\mathbf{p}_{D}$ is of the same magnitude as $\Delta^{B}\mathbf{p}_{R}$, some compensation of the effect must be performed. So, consider another experiment in which the physical placement of the ruler is altered by a small distance, in the range of 1-100 μm, in one coordinate direction. It would be expected that the error vector generated in this second experiment would be similar to that of the first, despite the complexities of a six degree-of-freedom serial robot. With this assumption,

$$\Delta\Theta = \mathbf{J}^{+}\Delta^{B}\mathbf{p}_{R}, \tag{3.12}$$

can be performed once to estimate the parameter deviations. However, in order to perform further iterations to refine the estimate, some means to update the error vector must be established.

In the conventional calibration scheme, the absolute position of the end-effector, ${}^{B}\mathbf{p}_{A}$, for the set of m measurements, is directly measured. With these measurements, a corresponding set of predicted end-effector positions is computed via the forward kinematics

with the nominal robot parameters, Θ. The error is then computed as their difference. For the next iteration in this scheme, the estimates from the first iteration, $\Delta\Theta$, are used to improve the accuracy of the nominal parameters, Θ. The forward kinematics are then re-computed and compared with the available measurements to generate a new error vector, which should be smaller in magnitude than the first. This new error vector is then used to further refine the estimates and this process continues until a convergence criterion is satisfied.

With the RMC method, all that is available is the error vector, $\Delta^B\mathbf{p}_R$, which cannot be used to locate the actual points, A, relative to the base frame. However, after one iteration of Equation 3.12, estimates for the improvement of the nominal parameters, Θ, become available. Computing the forward kinematics for the first pose with the improved nominal parameters, $\Theta + \Delta\Theta$, provides an estimate of the first actual point, the updated controller point C_{u_1}, seen in Figure 3.8. As the relative measurements were defined relative to A_{p_1}, the procedure can then continue by mimicking the conventional calibration scheme by expressing them relative to C_{u_1}. The difference between the original projected controller points, C_p, and the updated controller points, C_u, is close to the actual difference between C_p and A_p, and thus C and A. Iteration would continue as before until a convergence

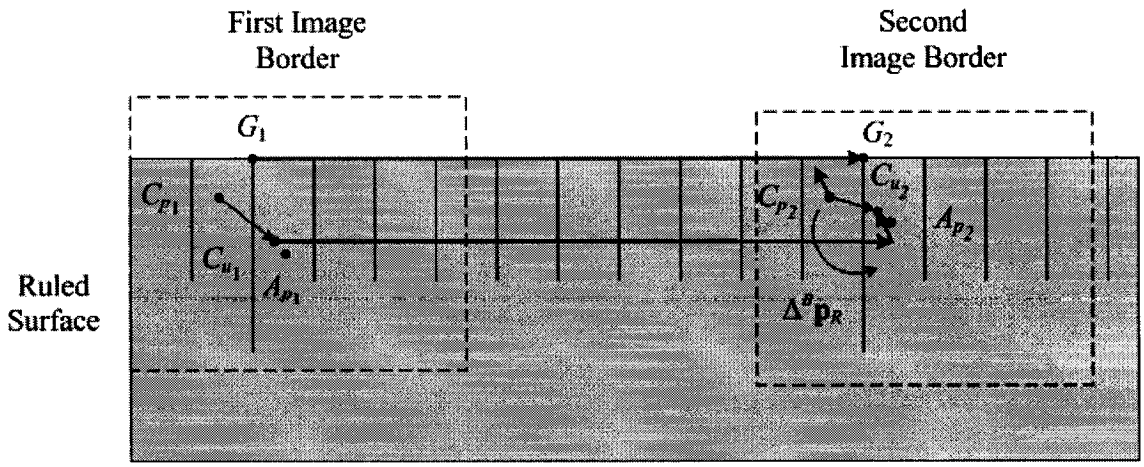

Figure 3.8: Relative error applied at estimates of the actual points.

criterion is satisfied, then the procedure is exited.

The benefits of the RMC method are numerous. First, the cost of such a system is relatively inexpensive. The current system consists of a precision-ruled straight edge, a Pulnix CCD camera, a 1× Rodenstock lens, a ring-light, a National Instruments IMAQ PCI-1409 data acquisition card, a dedicated computer, and a custom-built measurement head. The total cost of this system is approximately $6,000.00 CAD. This cost is at least an order of magnitude lower than the cost of an absolute position measurement device. This system is easily transportable and easily interfaced with serial robots. All that is required is to attach the measurement head to the tool flange, place the ruler, and generate a simple program to perform the measurement. The calibration procedure could be performed on-site, with no need to remove the robot from its workcell [3].

Chapter 4

Simulation Results

Simulation of any physical process is a beneficial tool that can lead to a more in-depth understanding and optimization of laboratory experiments. A simulation can be simple, with only a few modelled components, or complex, where the behaviour of the system is modelled by many parameters. Many different program components are needed, or included to provide additional options, in the construction of a simulation of a kinematic calibration procedure. Functions to compute robot kinematics, such as the forward and inverse kinematics, as well as procedural elements, such as pose generation and measurement noise, are included in the simulations created for this project.

The objective of the simulations is to successfully identify the specified robot parameter deviations. These deviations represent the error between the nominal and actual geometry of the robot. The error of a particular robot pose, for the simulations and the actual experiment, is the difference between the measured position of the end-effector and the predicted position. This error is related to the parameter deviations through the formulation of the Identification Jacobian matrix, $\mathbf{J}$. A solution for the linear system of equations, in n parameter deviation unknowns, is obtained in a least-squares sense. The SVD is used to approximate the pseudo-inverse of the Identification Jacobian. In using this formula-

tion, the error in the end-effector position is assumed to be small compared to the robot itself.

4.1 Applicability of the Jacobian Matrix

To confirm that the error in end-effector position allows the application of the Jacobian, a program was devised to compare the prediction of the motion of the end-effector with the Jacobian and a time-step approach. The Jacobian relates linear and angular velocities of the end-effector to the joint rates. To move the end-effector in a straight line in the workspace requires that the six motors provide varying angular speeds so that the links move relative to each other at the proper rates. So, a comparison between the kinematics approach, assuming the links to be rigid bodies, and the Jacobian approach had to be made. The angular velocities of the first three joints, which determine the end-effector position for wrist-partitioned serial robots, are stipulated such that the position of the end-effector, compared to the last time step, is displaced by an amount comparable to the error expected with the known parameter deviations. Essentially, a relative-motion analysis [38] is performed at each time-step. Comparing the two methods in this fashion yields an assessment of valid use of the Jacobian.

The error in end-effector positioning is expected to be, at best, within the repeatability of the robot. The repeatability for the KUKA KR 15/2 and Thermo CRS A465 is ± 100 μm and ± 50 μm, respectively. If the links were constructed with a CNC milling machine, with an accuracy of ± 2.5 μm, the expected errors in the link lengths would be in the range of ± 5.0 μm. To assess the application of the Jacobian, the end-effector of the KUKA KR 15/2 should be moved such that the end-effector centre-point is moved a distance of 0.2 mm relative to its previous position. This corresponds to the positional error encountered in the first pose with the specified parameter deviations, on the order of microns. The angular

velocities of the joints and the time increment would simply be specified to accommodate this relative distance.

A program was devised to compute the velocity of the end-effector through the use of the Jacobian and through positional level kinematics. The geometry of the KUKA KR 15/2 was specified in the program. Only the first joint was given an angular velocity while the other five joint angles remained constant at arbitrarily assigned values. A circle was traced in a plane perpendicular to the z direction of the base frame by the end-effector around the first joint axis with these stipulations. The linear velocity of the end-effector was computed at each time-step with the nominal kinematic parameters in the Jacobian method. For the second method, the forward kinematics were computed for each time-step. The average velocity components were then calculated. For a distance of approximately 0.2 μm, the xy velocity components were accurate to within an average of 0.016 %. The percent difference in the norms of the velocity vectors, as computed by both methods, was negligible. In assigning an angular velocity for each of the joints, still resulting in the desired distance, the error in the vector norms is significant, approximately 5 %. However, deviations of up to 1 mm in magnitude were specified in the absolute simulation and they were successfully identified.

For the experiment with the Thermo CRS A465, with the expected errors and parameter deviations, the application of the Identification Jacobian seems merited. However, as was found in the first experiment, the errors encountered reached a maximum of approximately 0.55 mm. In the experiment with the KUKA KR 15/2, the maximum error was approximately 0.8 mm.

4.2 Absolute Simulation

The conventional calibration scheme was followed in the absolute simulation. The position of the end-effector was measured and used in the calibration procedure but not its orientation. A general flow diagram for the simulation is shown in Figure 4.1 and a summary of the program code is provided in Appendix C.

In the first stage of the simulation, the robot parameters are initialized. A selection as to which model convention is made at the beginning of the program. The geometry of the KUKA KR 15/2 is currently specified. If the calibration procedure was to be simulated with other robots, all that need be done is alter the robot parameter values such that they reflect the robot's geometry. The simulation then attends to the following tasks: pose generation, measurement acquisition, optional noise addition, error vector computation, assembly of the Identification Jacobian matrix, and kinematic parameter deviation identification.

4.2.1 Pose Generation

From each measurement of the end-effector, as only the positional data is used, three kinematic equations are generated per robot pose, giving a calibration index of $C = 3$ [27]. As there are at least 24 system parameters, a minimum of 8 poses are required to ensure a unique solution. There are two ways in which robot poses can be supplied to the simulation. They can either be input from a data file or generated from random increments to the joint variables starting from a specified pose. The entire joint angle set is then stored so it can be accessed throughout the program. A visualization of the KUKA KR 15/2 experimental joint angle data as well as a randomly generated case are pictured in Figure 4.2.

Currently, joint angles are assumed to be ideal with no modelled error. In the experimental case, the robot can only achieve discretized joint angles. The joint encoders have

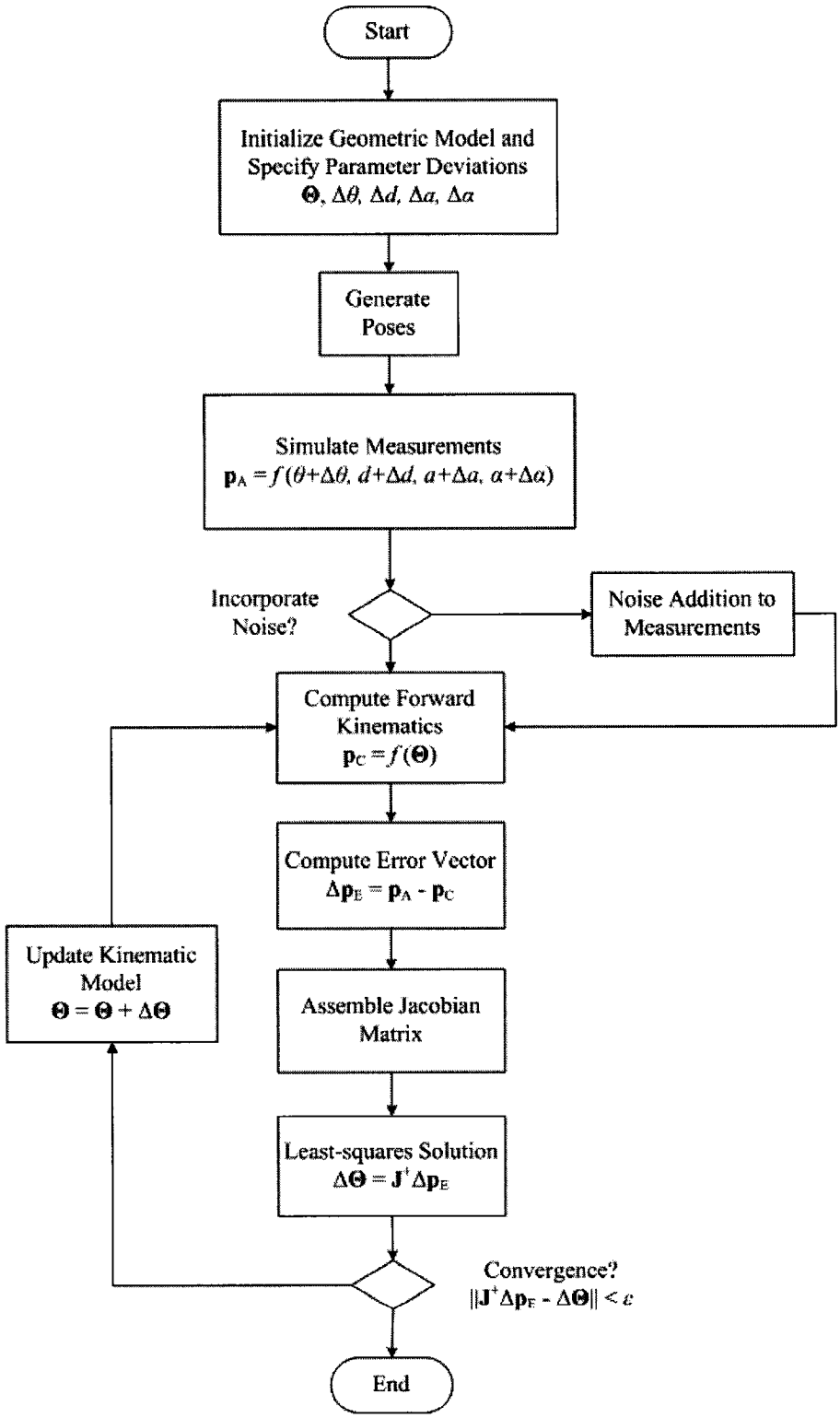

Figure 4.1: The absolute simulation flow diagram.

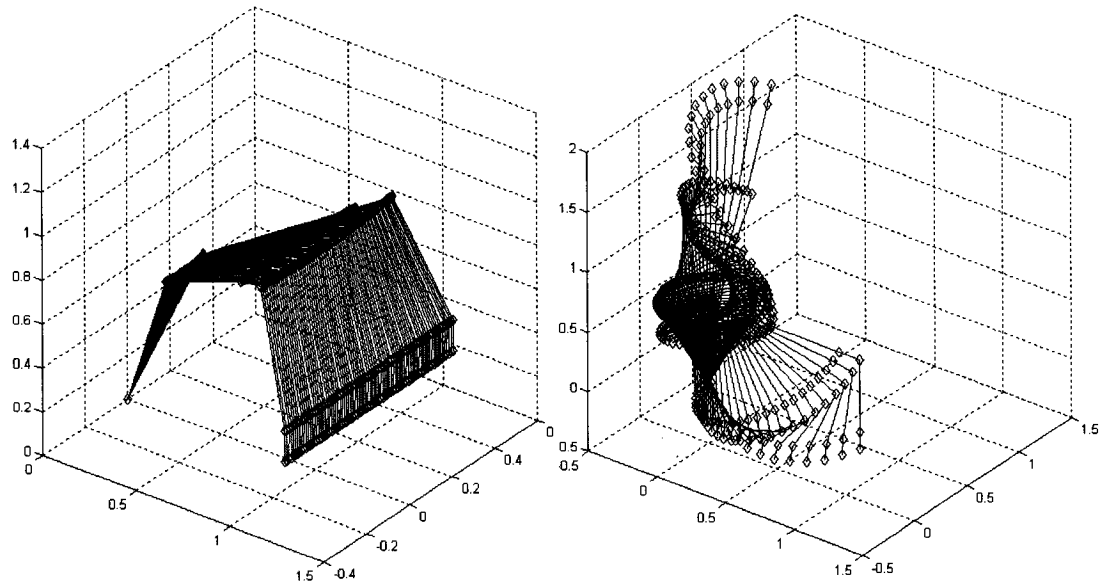

Figure 4.2: Visualization of the KUKA KR 15/2 experiment (left side) and a randomly-generated case (right side).

a finite resolution and thus each joint is commanded to the most suitable motor position for a particular pose. The resulting error in the pose is not due to the kinematic errors but rather the theoretically predicted joint angles and the actual ones. Compensation for this kind of error can be implemented into the experimental calibration procedure.

4.2.2 Measurement Acquisition and Noise

Simulated measurements of the end-effector are provided by computing the forward kinematics of the robot, but with the parameter deviations included. The DH transformation matrices, computed through use of a general function that accepts the four DH parameters as inputs and outputs the 4×4 homogeneous transformation matrix, experience slight changes in position and orientation due to these deviations. When multiplied together, the pose of the end-effector is known, but only the positional data is required. The coordinates of the tool flange centre-point, which are the actual points, $\mathbf{p}_A$, are recorded for later use.

At this point, some degree of noise can be incorporated into the measurements. This

is achieved by adding a random value of a specified standard deviation to each of the position components. The accuracy of the measurement device can be included through this option. Also, multiple sets of measurements with random noise can be simulated such that the average of the data could be used as the measurements. So, the effect of measurement noise can be ascertained by varying its magnitude and proceeding with the calibration. Also, compensation of the noise afflicting the measurements can be evaluated through a statistical approach by taking sets of measurements.

4.2.3 Jacobian Elements

The Identification Jacobian matrix contains elements that are derived from the kinematic equations describing the position of the end-effector with respect to the base frame. There are three vector components to position, thus three equations. An analytical expression for the derivative, with respect to each of the kinematic parameters, was obtained with Maple. These expressions are then converted into program code that can be interpreted by Matlab. Each element is kept as a separate function so that an arbitrary Identification Jacobian, dependent on any number of kinematic variables associated with the parameterization and derived with respect to one, two, or three basis directions, could be constructed.

The kinematic equations relating position of the end-effector to the base frame are dependent on the parameterization employed to represent the geometry of the robot. Each equation involves essentially all of the robot parameters. The position of the wrist-centre is dependent on only the parameters associated with the first three joints, whereas the tool flange centre-point is dependent on them all. With each convention, different elementary transformations are used and in a different order. There are three different parameterizations currently incorporated into the simulation: the DH model [31], the MDH model [32], and a DH model altered to cope with the nearly-parallel axes of the second and third joints [39]. The Identification Jacobian constructed with a parameterization is only valid

for that particular model.

4.2.4 Parameter Identification and Convergence

To proceed with the calibration, the stacked error vector must be assembled from the component errors of each pose. The measurements of the end-effector, $\mathbf{p}_A$, have already been computed. The forward kinematics of the robot are computed once again, but with only the nominal geometry. This gives the computed points, $\mathbf{p}_C$. For each pose, the difference in position, $\Delta\mathbf{p}_E = \mathbf{p}_A - \mathbf{p}_C$, is then calculated.

The robot parameters are estimated in a least-squares sense and then iteratively refined by adding the estimated deviations, $\Delta\Theta$, to the nominal parameters, Θ, until some convergence criterion is met. The robot parameter deviations are assumed to be small compared to the link lengths. The estimates of the deviations are stored as corrections to be added to the nominal parameters and refined through further iteration. The corrections vector was initialized as a zero vector and used in the computation of the forward kinematics with the nominal parameters. In the first iteration, there is no effect due to the corrections.

The residual of the linear least-squares approximation is a measure of convergence. As the procedure passes through each iteration of estimating the parameter deviations, $\Delta\Theta$, and then updating the correction vector, the estimates for each iteration become smaller in magnitude. Eventually, the corrections to the parameters become so small that the numeric precision of the computer is reached in computing the residual. At this point, the simulation is complete, the parameter deviations are assumed to be identified, and then the outputs of the program are displayed. The convergence criterion is given by:

$$||\mathbf{J}^{+}\Delta\mathbf{p}_E - \Delta\boldsymbol{\Theta}|| < \varepsilon. \tag{4.1}$$

4.2.5 Simulation Results

With respect to the DH parameterization, as it was the most successful, the following observations can be reported. There were difficulties in identifying d_2, d_3, and α_6. Joints 2 and 3 are nearly-parallel and this is the cause for the difficulties with d_2 and d_3. To explain this result, closer inspection of the expressions for their respective Identification Jacobian elements was required. Due to the geometry of the KUKA KR 15/2, these two elements were practically the same. In the nominal case they are identical as the expression for d_3 simplifies to the expression for d_2. The expressions for the Identification Jacobian elements for d_2 and d_3, with respect to the x, y, and z directions are:

$$\mathbf{J}_{d_2} = \begin{pmatrix} \sin\theta_1 \sin\alpha_1 \\ -\cos\theta_1 \sin\alpha_1 \\ \cos\alpha_1 \end{pmatrix}, \tag{4.2}$$

$$\mathbf{J}_{d_3} = \begin{pmatrix} \cos\theta_1 \sin\theta_2 \sin\alpha_2 + \sin\theta_1 \sin\alpha_1 \cos\alpha_2 + \sin\theta_1 \cos\alpha_1 \cos\theta_2 \sin\alpha_2 \\ \sin\theta_1 \sin\theta_2 \sin\alpha_2 - \cos\theta_1 \sin\alpha_1 \cos\alpha_2 - \cos\theta_1 \cos\alpha_1 \cos\theta_2 \sin\alpha_2 \\ -\sin\alpha_1 \cos\theta_2 \sin\alpha_2 + \cos\alpha_1 \cos\alpha_2 \end{pmatrix}. \tag{4.3}$$

As α_2 is nominally 0, the elements for d_3 simplify to those of d_2. This creates a linear dependency in their respective columns in the Identification Jacobian. These two parameters can only be successfully identified when there is sufficient error present to eliminate the dependency. It was discovered through trial and error that to be suitably identified, an error magnitude of approximately 50 μrad for α_2 had to exist. From absolute zero, the two tend towards their proper values as the error is increased.

The last joint offset, α_6, could not be identified using any of the parameter sets. Due to the nature of the general transformation matrices, α_6 did not appear in the translational component and thus the partial derivatives were zero with respect to these entries. The

column in the Identification Jacobian matrix is therefore a zero vector. Also, the rotation caused by α_6 is the last elementary motion of the chain for the entire robot. The physical effect of this rotation is simply not measured, as only positional data was considered, thus it cannot be identified.

One of the objectives of the calibration system was to avoid hand-to-sensor calibration. As the measurement head mounted on the tool flange can be considered rigid, and its orientation is assumed constant at each pose, the measurements provided by the camera are a direct measure of the tool-flange centre point. Again, this demonstrates that α_6 cannot be identified in this calibration procedure.

For a set of parameter deviations, where they were arbitrarily chosen between 0-100 μm and 0-100 μrad, the identified parameter deviations are listed in Table 4.1. As no noise was introduced into the system of equations, the minimum number of poses was specified. A unique solution was then possible. Notice that α_2 was less than the established value, 50 μm, to identify the link offsets for joints 2 and 3.

Some of the identified parameters converged to their respective true values, however, a small error existed for the others. The sum of the identified d_2 and d_3 equalled the sum of the actual parameter deviations. To eliminate the error in the estimates for d_2 and d_3, the error in α_2 was increased to 50 μm and more measurements were taken to yield an over-determined system. With these input modifications, all observable parameters were identified. The error in d_2 and d_3 was reduced to approximately 0.6 %, whereas the others showed negligible error.

From the $\mathbf{S}$ matrix of the first iteration, 5 singular values are eliminated. The columns in $\mathbf{V}$ corresponding with these singular values are a solution for the null space. These parameters are the least trustworthy. The entry in $\mathbf{V}$ possessing the highest magnitude reveals the parameter having the most influence. The implicated parameters deviations that correspond to these eliminated singular vales are: α_6, d_2 and d_3 (they are linked),

Table 4.1: Results of the absolute simulation with 8 pose measurements.

Parameter	Specified Deviations (m, rad $\times 10^{-6}$)	Identified Deviations (m, rad $\times 10^{-6}$)
θ_1	16.0000	16.0000
θ_2	34.0000	34.0860
θ_3	-56.0000	-56.0860
θ_4	-27.0000	-27.0000
θ_5	22.0000	21.9679
θ_6	13.0000	12.9733
d_1	38.0000	38.0000
d_2	-14.0000	-7005.1887
d_3	-53.0000	6938.1886
d_4	61.0000	61.0000
d_5	-30.0000	-29.9986
d_6	24.0000	24.0000
a_1	-17.0000	-17.0000
a_2	89.0000	89.0000
a_3	64.0000	64.0001
a_4	-45.0000	-45.0000
a_5	37.0000	37.0004
a_6	22.0000	22.0000
α_1	-11.0000	-11.0000
α_2	8.0000	8.0000
α_3	19.0000	19.0000
α_4	21.0000	21.0000
α_5	-15.0000	-14.9902
α_6	14.0000	0.0000

θ_6, θ_5, and α_5. As these parameters were untrustworthy, they were removed from from the analysis and another attempt was made. However, with them removed the simulation performed poorly. The error residual grew in magnitude with each subsequent iteration.

So, as only α_6 was impossible to determine it was the only one removed permanently from further analysis. The inclusion of this α_6 caused the Identification Jacobian to be ill-conditioned. Its corresponding column was a zero vector. Also, the singular value corresponding to this parameter was consistently eliminated anyways. After its removal,

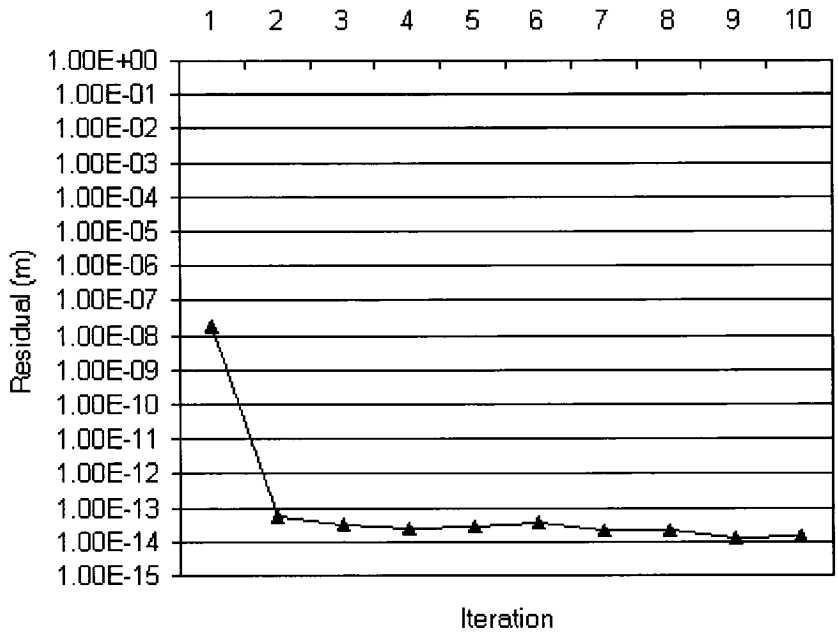

Figure 4.3: Absolute simulation residual *vs* iteration.

the absolute simulation performed equally as well as before, but the issue with d_2 and d_3 was less prominent. Less error in α_2 was required for the successful identification of these deviations.

For the next run, the parameter deviations were increased to the 0-100 mm and 0-100 mrad. As the parameter deviations are assumed to be small, the purpose of this test was to determine how large the errors could be and still yield successful results. At this level the procedure still successfully identified the parameter deviations. Anything larger and the solution would not converge. So, excluding large errors, the calibration scheme performs well. Errors in this range are also highly unlikely due to modern construction techniques and equipment.

In terms of convergence the absolute simulation, with the same parameter deviations listed in Table 4.1, was able to reduce the residual to approximately 10^{-14} in 2 iterations. This can be seen in the logarithmic plot of Figure 4.3. This was after the removal of α_6 from the analysis.

Unfortunately, the addition of noise had a profound effect. Using the parameter deviations of Table 4.1 and increasing the magnitude of the error incrementally, the effect of noise, following a normal distribution, could be determined. The magnitude of the error was dependent on a specified standard deviation, σ, with a mean, μ, of zero about each of the true component values. Given that the Thermo CRS A465 has a repeatability of ± 50 μm, and it is a highly complex mechanism, a measurement device capable of calibrating it would need to have an accuracy of at least one order of magnitude better. So, starting with no error and then increasing it up to this level would determine whether or not the calibration procedure would succeed.

With errors on the order of $\pm 1 - 10$ μm, about half of the estimated parameter deviations could be considered successful. Many of the other estimates were quite large relative to the successful cases. Most noteworthy was that d_2 and d_3 were identified at approximately $\pm$ 1207 m, and their sum still equalled the sum of the actual deviations, within 10 %. When checking for convergence, the value obtained was always of the same order as the specified noise. Thus, the convergence criterion, set at the numerical precision of the machine, was never satisfied with the incorporation of noise and rather a limit for the iterations in the program was reached.

Normally distributed random noise can be combatted by taking more measurements. This tactic was implemented in the simulation. However, despite specifying many sets of measurements, greater than 100, the accuracy of the measurement device remains quite influential. So, for the experimental setup, a more accurate measurement is desired.

The final items to be considered were the included basis directions for the error vector and the straight-line path of the measurements. Currently, for the A465 experimental setup, two directions are considered. Errors in two directions relative to the tool flange coordinate system are acquired through the digital processing of the images. The third direction will be measured when the system is expanded to include a laser displacement

sensor. The current simulation allows the specification of one, two, or all three, basis directions. Without measurement noise, and with the appropriate number of poses, the parameter deviations were still successfully identified. Thus, using only two basis directions has no foreseeable effect on the outcome of the calibration as long as sufficient measurements are taken. However, the straight-line path used for image acquisition does have an effect related to the directions considered.

Using the joint angles from the experiment with the KUKA KR 15/2, a straight-line path could be approximated. It is not exactly a straight-line path due to the resolution of the joint encoders, but close enough to one. It was also aligned with the y direction of the robot. Without measurement noise, the only difficulty was with d_2 and d_3. One singular value was eliminated and indicated that these two parameters were untrustworthy. Upon the addition of noise, in the range of 1-10 μm as before, not one parameter could be identified. All the estimates were excessively large. This has bearing on the experimental results and will be discussed in Chapter 5. Due to this result, it might be advantageous to orient the ruler such that is not aligned with any of the base frame directions.

As the DH parameterization produced the best results, this representational scheme was chosen for the relative simulation. The MDH convention requires two additional parameters to model the same kinematic structure with the DH. With the DH form altered to compensate for the nearly-parallel axes of joints 2 and 3, the added parameter, β_3, could never be successfully identified, thus negating the model's purpose. Other than the selection of an appropriate parameterization, some means of computing the inverse kinematics was required for the construction of the new simulation. Other than this additional requirement, the same processes and functions could be incorporated into the relative simulation. The purpose of the relative simulation was to incorporate the concept of the RMC into a calibration procedure. Issues that would translate to and affect the experimental calibration procedure were considered to be of key importance.

4.3 Relative Simulation

The relative simulation was created to determine whether or not the RMC could be used for measurement acquisition in a kinematic calibration procedure. Essentially, relative measurements were being substituted for absolute measurements. The substantial difference between the absolute simulation is that for the RMC the measurements must take place along a length standard. Thus, the end-effector must follow a straight-line path. In order to achieve these poses, the inverse kinematic problem for the robot must be solved. The remaining components of the program, attending to the forward kinematics, Identification Jacobian assembly and kinematic parameter deviation identification, are the same as those in the absolute simulation. A summary of the simulation code can be seen in Appendix D.

4.3.1 Pose Generation

To simulate the RMC, it was required that the end-effector traverse the length of a precision-ruled straight edge. In the pose generation stage, arbitrary increments could no longer be added to an initial joint angle set as specific poses were now required. In the RMC, the end-effector must maintain its orientation while its position is altered. Increments in its position are specified in the program and correspond to moving to each of the graduations on the ruler. The 3×3 rotation matrix and the 3×1 position vector are therefore known for each measurement pose. It is simply a matter of extracting the joint angles that constitute the pose from the known matrix.

For the MDH parameters, Pieper's solution [34] to the inverse kinematic solution, presented in Chapter 2, was utilized to obtain the joint angles. This solution was initially used until the closed-form solution for the inverse kinematics using the DH convention was developed. With the closed-form solution, essentially six equations, corresponding

to each joint angle, are amalgamated into one function call. Given the necessary inputs, the rotation elements, the translation elements, and a reference joint angle set, the joint angles for a pose can be computed. As the absolute distance between poses in the RMC are small, simple comparisons between the last pose and the current pose are employed to determine the correct joint variables. As with Pieper's approach, multiple solutions exist.

4.3.2 Relative Measurement Acquisition

A relative measurement is simulated by computing the actual position of the end-effector and comparing it to the expected position. With respect to Figure 3.6, consider the first pose, where the end-effector is actually at $\mathbf{p}_{A_1}$, computed via the forward kinematics with the parameter deviations, but it is believed that it is at $\mathbf{p}_{C_1}$, computed via the forward kinematics using the nominal geometry. The set of points, G, represent the graduations and are defined in the simulation by adding known increments to the first actual point, A_1. G_1 is coincident with A_1. With the simulation, imagine that the ruler is placed so that G_1 is in direct contact with the tool flange so that G_1 and A_1 are coincident. However, it is an imaginary ruler represented only by a set of points so no collisions are possible. The end-effector is moved a known distance along the length of the imaginary ruler. It was expected to arrive at G_2, but instead it moved to A_2. The difference between its expected position and its actual position is substituted as the error. This error is not the overall error, which is between A and C, but is a significant component of the overall error.

Increment size is an important factor in the success of the calibration. Increments that are too small would result in a singular Identification Jacobian matrix. Linear dependencies would be formed with respect to the rows as they would be too similar. Therefore, an increment of sufficient magnitude must be chosen to avoid this situation. Now that the means to generate the appropriate poses and measure the appropriate positions to implement the RMC has been established, the results of the relative simulation will now

be discussed.

4.3.3 Simulation Results

It was determined with the relative simulation that many of the parameter deviations were observable and capable of being successfully identified. Using increments in all three base reference frame directions, and the first joint angle set from the KUKA KR 15/2 data, one iteration of the parameter deviation identification equation yielded 15 parameter deviations within 10-15 % of their respective true values. The SVD gave an indication of which parameters were less trustworthy. This result alone is very encouraging. However, to refine the estimates, further iteration was required.

In the iterative solution of identified parameters, the error vector must be updated. Unlike the conventional calibration scheme, the error vector cannot be updated by simply re-computing the forward kinematics of the robot using the corrected geometry and comparing these to the measured values. There are no values to compare to in the relative case. The error vector implemented in the relative procedure is not the total error and is described with respect to the first actual point. This point has been computed in the simulation but would be unknown in the experimental case. In the first iteration, the errors are applied at the first controller point instead. As many of the parameter deviations were identified at close to their respective true values, an estimate of the first actual point can be computed. Applying the error data again at the estimated first actual point allows further iteration. Once it is determined that further iterations would not significantly improve the estimated parameter deviations, their current values could then be used to provide an even better estimate of the first actual point. This process can be repeated until the observable parameters are appropriately identified.

Using the same parameter deviations as listed in Table 4.1, the results of the relative simulation with 2 estimates of the first actual point with 3 iterations of parameter iden-

Table 4.2: Results for relative measurement simulation after 2 estimates of the first actual point.

Parameter	Specified Deviations (m, rad $\times 10^{-6}$)	Identified Deviations (m, rad $\times 10^{-6}$)
θ_1	16.0000	16.0000
θ_2	34.0000	34.0002
θ_3	-56.0000	-56.0003
θ_4	-27.0000	-27.0000
θ_5	22.0000	22.0003
θ_6	13.0000	39.5814
d_1	38.0000	37.9829
d_2	-14.0000	-33.4532
d_3	-53.0000	-33.5468
d_4	61.0000	61.0000
d_5	-30.0000	-30.0023
d_6	24.0000	23.9828
a_1	-17.0000	-17.0000
a_2	89.0000	89.0000
a_3	64.0000	64.0000
a_4	-45.0000	-45.0000
a_5	37.0000	37.0000
a_6	22.0000	21.9999
α_1	-11.0000	-11.0000
α_2	8.0000	8.0000
α_3	19.0000	19.0000
α_4	21.0000	21.0000
α_5	-15.0000	-15.0161
α_6	14.0000	0.0000

tification apiece are listed in Table 4.2. Again, α_6 was excluded from the analysis as it cannot be identified. The initial robot pose was specified as the first pose of the KUKA experimental data. 80 measurements were taken in total and increments in the base coordinate system were specified for all three directions. The increments are: -10.0 mm in the x direction, 10.0 mm in the y direction, and 5.0 mm in the z direction.

As can be seen in Table 4.2, the sum of estimates for d_2 and d_3 equaled that of their collective actual error, and α_6 was not identified correctly. In the first iteration, 4 singular

values were removed. The parameter deviations corresponding to these singular values are, starting with the last column (associated with the first eliminated singular value): d_2 and d_3, θ_6, α_5, and θ_5. For d_2 and d_3, the column had two entries of equal magnitude. In the subsequent iterations, only one singular value was eliminated. It corresponded to d_2 and d_3 every time. As these parameters appeared unobservable they were then removed from the analysis, as they are linked. However, the removal of both of these entries significantly worsened the results. Removal of just one, d_2, yielded similar results as in Table 4.2, except that d_3 was identified as 67.0000, which is again the collective sum of their actual error. Removal of them all yielded results similar to the absolute case.

In terms of convergence the relative simulation was able to reduce the residual to approximately 10^{-14} in 5 iterations and 2 estimates of the first actual point. This can be seen in the logarithmic plot of Figure 4.4. α_6 was not part of the analysis. As can be seen in the plot, there are two iterations in which the residual increases. For the sixth iteration, this increase is due to the second estimate of the first actual point. For the first set, the

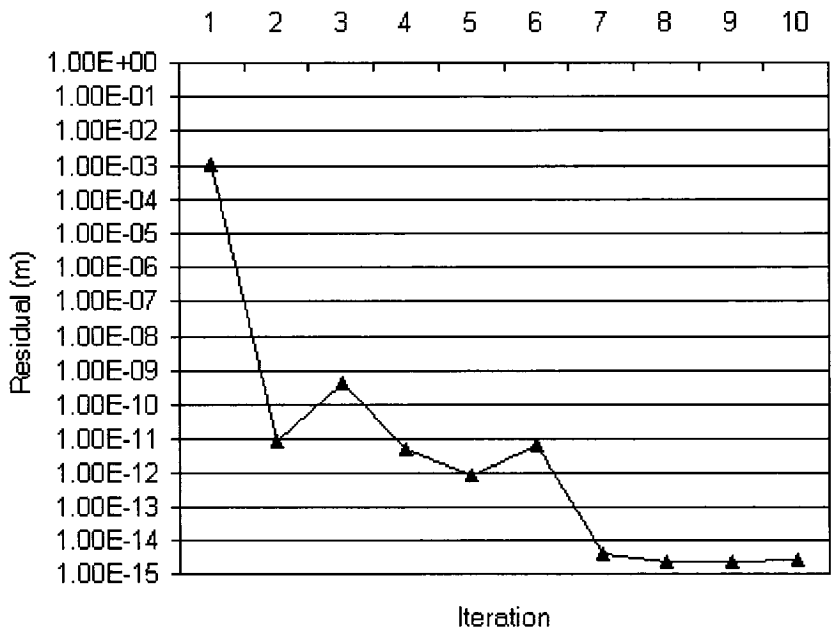

Figure 4.4: Relative simulation residual *vs* iteration.

estimates were converging on solutions that fit with the included error between the first point estimate and the actual first point. At the end of the fifth iteration, a new estimate of the first point is generated and the process is repeated.

As the experimental case will involve measurements in just two orthogonal directions, a check of its performance with a 2×24 Identification Jacobian was pursued. With increments specified in all three directions, the same result as achieved with three directions was produced. The unidentified parameters were the same as in the previous case as well.

Overall, the relative simulation proved to be quite successful. 20 of 24 parameters were successfully identified. Also, the same range of deviation magnitude from the absolute case worked for the relative case. However, the issue of noise was not tested in the simulation. From the effects of noise on the absolute simulation, in all certainty it would have the same detrimental effect on the relative simulation.

Chapter 5

Experimental Validation

The kinematic calibration procedure based on the RMC, through the analysis of the simulation results, has the potential to successfully identify the majority of a robot's kinematic parameters deviations. Several issues were brought to the forefront, such as measurement noise and the straight-line path of the end-effector, as these issues would have significant bearing on the experimental results. To validate the kinematic calibration procedure utilizing the RMC, an automated measurement acquisition procedure had to be devised. The joint angles for the measurement positions are supplied by the robot controller and are used as inputs in the experimental calibration program. The remainder of the program functions are exactly the same as in the simulations. For both robots, whose links are presumed to be constructed with numerically controlled milling machines, deviations on the order of microns were expected.

First, a set of data acquired in the KUKA KR 15/2 experiment [33] was analyzed. As previously mentioned, an experiment was conducted with this robot at the Mining University of Leöben, Austria. Its intent was also to validate the RMC, but it could not be completed due to time constraints. So, the first step in the validation of the RMC was to first utilize the measurements from this experiment in the calibration program. As some

unexpected results came of this and access to this robot was no longer possible, the simulations were developed. Finally, a new experimental setup was constructed and featured a new robot, the Thermo CRS A465. The experimental program code is summarized in Appendix E.

5.1 KUKA KR 15/2 Experiment and Results

The setup for this experiment is similar the setup involving the Thermo CRS A465. The main features of the system are a CCD camera with lens and a precision-ruled straight edge, so both must have these items. A Pulnix TM6-CN CCD camera was outfitted with a Rodenstock 2× lens and electronic shutter and mounted to the tool flange of the KUKA KR 15/2 with a custom-built metal bracket. The PZA ruler conformed to the Deutsches Institut für Normung (DIN) standard 865 and had the dimensions of 1000 mm × 20 mm × 20 mm. A National Instruments PCI-1408 monochrome framegrabber and AT-GPIB General Purpose Interface Board (GPIB) were used to acquire images from the camera. Additionally, two laser displacement devices were used: a MEL M5L/10 laser and a MEL M52L/2 stereo laser. Finally, a red LED array was attached to provide a uniform light source.

The ruler was aligned with the y direction of the base coordinate frame of the robot and the error vector consists of errors only in this direction. The measurements were taken from a span of 80 cm of the 1 m precision-ruled straight edge. The graph of the errors versus the length increment, Figure 5.1, illustrates that the inaccuracy of the KUKA KR 15/2 could be as much as ± 0.8 mm. Also note that there is a considerable influence due to noise contributions.

When this data was supplied to the kinematic calibration procedure, the outputted results were rather unexpected. The estimated kinematic parameters are listed in Table 5.1.

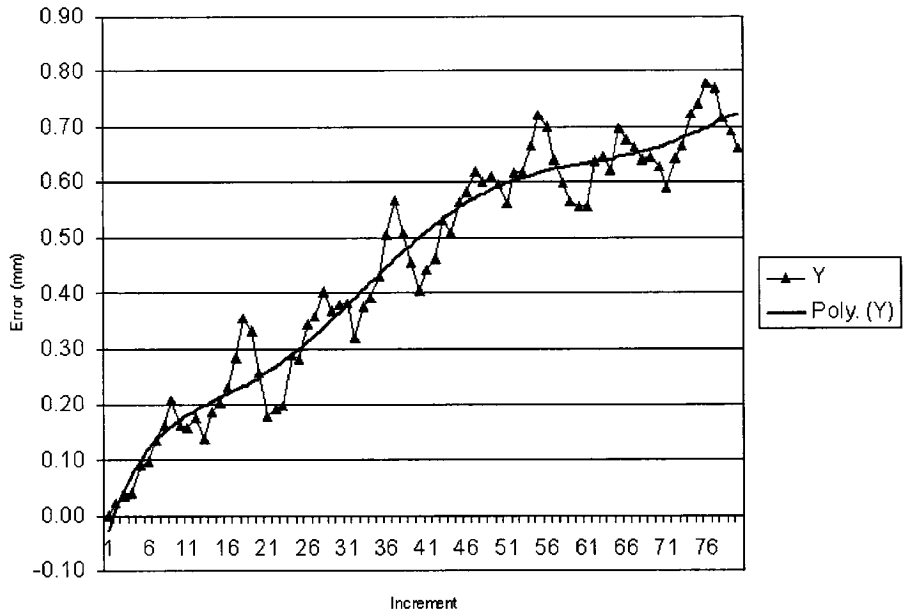

Figure 5.1: Graph of the y direction errors *vs* length increment for the KUKA KR 15/2 experiment.

Obviously, these corrections were completely unusable. They were obtained through one iteration of the identification equation. Further iteration was not possible as these corrections lead to divergence.

As no further access to the robot, equipment, or coded programs, was possible, the simulations were developed in an effort to uncover the negative influences on the procedure. The simulations yielded positive results in the absence of measurement noise so another experiment, featuring the Thermo CRS A465 six-axis serial robot, was devised.

5.2 Thermo CRS A465 Experiment

The Thermo CRS A465 was mounted to an aluminum surface, 1524 mm × 1219 mm × 12.7 mm, that was rigidly fastened to a steel table structure. There are 9 10-24 holes in the measurement area of the aluminum surface so that the ruler, a PZA 1000 mm × 20 mm × 20 mm precision-ruled straight edge, conforming to DIN 865, can be placed in numerous positions

Table 5.1: Results of the experimental calibration procedure applied to the KUKA KR 15/2 data.

Parameter	Identified Deviations (m, rad)
θ_1	-2.8452
θ_2	-94.3760
θ_3	-39759.3370
θ_4	114.7173
θ_5	20.1933
θ_6	0.0000
d_1	0.0000
d_2	-335.1989
d_3	-335.1989
d_4	-6192.9273
d_5	137.7800
d_6	8.6397
a_1	5374.3238
a_2	104.8907
a_3	23988.9568
a_4	-137.1946
a_5	144.2363
a_6	-3.4559
α_1	-0.0422
α_2	-21.9418
α_3	-913.3603
α_4	1187.1736
α_5	19.3085
α_6	0.0000

against 6.35 mm ($\frac{1}{4}$") shoulder screws. DIN 865 specifies a 20 μm tolerance in the distance between graduation centres. The A465 is interfaced with a C500C controller, which is serially connected to a development computer. The controller is also serially connected to another development computer that contains the data acquisition card and software necessary for image extraction from the Pulnix TM200 CCD camera. The camera is outfitted with a Rodenstock 1× lens and a red LED ring-light. The experimental setup is pictured in Figure 5.2.

The A465 was commissioned in accordance with its user guide [40] and a program, executed through its application shell [41], was designed to accomplish the necessary robot movements using the RAPL-III programming language [42]. This program communicated with the other computer via the serial connection when a desired pose was reached. A Labview Virtual Instrument (VI) program then snapped an image and signified that the robot controller execute the next pose. This automated procedure was used to acquire images along the length of the measurement artifact. An outline of the programs written to automate the procedure can be found in Appendix F.

In the experiment, the ruler was placed nearly parallel to the y direction of the robot base frame. On the table, a slight curvature could be seen underneath the ruler, indicating that the surface was not precisely flat. The first measurement pose was taught through use of the teach pendant. The edge of the ruler was aligned so that it would be horizontal in the image and the graduations were therefore vertical. The ruler's edge was approximately 1 mm (100 pixels) away from the top border of the image. As the ruler was not perfectly

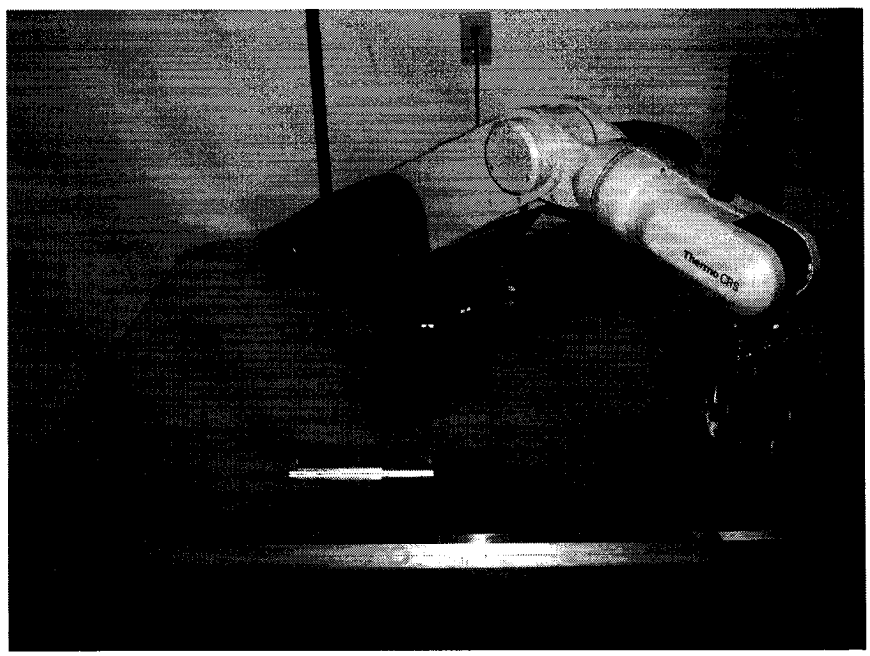

Figure 5.2: Thermo CRS A465 experimentation setup with ruler placement.

aligned with the y direction of the base frame, the end-effector had to be appropriately rotated so that when it was commanded to move the entire 80 cm span of the ruler considered, the first pose image appeared to be the same as the final pose image. With respect to the first point, the final point was displaced by approximately 2 mm in the z direction and 2 mm in the x direction.

To account for the effects of measurement noise and the repeatability of the robot, 50 sets of data were taken. The robot was first commanded to mimic the image acquisition process for 45 minutes to warm it up. The robot was operating at 10 % of its maximum speed. For this experiment, every pose was defined relative to the first pose. As the first pose is a taught position, it has a tool coordinate frame that is the least affected by potential error. So, the subsequent poses were defined as multiples of a specified increment, 1 cm, from the first pose. After reaching these poses, the robot would return to the first pose to re-establish the proper tool coordinate frame. In this fashion, one set of data, 80 measurements, took approximately 45 minutes to complete.

5.3 Experimental Results

Once the images were acquired they were digitally processed to extract metric information. Figure 5.3 is the reference image of the first data set. In the image processing algorithm, the edge of the ruler and the five visible minor graduations are used as markers. The distance between minor graduations is 0.1 mm and is certified, according to its calibration certificate, to be accurate to $\pm$ 3 μm. This known distance achieves scale in the image coordinate system as the distance between graduations can be calculated in pixels. When a comparison is made between the reference image and a subsequent image the relative distance between these image markers, using the scale to compute metric information, is the error.

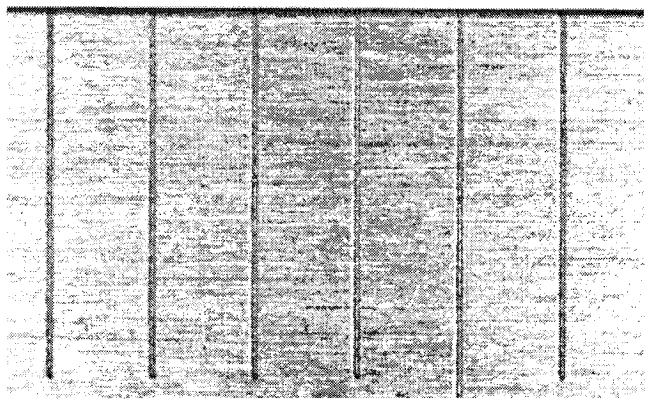

Figure 5.3: Reference image of the first data set.

To confirm that the A465 did indeed have a repeatability of ± 50 μm, an analysis of the reference images, the first image of each data set, was performed. The first reference image of the first data set was used as the reference for this analysis. With respect to the graph of Figure 5.4, certain anomalies were present.

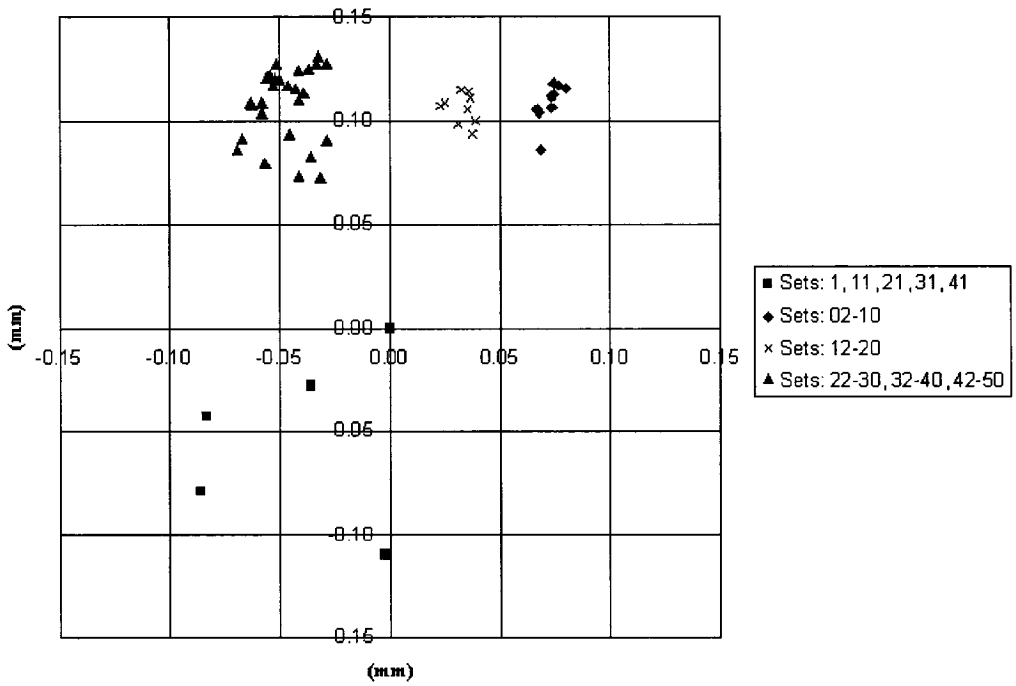

Figure 5.4: Repeatability analysis of the reference images.

The first issue that should be noted is that these measurements were taken over a five-day period. Ten data sets were collected in one day with 45 minutes allotted for warm-up. An issue with the first pose of a set of measurements has been noticed before with laboratory experiments investigating repeatability [43]. This issue was confirmed when checking the recorded joint angles. The joint angles for each set are exactly the same, except for the first recorded pose in the program. As the first pose is a taught position, this discrepancy should not exist. However, it may be an irregularity in the controller software and unfortunately no compensation can be performed. This does not mean that the remaining images of those particular sets are useless. According to the recorded joint angles, these poses were consistent with their counterparts in other data sets, so just the reference image may be unusable.

The second issue was that three clusters of data points were observed. Each corresponding to sets 2-10, sets 12-20, and finally a larger cluster representing sets 22-30, 32-40, and 42-50. As the ruler was not moved over the course of the five days, no tangible evidence can be offered to explain this result. No aspect of the physical setup was changed during this time. It may be an artifact of the digital image processing, however, it might also have been the ambient temperature and humidity. These were not recorded and any difference remained unnoticed while present during the entire experiment. Despite this result a significant portion of the data seems appropriate for further use. The larger cluster represents 27 data sets and is within the repeatability of the machine.

The relative error for the first acceptable data set, the 22^{nd}, is plotted in Figure 5.5. Also plotted are the trend-polynomials for each direction. The reference position was taken as the averaged value of the positional coordinates for the reference images of the 27 data sets. As can be seen, there is a significant presence of measurement noise. Also, the inaccuracy of the A465 reached approximately 0.55 mm. It is unfortunate that the data is plagued by measurement noise, but this experiment has at least been consistent with

the KUKA KR 15/2 experiment in this regard. Two trend-polynomials were generated, using sixth degree polynomials, so that error data in this manner could be supplied to the calibration procedure as well.

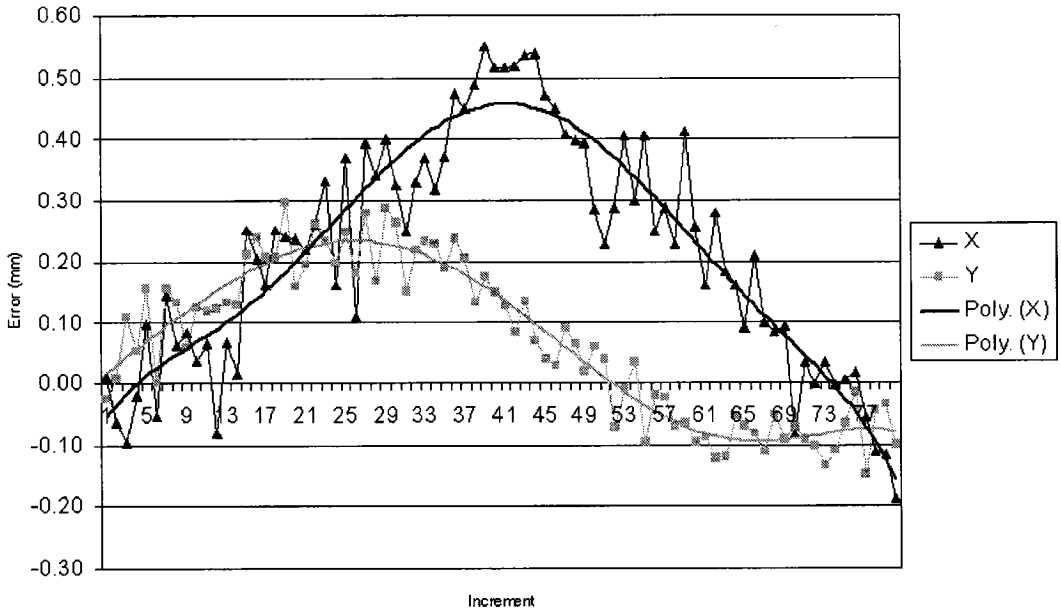

Figure 5.5: Relative error of the 22^{nd} data set.

Both sets of error data were supplied to the experimental calibration program. The first set, afflicted with measurement noise, resulted in the same kind of parameter corrections as with the KUKA KR 15/2 data. The corrections were again quite large. With the trend-polynomial data, the magnitude of the corrections were significantly reduced. As opposed to a -1771 rad correction to θ_3, an 80.8 rad was produced. The estimates, for both cases, can be seen in Table 5.2. Unfortunately, these corrections are again not very helpful as they make the calibration process diverge.

As more data was available, the averages, computed for the 27 applicable measurement sets, were substituted for the 80 data points. With respect to the plot of Figure 5.6, it can be seen that the measured points, even when averaged, follow the same general path. This indicates that the noise in the data may not be primarily influenced by the robot's

Table 5.2: Results of the experimental calibration procedure applied to the 22^{nd} data and trend-polynomial.

Parameter	Noisy Data Identified Deviations (m, rad)	Trend-polynomial Identified Deviations (m, rad)
θ_1	0.0523	-0.0143
θ_2	-0.0886	0.0521
θ_3	-1770.6634	-80.7586
θ_4	-0.0089	0.1893
θ_5	0.0089	-0.0026
θ_6	0.0000	0.0000
d_1	0.0000	0.0000
d_2	-0.0028	-0.0755
d_3	-0.0028	-0.0755
d_4	0.0286	0.0225
d_5	0.6926	0.0914
d_6	-0.0801	0.0231
a_1	134.7948	6.1774
a_2	-0.0181	-0.0121
a_3	588.0014	26.7630
a_4	-0.1428	0.0476
a_5	0.1173	-0.0344
a_6	-0.0000	-0.0001
α_1	-0.0447	-0.0140
α_2	-0.0387	-0.0004
α_3	2.0006	-0.1948
α_4	-1.5516	0.1988
α_5	-0.0053	-0.0070
α_6	0.0000	0.0000

repeatability and may be due to other sources. A repeatability analysis was performed for the applicable data sets for the first 20 points. Plots corresponding to these points illustrated that they were within the repeatability of the A465. The standard deviation of the radial distances for all 80 points did not exceed 0.046 μm. The results of the experimental calibration program were nearly the same as with the 22^{nd} data set.

One significant observation can be made about the results of the experimental calibra-

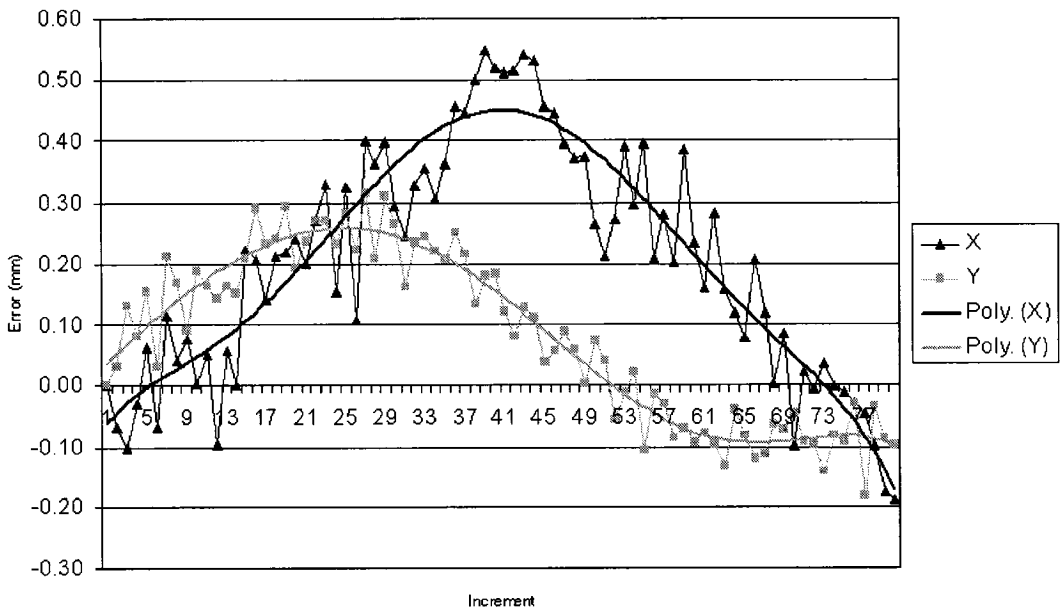

Figure 5.6: Relative error of the considered data sets.

tion procedure. The estimates were all large relative to the assumed deviation magnitudes in the simulations. Recall that when a straight-line path was used in the absolute simulation and measurement noise was present, the estimates also grew in size. The included random measurement error was of the same magnitude as the specified deviations. The results of both of these case are very similar and indicates that the issues to be resolved with the experimental setup are: aligning the ruler with one of the base frame coordinate directions and an accumulation of noise. One of these can be addressed rather easily by just orienting the ruler so that it no aligned with any of the base frame directions. This may involve the construction of a special stand to hold it in such a position. The other, the more difficult to address, is the issue of noise. However, to eliminate noise the sources must be identified and some of these sources are presented next.

5.3.1 Sources of Error

The future of the project may lie in a complete error analysis of the system to compensate for the multiple sources of noise. As noted with the absolute simulation the inclusion of measurement noise caused the parameter estimates to become inaccurate and grow in magnitude. However, they grew to be of the same magnitude as the specified magnitude of the noise. The estimates for d_2 and d_3 are excluded from this observation as they seem to be generally sensitive to many factors. Therefore, to see two experiments result in corrections measured in kilometers was unexpected. When trend-polynomials were employed, the estimates became smaller, but not enough to lead to convergence. So, a more advanced simulation, that incorporates all the potential sources of error in the measurement acquisition process, is required.

The first source of error is the measurement artifact itself. It is a precision-ruled straight edge, but is still subject to inaccuracies in the manufacturing process. The PZA ruler procured for the experiment conforms to DIN 865, which stipulates that the centre-to-centre distance between graduations is within ± 10 μm. This may be too great a tolerance given the experimentally-verified repeatability of the robot.

Noise generated during the image processing is another source of error. The CCD camera creates images that are 640×480 pixels in size. From the analysis of Figure 5.3 the distance between graduations is approximately 100 pixels. One pixel is therefore equal to approximately 10 μm. Thus, the error instilled in the measurements could be as much as this amount [5].

The next source of error originates from the limitations of the robot's joints. When the end-effector is commanded to a pose, from the first pose, the orientation component of the transformation should remain constant while the translational component indicates the desired increment. However, the theoretical joint angles, as computed by the controller may not be reachable. The first three joint encoders of the A465 have a resolution of

0.00712871 degrees which corresponds to approximately 50000 motor positions for 360 degrees of revolution. The last three have half of that with a resolution of 0.00356436 degrees. The resolution of the joint encoders, in part, determine the repeatability of the robot. So, when the robot is commanded to a particular set of joint angles, it moves to the motor positions that are the closest to those values rather than the desired values.

Also associated with the joint encoders is error due to their inaccuracy. In general, angular positions are recognized by markers on the motor shaft. However, some inaccuracy exists when the location of the marker is sensed by the encoder. This error is potentially of the same order of magnitude as the difference between the discrete motor positions and the desired angular positions.

Finally, another source of error arises in traversing the length of the ruler. The only way to determine whether the end-effector has gone in a straight line along the length of the ruler is to compare the first and last images. As these images are subjected to various errors, including the repeatability of the robot and the pixilation of the image, the edge of the ruler may not be precisely followed. Compensation of this type of error is possible by removing the difference between the desired path, where the first and last images are almost identical, and the actual path, where the difference between the first and last image is greater than the repeatability of the robot. As this error is linear in nature it can be easily computed. However, the accuracy of the image processing would be important as the metric information used to determine this path error would be subject to the error generated in the image processing.

Chapter 6

Summary & Conclusions and Recommendations

6.1 Summary & Conclusions

This thesis provides a foundation for the development of a kinematic calibration procedure for six-axis serial robots and is based on the RMC. The use of relative measurements, as opposed to absolute measurements, were substituted to determine the deviations between the nominal robot parameters, derived from the ideal robot geometry, and the actual parameters, which are unknown and dependent on the accuracy of the manufacturing methods employed during the robot's construction. Simulations were created to evaluate the use of relative measurements and, based on their findings, proved to be quite promising. The simulation and preliminary experimentation results serve as a basis for future efforts directed towards an automated camera-based kinematic calibration system. Two key items were addressed:

(i) The developmentof a simulation of the kinematic calibration of six-axis serial robots using relative measurements, and the analytical expressions for its fundamental com-

ponents.

(ii) The identification of the sources of error in the experimental calibration procedure through the development of an automated camera-based measurement acquisition system.

In order to perform the kinematic calibration of serial robots several analytic tools were required. The capability to compute the forward kinematics, inverse kinematics, and Identification Jacobian elements were necessary for any simulation or experimental program created as a part of this project. Inherent to their use was the selection of an appropriate parameterization method. The DH and MDH conventions were utilized to model two six-axis serial robots: the KUKA KR 15/2 and the Thermo CRS A465. The fundamental principles associated with robot kinematics were discussed in Chapter 2 and applied in the calibration procedure.

A kinematic calibration procedure was developed to first use absolute measurements, those described with respect to the base coordinate frame of the robot, and then adapted to use relative measurements, which are described with respect to an arbitrary position. An estimate for the robot parameter deviations can be computed as they are related to the errors in the position of the end-effector by the Identification Jacobian. The RMC was introduced to provide an alternate means by which to measure these errors. The RMC is a novel means of acquiring error data in that all of the images corresponding to the measurement poses are compared to the reference image of the first pose. Along with the development of the conventional calibration scheme and a description of SVD, the RMC was developed in Chapter 3.

The structure of the simulations and their results were presented in Chapter 4. For the absolute simulation a means to generate poses, acquire measurements, incorporate noise and establish convergence were discussed. The absolute simulation proved to be quite

successful in identifying the specified parameters. The incorporation of measurement noise was found to be detrimental to the successful identification of the parameter deviations.

Finally, the experimental results for a KR 15/2 and an A465 six-axis serial robots were presented in Chapter 5. It was observed that the experimental data suffered from a great deal of measurement noise. The primary source of this noise was not the repeatability of the robot but, other sources influenced by the capabilities of the robot. It was also noted that the maximum measured error was an order of magnitude larger than that of the repeatability. This stands to reason as robot accuracy is typically an order of magnitude worse than repeatability.

The following major conclusions have been drawn from the work presented herein.

- In its current state of development, the kinematic calibration simulation, based on the *Relative Measurement Concept*, can successfully identify 20 of 24 robot parameter deviations.

- The joint twist of the sixth joint, α_6, is unobservable. α_6 is a rotation about an axis through the point that is measured. The effect of this rotation does not affect subsequent links and is thus unobservable.

- To identify the link offsets of the second and third joints, d_2 and d_3, sufficient error in the joint twist for the second joint, α_2, must be present.

- The error observed in the experimental data is due to multiple sources: the repeatability of the robot; the measurement artifact; the digital image processing; the resolution and inaccuracy of the joint encoders; and the straight-line path of the ruler.

6.2 Recommendations

- Attempt a non-linear solution for the kinematic parameter deviations in the simulations.

- Modify the absolute simulation to include the discrete motor angular positions, noise in motor angular position, measurement artifact inaccuracy, and noise in the measurements. This would serve to confirm the magnitude of errors experienced in the KR 15/2 and A465 experiments (0.55-0.80 mm).

- Implement the Hayati parameters or the General DH parameterization to cope with the nearly-parallel axes of the second and third joints.

- Acquire another set of measurements but place the ruler such that it is no longer aligned with a base coordinate direction. Construct a set of discs that would be mounted to the table surface via the existing 10-24 holes. A great number of ruler orientations could be achieved in the measurement area of the table with these discs. Also, attempt to place the ruler not only at an angle in the xy plane of the base coordinate system, but raise one of the ends.

- As a CMM is now locally available, calibrate the A465 using absolute measurements. This could be accomplished with a tooling ball and a suitable plate attachment. Some of the error sources mentioned in Chapter 5 would then no longer apply, but be replaced by the inaccuracy of the CMM. However, the only sources of error would be due to the CMM's inaccuracy, the parameter deviations and the repeatability of the A465. A comparison to an absolute calibration method in terms of time, setup, and parameter estimates, would be beneficial.

- Investigate the possibility of adapting the procedure to cope with different tasks, applied loads, working speed, and temperature. As accuracy and repeatability vary

over the workspace and change according to these influences, the calibration results also depend on these factors.

References

[1] N. W. Simpson and M. John D. Hayes. Kinematic Calibration of Industrial Manipulators. *19th Canadian Congress of Applied Mechanics*, 1:180–181, 2003.

[2] Andrew Fratpietro, Nick Simpson, and M. John D. Hayes. Advances in Robot Kinematic Calibration. Technical report, Carleton University, 2003.

[3] N.W. Simpson and M.J.D. Hayes. Simulation of a Kinematic Calibration Procedure that Employs the Relative Measurement Concept. *CSME Forum 2004*, pages 62–71, 2004.

[4] Geo-Ry Tang and Lu-Sin Liu. A Study of Three Robot Calibration Methods Based on Flat Surfaces. *Mechanism and Machine Theory*, 29(2):195–206, 1994.

[5] Andrew Fratpietro and M.J.D. Hayes. Relative Measurement for Kinematic Calibration Using Digital Image Processing. *CSME Forum 2004*, pages 758–767, 2004.

[6] A. Doria, F. Angrilli, and S. de Marchi. Inverse Kinematics Robot Calibration by Spline Functions. *Applied Mathematical Modelling*, 17(9):492–498, September 1993.

[7] Jure Zupancic. Calibration of an SMT Robot Assembly Cell. *Journal of Robotic Systems*, 11(4):301–310, 1994.

[8] Edward J. Park, Weihua Xu, and James K. Mills. Calibration-Based Absolute Localization of Parts for Multi-Robot Assembly. *Robotics*, 20(4):359–366, 2002.

[9] M. Khoshzaban, F. Sassani, and P.L. Lawrence. Kinematic Calibration of Industrial Hydraulic Manipulators. *Robotica*, 14(5):541–551, 1996.

[10] Alberto Omodei, Giovanni Legnani, and Riccardo Adamini. Calibration of a Measuring Robot: Experimental Results on a 5 DOF Structure. *Journal of Robotic Systems*, 18(5):237–250, 2001.

[11] M. Ikits and John M. Hollerbach. Kinematic Calibration Using a Plane Constraint. *Proc. IEEE International Conference on Robotics and Automation, Albuquerque, NM*, pages 3191–3196, 1997.

[12] W. Khalil, S. Bernard, and P. Lemoine. Comparison Study of the Geometric Parameter Calibration Methods. *International Journal of Robotics and Automation*, 15(2):56–68, 2000.

[13] M.R. Driels and U.S. Pathre. Robot Calibration Using an Automatic Theodolite. *International Journal of Advanced Manufacturing Technology*, 9(2):114–125, 1994.

[14] Ibrahim A. Sultan and John G. Wager. A Technique for the Independent-Axis Calibration of Robot Manipulators with Experimental Verification. *International Journal of Computer Integrated Manufacturing*, 14(5):501–512, 2001.

[15] M. Abderrahim and A.R. Whittaker. Kinematic Model Identification of Industrial Manipulators. *Robotics and Computer Integrated Manufacturing*, 16:1–8, 2000.

[16] Alberto Omodei, Giovanni Legnani, and Riccardo Adamini. Three Methodologies for the Calibration of Industrial Manipulators: Experimental Results on a SCARA Robot. *Journal of Robotic Systems*, 17(6):291–307, 2000.

[17] P. Drouet, S. Dubowsky, S. Zeghloul, and C. Mavroidis. Compensation of Geometric and Elastic Errors in Large Manipulators with an Application to a High Accuracy Medical System. *Robotica*, 20(3):341–352, 2002.

[18] Hanqi Zhuang, K. Wang, and Z.S. Roth. Simultaneous Calibration of a Robot and a Hand-Mounted Camera. *IEEE Transactions on Robotics and Automation*, 11(5):649–661, 1995.

[19] Hanqi Zhuang, Wen-Chiang Wu, and Zvi S. Roth. Camera-Assisted Calibration of SCARA Arms. *IEEE Robotics and Automation Magazine*, 3(4):46–52, 1996.

[20] Jose Mauricio S.T. Motta, Guilherme C. de Carvalho, and R.S. McMaster. Robot Calibration Using a 3D Vision-based Measurement System with a Single Camera. *Robotics and Computer Integrated Manufacturing*, 17(6):487–497, 2001.

[21] P. Rousseau, A. Desrochers, and N. Krouglicof. Machine Vision System for the Automatic Identification of Robot Kinematic Parameters. *IEEE Transactions on Robotics and Automation*, 17(6):972–999, 2001.

[22] Yan Meng and Hanqi Zhuang. Self-Calibration of Camera-Equipped Robot Manipulators. *International Journal of Robotics Research*, 20(11):909–921, 2001.

[23] H. Zhuang, L.K. Wang, and Z.S. Roth. Error-Model-Based Robot Calibration Using a Modified CPC Model. *Robotics and Computer Integrated Manufacturing*, 10(4):287–299, 1993.

[24] Koichiro Okamura and F.C. Park. Kinematic Calibration Using the Product of Exponentials Formula. *Robotica*, 14(4):415–421, 1996.

[25] I-Ming Chen, Guilin Yang, Chee Tat Tan, and Song Huat Yeo. Local POE Model for Robot Kinematic Calibration. *Mechanism and Machine Theory*, 36(11):1215–1239, 2001.

[26] G. Zak, R. Fenton, and B. Benhabib. A Simulation Technique for the Improvement of Robot Calibration. *Transactions of the ASME Journal of Mechanical Design*, 115(3):674–679, September 1993.

[27] John M. Hollerbach and Charles W. Wampler. The Calibration Index and Taxonomy for Robot Kinematic Calibration Methods. *International Journal of Robotics Research*, 15(6):573–592, 1996.

[28] Hanqi Zhuang, Jie Wu, and Weizhen Huang. Optimal Planning of Robot Calibration Experiments by Genetic Algorithms. *Journal of Robotic Systems*, 14(10):741–752, 1997.

[29] Chunhe Gong, Jingxia Yuan, and Jun Ni. A Self-Calibration Method for Robotic Measurement System. *Journal of Manufacturing Science and Engineering*, 122(1):174–181, 2000.

[30] David C. Lay. *Linear Algebra and its Applications.* Addison Wesley Longman, Inc., 2nd edition, 1997.

[31] J. Denavit and R.S. Hartenberg. A Kinematic Notation for Lower Pair Mechanisms Based on Matrices. *Journal of Applied Mechanics*, 22:215–221, 1955.

[32] John J. Craig. *Introduction to Robotics: Mechanics and Control.* Addison-Wesley Publishing Company, 2nd edition, 1989.

[33] M.J.D. Hayes and P.L. O'Leary. Kinematic Calibration Procedure for Serial Robots with Six Revolute Axes. Technical report, Institute for Automation, University of Leoben, April 12 2001.

[34] D. Pieper and B. Roth. The Kinematics of Manipulators Under Computer Control. *Proceedings of the Second International Congress on Theory of Machines and Mechanisms, Zakopane, Poland*, 2:159–169, 1969.

[35] Saeed B. Niku. *Introduction to Robotics: Analysis, Systems, Applications.* Prentice Hall, 2001.

[36] Benjamin W. Mooring, Zvi S. Roth, and Morris R. Driels. *Fundamentals of Manipulator Calibration.* John Wiley Sons, Inc., 1991.

[37] Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, and William H. Press. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 2nd edition, 1992.

[38] R.C. Hibbeler. *Engineering Mechanics: Statics and Dynamics.* Prentice Hall, 8th edition, 1998.

[39] S.A. Hayati. Robot Arm Geometric Link Parameter Estimation. *Proc. 22nd IEEE Conf. on Decision and Control San Antonio, TX*, pages 1477–1483, 1983.

[40] Thermo CRS, Ltd. *A465 Robot User System Guide UMI-A465-400*, 002a edition, July 2002.

[41] Thermo CRS, Ltd. *Application Shell ASH UMI-R3-161*, 002 edition, November 2000.

[42] CRS Robotics Corporation. *RAPL-3 Language Reference Guide UMI-R3-210*, 001b edition, November 2000.

[43] Nicholas W. Simpson and M.J.D. Hayes. *Repeatability of Serial Robots.* Laboratory Manual, Dept. of Mechanical and Aerospace Engineering, Carleton University, January 2004.

Appendix A

Pieper's Inverse Kinematic Solution Method

Pieper's solution to the inverse kinematic problem is presented here for all cases. The simplifying functions are as follows:

$$f_1 = a_3 \cos\theta_3 + d_4 \sin\alpha_3 \sin\theta_3 + a_2, \tag{A.1}$$

$$f_2 = a_3 \cos\alpha_2 \sin\theta_3 - d_4 \sin\alpha_3 \cos\alpha_2 \cos\theta_3 - d_4 \sin\alpha_2 \cos\alpha_3 - d_3 \sin\alpha_2, \tag{A.2}$$

$$f_3 = a_3 \sin\alpha_2 \sin\theta_3 - d_4 \sin\alpha_3 \sin\alpha_2 \cos\theta_3 + d_4 \cos\alpha_2 \cos\alpha_3 + d_3 \cos\alpha_2, \tag{A.3}$$

$$\begin{aligned} g_1 = {} & a_3 \cos\theta_2 \cos\theta_3 + d_4 \cos\theta_2 \sin\alpha_3 \sin\theta_3 + a_2 \cos\theta_2 \\ & - a_3 \sin\theta_2 \cos\alpha_2 \sin\theta_3 + d_4 \sin\theta_2 \sin\alpha_3 \cos\alpha_2 \cos\theta_3 \\ & + d_4 \sin\theta_2 \sin\alpha_2 \cos\alpha_3 + d_3 \sin\theta_2 \sin\alpha_2 + a_1, \end{aligned} \tag{A.4}$$

$$\begin{aligned} g_2 = {} & a_3 \sin\theta_2 \cos\alpha_1 \cos\theta_3 + d_4 \sin\theta_2 \cos\alpha_1 \sin\alpha_3 \sin\theta_3 \\ & + a_2 \sin\theta_2 \cos\alpha_1 + a_3 \cos\theta_2 \cos\alpha_1 \cos\alpha_2 \sin\theta_3 \end{aligned}$$

$$
\begin{aligned}
& -d_4 \cos\theta_2 \cos\alpha_1 \sin\alpha_3 \cos\alpha_2 \cos\theta_3 - d_4 \cos\theta_2 \cos\alpha_1 \sin\alpha_2 \cos\alpha_3 \\
& -d_3 \cos\theta_2 \cos\alpha_1 \sin\alpha_2 - a_3 \sin\alpha_1 \sin\alpha_2 \sin\theta_3 \\
& +d_4 \sin\alpha_1 \sin\alpha_3 \sin\alpha_2 \cos\theta_3 - d_4 \sin\alpha_1 \cos\alpha_2 \cos\alpha_3 \\
& -d_3 \sin\alpha_1 \cos\alpha_2 - d_2 \sin\alpha_1, \qquad \text{(A.5)}
\end{aligned}
$$

$$
\begin{aligned}
g_3 \;=\; & a_3 \sin\theta_2 \sin\alpha_1 \cos\theta_3 + d_4 \sin\theta_2 \sin\alpha_1 \sin\alpha_3 \sin\theta_3 \\
& +a_2 \sin\theta_2 \sin\alpha_1 + a_3 \cos\theta_2 \sin\alpha_1 \cos\alpha_2 \sin\theta_3 \\
& -d_4 \cos\theta_2 \sin\alpha_1 \sin\alpha_3 \cos\alpha_2 \cos\theta_3 \\
& -d_4 \cos\theta_2 \sin\alpha_1 \sin\alpha_2 \cos\alpha_3 - d_3 \cos\theta_2 \sin\alpha_1 \sin\alpha_2 \\
& +a_3 \cos\alpha_1 \sin\alpha_2 \sin\theta_3 - d_4 \cos\alpha_1 \sin\alpha_3 \sin\alpha_2 \cos\theta_3 \\
& +d_4 \cos\alpha_1 \cos\alpha_2 \cos\alpha_3 + d_3 \cos\alpha_1 \cos\alpha_2 + d_2 \cos\alpha_1, \qquad \text{(A.6)}
\end{aligned}
$$

$$
k_1 \;=\; a_3 \cos\theta_3 + d_4 \sin\alpha_3 \sin\theta_3 + a_2, \qquad \text{(A.7)}
$$

$$
k_2 \;=\; -a_3 \cos\alpha_2 \sin\theta_3 + d_4 \sin\alpha_3 \cos\alpha_2 \cos\theta_3 + d_4 \sin\alpha_2 \cos\alpha_3, \qquad \text{(A.8)}
$$

$$
\begin{aligned}
k_3 \;=\; & +d_3 \sin\alpha_2 - 2d_2 d_4 \sin\alpha_3 \sin\alpha_2 \cos\theta_3 + a_1^2 + d_2^2 + 2d_2 d_3 \cos\alpha_2 \\
& +2a_2 a_3 \cos\theta_3 + a_2^2 + 2a_2 d_4 \sin\alpha_3 \sin\theta_3 + 2a_3 d_2 \sin\alpha_2 \sin\theta_3 \\
& +2d_2 d_4 \cos\alpha_2 \cos\alpha_3 + d_4^2 + a_3^2 + d_3^2 + 2d_3 d_4 \cos\alpha_3, \qquad \text{(A.9)}
\end{aligned}
$$

$$
\begin{aligned}
k_4 \;=\; & a_3 \cos\alpha_1 \sin\alpha_2 \sin\theta_3 - d_4 \sin\alpha_3 \sin\alpha_2 \cos\theta_3 \\
& +d_4 \cos\alpha_2 \cos\alpha_3 + d_3 \cos\alpha_2 + d_2. \qquad \text{(A.10)}
\end{aligned}
$$

The first angle that is identified is θ_3. For Case 1, $a_1 = 0$, and therefore Equation 2.40 reduces to:

$$
r^2 = k_3. \qquad \text{(A.11)}
$$

Through substitution of a single variable, u_3, using the identities of Equation 2.44,

Equation A.11 can be re-written as a quadratic equation,

$$\begin{aligned}
&(-2a_3a_2 + 2d_2d_4\cos\alpha_2\cos\alpha_3 + 2d_2d_4\sin\alpha_3\sin\alpha_2 + a_3^2 + a_1^2 + d_2^2 \\
&+d_4^2 + a_2^2 + d_3^2 + 2d_3d_4\cos\alpha_3 + 2d_2d_3\cos\alpha_2 - r^2)u_3^2 \\
&+(4a_2d_4\sin\alpha_3 + 4d_2a_3\sin\alpha_2)u_3 \\
&+2d_2d_3\cos\alpha_2 + 2a_2a_3 - r^2 - 2d_2d_4\sin\alpha_3\sin\alpha_2 + d_3^2 \\
&+2d_2d_4\cos\alpha_2\cos\alpha_3 + 2d_3d_4\cos\alpha_3 + a_3^2 + a_1^2 + d_2^2 + a_2^2 + d_4^2 = 0. \qquad \text{(A.12)}
\end{aligned}$$

The coefficients of the powers of u_3 are converted into Matlab code in order to calculate the roots. After a comparison is made, the proper θ_3 can be identified.

Once the proper root is obtained, θ_3 can be computed with:

$$\theta_i = 2\tan^{-1}(u_i). \qquad \text{(A.13)}$$

The solution for θ_2 and then θ_1 will be discussed shortly.

For Case 2, $\sin\alpha_1 = 0$, and therefore Equation 2.41 reduces:

$$z = k_4. \qquad \text{(A.14)}$$

The resulting transcendental equation is reduced to a quadratic equation via the geometric substitutions:

$$\begin{aligned}
&(d_4\cos\alpha_1\sin\alpha_3\sin\alpha_2 + d_4\cos\alpha_1\cos\alpha_2\cos\alpha_3 + d_3\cos\alpha_1\cos\alpha_2 \\
&+d_2\cos\alpha_1 - z)u^2 + (2a_3\cos\alpha_1\sin\alpha_2)u - d_4\cos\alpha_1\sin\alpha_3\sin\alpha_2 \\
&+d_4\cos\alpha_1\cos\alpha_2\cos\alpha_3 + d_3\cos\alpha_1\cos\alpha_2 + d_2\cos\alpha_1 - z = 0. \qquad \text{(A.15)}
\end{aligned}$$

For Case 3, Equation 2.43 applies:

$$\frac{(r-k_3)^2}{4a_1^2}+\frac{(z-k_4)^2}{\sin^2\alpha_1}=k_1^2+k_2^2. \tag{A.16}$$

As before, the geometric identities are applied to obtain a polynomial expression. As this is the general case, the polynomial is 4^{th} and quite complex.

$$\begin{aligned}
&(-8a_1^2z\cos\alpha_1d_4\cos\alpha_2\cos\alpha_3+8d_2^2d_3\cos\alpha_2^2d_4\cos\alpha_3\\
&-8d_2^2d_4^2\cos\alpha_2^2\cos\alpha_3^2\cos\alpha_1^2+4d_4^3\cos\alpha_3d_3+4a_3^2d_4\cos\alpha_3d_3\\
&+4a_1^2z^2+4a_1^2d_3^2\cos\alpha_2^2-4a_2^2d_2d_4\sin\alpha_3\sin\alpha_2\cos\alpha_1^2-8a_3a_2d_2d_4\sin\alpha_3\sin\alpha_2\\
&-4d_4^3d_2\sin\alpha_3\sin\alpha_2\cos\alpha_1^2+4r^2d_2d_4\sin\alpha_3\sin\alpha_2\cos\alpha_1^2+r^4+4d_4^2\cos\alpha_3^2d_3^2\\
&+4a_1^2d_2d_4\cos\alpha_2\cos\alpha_3+4a_3^2d_2d_4\sin\alpha_3\sin\alpha_2-a_1^4\cos\alpha_1^2-2d_2^2r^2-d_4^4\cos\alpha_1^2\\
&-2a_2^2r^2+12d_2d_3^2\cos\alpha_2d_4\cos\alpha_3+8a_1^2d_4\cos\alpha_3d_3\cos\alpha_2^2+4d_3^3d_4\cos\alpha_3\\
&-r^4\cos\alpha_1^2+8d_2^2d_3\cos\alpha_2d_4\sin\alpha_3\sin\alpha_2-8a_3a_2d_2d_3\cos\alpha_2-d_2^4\cos\alpha_1^2\\
&-d_3^4\cos\alpha_1^2+4d_2^2d_3^2\cos\alpha_2^2+4d_2^3d_4\sin\alpha_3\sin\alpha_2-4d_2^2d_4^2\cos\alpha_2^2\\
&-a_3^4\cos\alpha_1^2+8d_2d_4^2\cos\alpha_2\cos\alpha_3^2d_3-4d_2^2d_4\cos\alpha_3d_3\cos\alpha_1^2\\
&-4d_2d_4\cos\alpha_2\cos\alpha_3a_3^2\cos\alpha_1^2+4d_2d_3^3\cos\alpha_2+4a_1^2d_4\cos\alpha_3d_3\cos\alpha_1^2\\
&-4d_2d_3\cos\alpha_2a_2^2\cos\alpha_1^2+8a_3a_2d_4\cos\alpha_3d_3\cos\alpha_1^2-4d_3^2d_2d_4\sin\alpha_3\sin\alpha_2\cos\alpha_1^2\\
&-4r^2d_2d_4\sin\alpha_3\sin\alpha_2+4a_2^2d_2d_4\cos\alpha_2\cos\alpha_3+4d_2d_4\cos\alpha_2\cos\alpha_3a_3^2\\
&-8a_3a_2d_2d_4\cos\alpha_2\cos\alpha_3-8a_3a_2d_4\cos\alpha_3d_3+8a_1^2d_4^2\sin\alpha_3\cos\alpha_2\sin\alpha_2\cos\alpha_3\\
&-4d_4\cos\alpha_3d_3r^2-12d_2d_3^2\cos\alpha_2d_4\cos\alpha_3\cos\alpha_1^2-a_2^4\cos\alpha_1^2\\
&-4d_2d_3\cos\alpha_2d_4^2\cos\alpha_1^2+4d_2d_4\cos\alpha_2\cos\alpha_3r^2\cos\alpha_1^2+4a_3a_2a_1^2-4a_3a_2d_2^2\\
&-4a_3a_2d_4^2-4a_3a_2d_3^2+6a_3^2a_2^2-4a_3a_2^3-4a_3^3a_2+2a_1^2d_2^2-2a_1^2a_2^2+2a_1^2d_4^2-2a_1^2a_3^2\\
&-2a_1^2d_3^2+2d_2^2a_2^2+6d_2^2d_4^2+2d_2^2a_3^2+2d_2^2d_3^2+2a_2^2d_4^2+2a_2^2d_3^2+2d_4^2a_3^2+2d_4^2d_3^2+2a_3^2d_3^2
\end{aligned}$$

$$-2a_1^2r^2 - 2d_4^2r^2 - 2a_3^2r^2 - 2d_3^2r^2 + a_1^4 + d_2^4 + a_2^4 + d_4^4 + a_3^4 + d_3^4 + 4a_3a_2r^2$$

$$+2a_1^2d_2^2\cos\alpha_1^2 - 8a_1^2zd_2\cos\alpha_1 + 4a_1^2\cos\alpha_1^2d_4\sin\alpha_3\sin\alpha_2d_2 + 4a_1^2\cos\alpha_1^2d_3\cos\alpha_2d_2$$

$$-8a_1^2z\cos\alpha_1d_3\cos\alpha_2 + 4a_1^2\cos\alpha_1^2d_4\cos\alpha_2\cos\alpha_3d_2$$

$$-8a_1^2z\cos\alpha_1d_4\sin\alpha_3\sin\alpha_2 - 4a_2^2d_2d_4\cos\alpha_2\cos\alpha_3\cos\alpha_1^2$$

$$+4a_1^2d_2d_4\sin\alpha_3\sin\alpha_2 - 4d_2^3d_3\cos\alpha_2\cos\alpha_1^2 - 4d_4^3\cos\alpha_3d_3\cos\alpha_1^2$$

$$+4a_1^2d_2d_3\cos\alpha_2 - 4a_1^2d_4\cos\alpha_3d_3 + 4d_2^3d_4\cos\alpha_2\cos\alpha_3 + 4d_2^2d_4\cos\alpha_3d_3$$

$$+4d_2d_3\cos\alpha_2a_2^2 + 4d_2d_3\cos\alpha_2d_4^2 + 4d_2d_3\cos\alpha_2a_3^2 + 4a_2^2d_4\cos\alpha_3d_3$$

$$+8d_2^2d_4^2\cos\alpha_2^2\cos\alpha_3^2 + 4d_2d_4^3\cos\alpha_2\cos\alpha_3 - 4d_2d_3\cos\alpha_2r^2$$

$$+4d_4^3d_2\sin\alpha_3\sin\alpha_2 - 4d_2^2d_4^2\cos\alpha_3^2 - 4a_1^2d_4^2\cos\alpha_2^2 - 6a_3^2a_2^2\cos\alpha_1^2$$

$$+4d_2^3d_3\cos\alpha_2 + 4a_2^2d_2d_4\sin\alpha_3\sin\alpha_2 - 4d_2d_4\cos\alpha_2\cos\alpha_3r^2$$

$$+8d_2^2d_4^2\cos\alpha_2\cos\alpha_3\sin\alpha_3\sin\alpha_2 + 4d_3^2d_2d_4\sin\alpha_3\sin\alpha_2$$

$$+8d_4^2\cos\alpha_3d_3d_2\sin\alpha_3\sin\alpha_2 + 8a_1^2d_4^2\cos\alpha_3^2\cos\alpha_2^2$$

$$-4d_2d_3^3\cos\alpha_2\cos\alpha_1^2 - 4d_4^2\cos\alpha_3^2d_3^2\cos\alpha_1^2 - 4a_3a_2r^2\cos\alpha_1^2$$

$$-4a_3a_2a_1^2\cos\alpha_1^2 + 4a_3a_2d_2^2\cos\alpha_1^2 + 4a_3a_2d_4^2\cos\alpha_1^2 + 4a_3a_2d_3^2\cos\alpha_1^2$$

$$-4d_2^2d_3^2\cos\alpha_2^2\cos\alpha_1^2 - 4d_3^3d_4\cos\alpha_3\cos\alpha_1^2 + 4d_2^2d_4^2\cos\alpha_1^2\cos\alpha_2^2$$

$$+4d_2^2d_4^2\cos\alpha_1^2\cos\alpha_3^2 + 4d_4\cos\alpha_3d_3r^2\cos\alpha_1^2 - 8d_2d_4^2\cos\alpha_2\cos\alpha_3^2d_3\cos\alpha_1^2$$

$$-4d_2^3d_4\cos\alpha_2\cos\alpha_3\cos\alpha_1^2 - 2a_3^2d_3^2\cos\alpha_1^2 - 6d_2^2d_4^2\cos\alpha_1^2$$

$$-2d_4^2d_3^2\cos\alpha_1^2 + 2a_1^2d_3^2\cos\alpha_1^2 - 2a_2^2d_3^2\cos\alpha_1^2 + 2d_4^2r^2\cos\alpha_1^2$$

$$+2a_1^2a_3^2\cos\alpha_1^2 + 2a_1^2d_4^2\cos\alpha_1^2 - 2a_2^2d_4^2\cos\alpha_1^2 + 2d_3^2r^2\cos\alpha_1^2$$

$$+2d_2^2r^2\cos\alpha_1^2 + 2a_1^2r^2\cos\alpha_1^2 - 2d_2^2a_3^2\cos\alpha_1^2 + 2a_2^2r^2\cos\alpha_1^2$$

$$-2d_2^2a_2^2\cos\alpha_1^2 - 2d_4^2a_3^2\cos\alpha_1^2 + 4a_3a_2^3\cos\alpha_1^2 + 2a_3^2r^2\cos\alpha_1^2$$

$$-2d_2^2d_3^2\cos\alpha_1^2 + 2a_1^2a_2^2\cos\alpha_1^2 + 4a_3^3a_2\cos\alpha_1^2 - 4a_1^2d_4^2\cos\alpha_3^2$$

$$-4a_3^2d_2d_4\sin\alpha_3\sin\alpha_2\cos\alpha_1^2 + 4d_2d_3\cos\alpha_2r^2\cos\alpha_1^2 - 4a_2^2d_4\cos\alpha_3d_3\cos\alpha_1^2$$

$$
\begin{aligned}
&-8d_2^2 d_3 \cos\alpha_2^2 d_4 \cos\alpha_3 \cos\alpha_1^2 - 4d_2^3 d_4 \sin\alpha_3 \sin\alpha_2 \cos\alpha_1^2 \\
&-4d_2 d_3 \cos\alpha_2 a_3^2 \cos\alpha_1^2 + 8a_3 a_2 d_2 d_3 \cos\alpha_2 \cos\alpha_1^2 - 4a_3^2 d_4 \cos\alpha_3 d_3 \cos\alpha_1^2 \\
&+8a_3 a_2 d_2 d_4 \sin\alpha_3 \sin\alpha_2 \cos\alpha_1^2 - 4d_2 d_4^3 \cos\alpha_2 \cos\alpha_3 \cos\alpha_1^2 \\
&-8d_2^2 d_3 \cos\alpha_2 d_4 \sin\alpha_3 \sin\alpha_2 \cos\alpha_1^2 + 8a_3 a_2 d_2 d_4 \cos\alpha_2 \cos\alpha_3 \cos\alpha_1^2 \\
&-8d_2^2 d_4^2 \cos\alpha_2 \cos\alpha_3 \sin\alpha_3 \sin\alpha_2 \cos\alpha_1^2 \\
&-8d_4^2 \cos\alpha_3 d_3 d_2 \sin\alpha_3 \sin\alpha_2 \cos\alpha_1^2 + 8a_1^2 d_4 \sin\alpha_3 \cos\alpha_2 d_3 \sin\alpha_2) u_3^4 + \qquad \text{(A.17)}
\end{aligned}
$$

$$
\begin{aligned}
&(16d_2^2 a_3 \sin\alpha_2 d_4 \cos\alpha_2 \cos\alpha_3 + 16d_2 d_3 \cos\alpha_2 d_4 \sin\alpha_3 a_2 \\
&-16d_4^2 \sin\alpha_3 a_2 \cos\alpha_3 d_3 \cos\alpha_1^2 + 16a_1^2 a_3 \cos\alpha_2 d_4 \sin\alpha_2 \cos\alpha_3 \\
&+16d_4^2 a_2 d_2 \sin\alpha_2 \cos\alpha_1^2 \cos\alpha_3^2 + 16d_2^2 a_3 d_4 \sin\alpha_3 \cos\alpha_1^2 \cos\alpha_2^2 \\
&-16d_2^2 d_3 \cos\alpha_2 a_3 \sin\alpha_2 \cos\alpha_1^2 + 16d_4^2 \sin\alpha_3 a_2 d_2 \cos\alpha_2 \cos\alpha \\
&+8d_2 a_3 \sin\alpha_2 d_4^2 + 16d_2 a_3 \sin\alpha_2 d_4 \cos\alpha_3 d_3 + 16a_1^2 a_3 d_4 \sin\alpha_3 \\
&-16a_3 a_2^2 d_4 \sin\alpha_3 - 16a_3^2 a_2 d_2 \sin\alpha_2 + 8d_2 a_3 \sin\alpha_2 a_2 \\
&-8d_4 \sin\alpha_3 a_2^3 \cos\alpha_1^2 - 8a_1^2 d_4 \sin\alpha_3 a_2 + 8a_1^2 d_2 a_3 \sin\alpha_2 \\
&-8d_2 a_3 \sin\alpha_2 r^2 - 8d_2 a_3^3 \sin\alpha_2 \cos\alpha_1^2 + 16d_2^2 a_3 d_4 \sin\alpha_3 \\
&+8d_4 \sin\alpha_3 a_2 a_3^2 + 8d_4 \sin\alpha_3 a_2 d_3^2 - 8d_4 \sin\alpha_3 a_2 r^2 \\
&-16d_2 a_3 \sin\alpha_2 d_4 \cos\alpha_3 d_3 \cos\alpha_1^2 - 16d_2^2 a_3 \sin\alpha_2 d_4 \cos\alpha_2 \cos\alpha_3 \cos\alpha_1^2 \\
&-16d_2 d_3 \cos\alpha_2 d_4 \sin\alpha_3 a_2 \cos\alpha_1^2 - 16d_4^2 \sin\alpha_3 a_2 d_2 \cos\alpha_2 \cos\alpha_3 \cos\alpha_1^2 \\
&+8d_2 a_3 \sin\alpha_2 r^2 \cos\alpha_1^2 - 8d_2 a_3 \sin\alpha_2 a_2^2 \cos\alpha_1^2 + 8d_4 \sin\alpha_3 a_2 r^2 \cos\alpha_1^2 \\
&+16a_1^2 a_3 \cos\alpha_2 d_3 \sin\alpha_2 - 8d_4 \sin\alpha_3 a_2 a_3^2 \cos\alpha_1^2 - 8d_4 \sin\alpha_3 a_2 d_3^2 \cos\alpha_1^2 \\
&+16d_4^2 \sin\alpha_3 a_2 \cos\alpha_3 d_3 - 8d_2^2 d_4 \sin\alpha_3 a_2 \cos\alpha_1^2 - 16d_4^2 a_2 d_2 \sin\alpha_2 \cos\alpha_3^2 \\
&+8a_1^2 \cos\alpha_1^2 a_3 \sin\alpha_2 d_2 - 16a_1^2 z \cos\alpha_1 a_3 \sin\alpha_2 + 16d_2^2 d_3 \cos\alpha_2 a_3 \sin\alpha_2 \\
&-16d_4^2 a_2 d_2 \sin\alpha_2 \cos\alpha_1^2 - 8d_2 a_3 \sin\alpha_2 d_4^2 \cos\alpha_1^2 - 16d_2^2 a_3 d_4 \sin\alpha_3 \cos\alpha_1^2
\end{aligned}
$$

$$
\begin{aligned}
&-16d_2^2a_3d_4\sin\alpha_3\cos\alpha_2^2 - 8d_2a_3\sin\alpha_2d_3^2\cos\alpha_1^2 + 16a_3^2a_2d_2\sin\alpha_2\cos\alpha_1^2 \\
&+8a_1^2d_4\sin\alpha_3a_2\cos\alpha_1^2 + 16a_3a_2^2d_4\sin\alpha_3\cos\alpha_1^2 \\
&-16a_1^2a_3\cos\alpha_2^2d_4\sin\alpha_3 + 16d_4^2a_2d_2\sin\alpha_2 - 8d_4^3\sin\alpha_3a_2\cos\alpha_1^2 \\
&+8d_2a_3\sin\alpha_2d_3^2 + 8d_2^2d_4\sin\alpha_3a_2 + 8d_2a_3^3\sin\alpha_2 + 8d_4\sin\alpha_3a_2^3 \\
&+8d_4^3\sin\alpha_3a_2 + 8d_2^3a_3\sin\alpha_2 - 8d_2^3a_3\sin\alpha_2\cos\alpha_1^2)u_3^3 + \qquad \text{(A.18)}
\end{aligned}
$$

$$
\begin{aligned}
&(-16a_1^2z\cos\alpha_1d_4\cos\alpha_2\cos\alpha_3 + 16d_2^2d_3\cos\alpha_2^2d_4\cos\alpha_3 \\
&+8d_4^3\cos\alpha_3d_3 + 8a_3^2d_4\cos\alpha_3d_3 + 8a_1^2z^2 + 8a_1^2d_3^2\cos\alpha_2^2 \\
&+48a_3a_2d_2d_4\sin\alpha_3\sin\alpha_2 + 2r^4 + 8d_4^2\cos\alpha_3^2d_3^2 + 8a_1^2d_2d_4\cos\alpha_2\cos\alpha_3 \\
&-2a_1^4\cos\alpha_1^2 - 4d_2^2r^2 - 2d_4^4\cos\alpha_1^2 - 4a_2^2r^2 + 24d_2d_3^2\cos\alpha_2d_4\cos\alpha_3 \\
&+16a_1^2d_4\cos\alpha_3d_3\cos\alpha_2^2 + 8d_3^3d_4\cos\alpha_3 - 2r^4\cos\alpha_1^2 - 2d_2^4\cos\alpha_1^2 \\
&-2d_3^4\cos\alpha_1^2 + 8d_2^2d_3^2\cos\alpha_2^2 + 8d_2^2d_4^2\cos\alpha_2^2 - 2a_3^4\cos\alpha_1^2 \\
&+16d_2d_4^2\cos\alpha_2\cos\alpha_3^2d_3 - 8d_2^2d_4\cos\alpha_3d_3\cos\alpha_1^2 - 8d_2d_4\cos\alpha_2\cos\alpha_3a_3^2\cos\alpha_1^2 \\
&+8d_2d_3^3\cos\alpha_2 + 8a_1^2d_4\cos\alpha_3d_3\cos\alpha_1^2 - 8d_2d_3\cos\alpha_2a_2^2\cos\alpha_1^2 \\
&+8a_2^2d_2d_4\cos\alpha_2\cos\alpha_3 + 8d_2d_4\cos\alpha_2\cos\alpha_3a_3^2 - 8d_4\cos\alpha_3d_3r^2 \\
&-24d_2d_3^2\cos\alpha_2d_4\cos\alpha_3\cos\alpha_1^2 - 2a_2^4\cos\alpha_1^2 - 8d_2d_3\cos\alpha_2d_4^2\cos\alpha_1^2 \\
&+8d_2d_4\cos\alpha_2\cos\alpha_3r^2\cos\alpha_1^2 - 4a_3^2a_2^2 + 4a_1^2d_2^2 - 4a_1^2a_2^2 - 12a_1^2d_4^2 + 12a_1^2a_3^2 \\
&-4a_1^2d_3^2 + 4d_2^2a_2^2 - 4d_2^2d_4^2 + 20d_2^2a_3^2 + 4d_2^2d_3^2 + 20a_2^2d_4^2 + 4a_2^2d_3^2 + 4d_4^2a_3^2 + 4d_4^2d_3^2 \\
&+4a_3^2d_3^2 - 4a_1^2r^2 - 4d_4^2r^2 - 4a_3^2r^2 - 4d_3^2r^2 + 2a_1^4 + 2d_2^4 + 2a_2^4 + 2d_4^4 + 2a_3^4 + 2d_3^4 \\
&+16d_2^2a_3^2\cos\alpha_1^2\cos\alpha_2^2 + 4a_1^2d_2^2\cos\alpha_1^2 - 16a_1^2zd_2\cos\alpha_1 \\
&+8a_1^2\cos\alpha_1^2d_3\cos\alpha_2d_2 - 16a_1^2z\cos\alpha_1d_3\cos\alpha_2 + 8a_1^2\cos\alpha_1^2d_4\cos\alpha_2\cos\alpha_3d_2 \\
&-8a_2^2d_2d_4\cos\alpha_2\cos\alpha_3\cos\alpha_1^2 - 16a_2^2d_4^2\cos\alpha_3^2 - 16a_1^2a_3^2\cos\alpha_2^2 \\
&-16d_2^2a_3^2\cos\alpha_2^2 - 8d_2^3d_3\cos\alpha_2\cos\alpha_1^2 - 8d_4^3\cos\alpha_3d_3\cos\alpha_1^2
\end{aligned}
$$

$$
\begin{aligned}
&+8a_1^2d_2d_3\cos\alpha_2 - 8a_1^2d_4\cos\alpha_3d_3 + 8d_2^3d_4\cos\alpha_2\cos\alpha_3 + 8d_2^2d_4\cos\alpha_3d_3 \\
&+8d_2d_3\cos\alpha_2a_2^2 + 8d_2d_3\cos\alpha_2d_4^2 + 8d_2d_3\cos\alpha_2a_3^2 + 8a_2^2d_4\cos\alpha_3d_3 \\
&+8d_2d_4^3\cos\alpha_2\cos\alpha_3 - 8d_2d_3\cos\alpha_2r^2 + 8d_2^2d_4^2\cos\alpha_3^2 + 8a_1^2d_4^2\cos\alpha_2^2 \\
&+4a_3^2a_2^2\cos\alpha_1^2 + 8d_2^3d_3\cos\alpha_2 - 8d_2d_4\cos\alpha_2\cos\alpha_3r^2 - 8d_2d_3^3\cos\alpha_2\cos\alpha_1^2 \\
&-8d_4^2\cos\alpha_3^2d_3^2\cos\alpha_1^2 - 8d_2^2d_3^2\cos\alpha_2^2\cos\alpha_1^2 - 8d_3^3d_4\cos\alpha_3\cos\alpha_1^2 \\
&-8d_2^2d_4^2\cos\alpha_1^2\cos\alpha_2^2 - 8d_2^2d_4^2\cos\alpha_1^2\cos\alpha_3^2 + 8d_4\cos\alpha_3d_3r^2\cos\alpha_1^2 \\
&-16d_2d_4^2\cos\alpha_2\cos\alpha_3^2d_3\cos\alpha_1^2 - 8d_2^3d_4\cos\alpha_2\cos\alpha_3\cos\alpha_1^2 \\
&-4a_3^2d_3^2\cos\alpha_1^2 + 4d_2^2d_4^2\cos\alpha_1^2 - 4d_4^2d_3^2\cos\alpha_1^2 + 4a_1^2d_3^2\cos\alpha_1^2 \\
&-4a_2^2d_3^2\cos\alpha_1^2 + 4d_4^2r^2\cos\alpha_1^2 + 4a_1^2a_3^2\cos\alpha_1^2 + 4a_1^2d_4^2\cos\alpha_1^2 \\
&-20a_2^2d_4^2\cos\alpha_1^2 + 4d_3^2r^2\cos\alpha_1^2 + 4d_2^2r^2\cos\alpha_1^2 + 4a_1^2r^2\cos\alpha_1^2 \\
&-20d_2^2a_3^2\cos\alpha_1^2 + 4a_2^2r^2\cos\alpha_1^2 - 4d_2^2a_2^2\cos\alpha_1^2 - 4d_4^2a_3^2\cos\alpha_1^2 \\
&+4a_3^2r^2\cos\alpha_1^2 - 4d_2^2d_3^2\cos\alpha_1^2 + 4a_1^2a_2^2\cos\alpha_1^2 + 8a_1^2d_4^2\cos\alpha_3^2 \\
&+8d_2d_3\cos\alpha_2r^2\cos\alpha_1^2 - 8a_2^2d_4\cos\alpha_3d_3\cos\alpha_1^2 - 16d_2^2d_3\cos\alpha_2^2d_4\cos\alpha_3\cos\alpha_1^2 \\
&-8d_2d_3\cos\alpha_2a_3^2\cos\alpha_1^2 - 8a_3^2d_4\cos\alpha_3d_3\cos\alpha_1^2 - 48a_3a_2d_2d_4\sin\alpha_3\sin\alpha_2\cos\alpha_1^2 \\
&-8d_2d_4^3\cos\alpha_2\cos\alpha_3\cos\alpha_1^2 + 16a_2^2d_4^2\cos\alpha_1^2\cos\alpha_3^2)u_3^2 +
\end{aligned}
\tag{A.19}
$$

$$
\begin{aligned}
&(16d_2^2a_3\sin\alpha_2d_4\cos\alpha_2\cos\alpha_3 + 16d_2d_3\cos\alpha_2d_4\sin\alpha_3a_2 \\
&-16d_4^2\sin\alpha_3a_2\cos\alpha_3d_3\cos\alpha_1^2 + 16a_1^2a_3\cos\alpha_2d_4\sin\alpha_2\cos\alpha_3 \\
&-16d_4^2a_2d_2\sin\alpha_2\cos\alpha_1^2\cos\alpha_3^2 - 16d_2^2a_3d_4\sin\alpha_3\cos\alpha_1^2\cos\alpha_2^2 \\
&-16d_2^2d_3\cos\alpha_2a_3\sin\alpha_2\cos\alpha_1^2 + 16d_4^2\sin\alpha_3a_2d_2\cos\alpha_2\cos\alpha_3 \\
&+8d_2a_3\sin\alpha_2d_4^2 + 16d_2a_3\sin\alpha_2d_4\cos\alpha_3d_3 - 16a_1^2a_3d_4\sin\alpha_3 \\
&+16a_3a_2^2d_4\sin\alpha_3 + 16a_3^2a_2d_2\sin\alpha_2 + 8d_2a_3\sin\alpha_2a_2^2 - 8d_4\sin\alpha_3a_2^3\cos\alpha_1^2 \\
&-8a_1^2d_4\sin\alpha_3a_2 + 8a_1^2d_2a_3\sin\alpha_2 - 8d_2a_3\sin\alpha_2r^2 - 8d_2a_3^3\sin\alpha_2\cos\alpha_1^2
\end{aligned}
$$

$$
\begin{aligned}
&-16d_2^2a_3d_4\sin\alpha_3+8d_4\sin\alpha_3a_2a_3^2+8d_4\sin\alpha_3a_2d_3^2-8d_4\sin\alpha_3a_2r^2\\
&-16d_2a_3\sin\alpha_2d_4\cos\alpha_3d_3\cos\alpha_1^2-16d_2^2a_3\sin\alpha_2d_4\cos\alpha_2\cos\alpha_3\cos\alpha_1^2\\
&-16d_2d_3\cos\alpha_2d_4\sin\alpha_3a_2\cos\alpha_1^2-16d_4^2\sin\alpha_3a_2d_2\cos\alpha_2\cos\alpha_3\cos\alpha_1^2\\
&+8d_2a_3\sin\alpha_2r^2\cos\alpha_1^2-8d_2a_3\sin\alpha_2a_2^2\cos\alpha_1^2+8d_4\sin\alpha_3a_2r^2\cos\alpha_1^2\\
&+16a_1^2a_3\cos\alpha_2d_3\sin\alpha_2-8d_4\sin\alpha_3a_2a_3^2\cos\alpha_1^2-8d_4\sin\alpha_3a_2d_3^2\cos\alpha_1^2\\
&+16d_4^2\sin\alpha_3a_2\cos\alpha_3d_3-8d_2^2d_4\sin\alpha_3a_2\cos\alpha_1^2+16d_4^2a_2d_2\sin\alpha_2\cos\alpha_3^2\\
&+8a_1^2\cos\alpha_1^2a_3\sin\alpha_2d_2-16a_1^2z\cos\alpha_1a_3\sin\alpha_2+16d_2^2d_3\cos\alpha_2a_3\sin\alpha_2\\
&+16d_4^2a_2d_2\sin\alpha_2\cos\alpha_1^2-8d_2a_3\sin\alpha_2d_4^2\cos\alpha_1^2+16d_2^2a_3d_4\sin\alpha_3\cos\alpha_1^2\\
&+16d_2^2a_3d_4\sin\alpha_3\cos\alpha_2^2-8d_2a_3\sin\alpha_2d_3^2\cos\alpha_1^2-16a_3^2a_2d_2\sin\alpha_2\cos\alpha_1^2\\
&+8a_1^2d_4\sin\alpha_3a_2\cos\alpha_1^2-16a_3a_2^2d_4\sin\alpha_3\cos\alpha_1^2+16a_1^2a_3\cos\alpha_2^2d_4\sin\alpha_3\\
&-16d_4^2a_2d_2\sin\alpha_2-8d_4^3\sin\alpha_3a_2\cos\alpha_1^2+8d_2a_3\sin\alpha_2d_3^2+8d_2^2d_4\sin\alpha_3a_2\\
&+8d_2a_3^3\sin\alpha_2+8d_4\sin\alpha_3a_2^3+8d_4^3\sin\alpha_3a_2+8d_2^3a_3\sin\alpha_2\\
&-8d_2^3a_3\sin\alpha_2\cos\alpha_1^2)u_3
\end{aligned}
\tag{A.20}
$$

$$
\begin{aligned}
&-8a_1^2z\cos\alpha_1d_4\cos\alpha_2\cos\alpha_3+8d_2^2d_3\cos\alpha_2^2d_4\cos\alpha_3\\
&-8d_2^2d_4^2\cos\alpha_2^2\cos\alpha_3^2\cos\alpha_1^2+4d_4^3\cos\alpha_3d_3+4a_3^2d_4\cos\alpha_3d_3\\
&+4a_1^2z^2+4a_1^2d_3^2\cos\alpha_2^2+4a_2^2d_2d_4\sin\alpha_3\sin\alpha_2\cos\alpha_1^2\\
&-8a_3a_2d_2d_4\sin\alpha_3\sin\alpha_2+4d_4^3d_2\sin\alpha_3\sin\alpha_2\cos\alpha_1^2\\
&-4r^2d_2d_4\sin\alpha_3\sin\alpha_2\cos\alpha_1^2+r^4+4d_4^2\cos\alpha_3^2d_3^2\\
&+4a_1^2d_2d_4\cos\alpha_2\cos\alpha_3-4a_3^2d_2d_4\sin\alpha_3\sin\alpha_2-a_1^4\cos\alpha_1^2\\
&-2d_2^2r^2-d_4^4\cos\alpha_1^2-2a_2^2r^2+12d_2d_3^2\cos\alpha_2d_4\cos\alpha_3\\
&+8a_1^2d_4\cos\alpha_3d_3\cos\alpha_2^2+4d_3^3d_4\cos\alpha_3-r^4\cos\alpha_1^2\\
&-8d_2^2d_3\cos\alpha_2d_4\sin\alpha_3\sin\alpha_2+8a_3a_2d_2d_3\cos\alpha_2-d_2^4\cos\alpha_1^2
\end{aligned}
$$

$$-d_3^4 \cos\alpha_1^2 + 4d_2^2 d_3^2 \cos\alpha_2^2 - 4d_2^3 d_4 \sin\alpha_3 \sin\alpha_2 - 4d_2^2 d_4^2 \cos\alpha_2^2$$

$$-a_3^4 \cos\alpha_1^2 + 8d_2 d_4^2 \cos\alpha_2 \cos\alpha_3^2 d_3 - 4d_2^2 d_4 \cos\alpha_3 d_3 \cos\alpha_1^2$$

$$-4d_2 d_4 \cos\alpha_2 \cos\alpha_3 a_3^2 \cos\alpha_1^2 + 4d_2 d_3^3 \cos\alpha_2 + 4a_1^2 d_4 \cos\alpha_3 d_3 \cos\alpha_1^2$$

$$-4d_2 d_3 \cos\alpha_2 a_2^2 \cos\alpha_1^2 - 8a_3 a_2 d_4 \cos\alpha_3 d_3 \cos\alpha_1^2$$

$$+4d_3^2 d_2 d_4 \sin\alpha_3 \sin\alpha_2 \cos\alpha_1^2 + 4r^2 d_2 d_4 \sin\alpha_3 \sin\alpha_2$$

$$+4a_2^2 d_2 d_4 \cos\alpha_2 \cos\alpha_3 + 4d_2 d_4 \cos\alpha_2 \cos\alpha_3 a_3^2 + 8a_3 a_2 d_2 d_4 \cos\alpha_2 \cos\alpha_3$$

$$+8a_3 a_2 d_4 \cos\alpha_3 d_3 - 8a_1^2 d_4^2 \sin\alpha_3 \cos\alpha_2 \sin\alpha_2 \cos\alpha_3$$

$$-4d_4 \cos\alpha_3 d_3 r^2 - 12 d_2 d_3^2 \cos\alpha_2 d_4 \cos\alpha_3 \cos\alpha_1^2 - a_2^4 \cos\alpha_1^2$$

$$-4d_2 d_3 \cos\alpha_2 d_4^2 \cos\alpha_1^2 + 4d_2 d_4 \cos\alpha_2 \cos\alpha_3 r^2 \cos\alpha_1^2$$

$$-4a_3 a_2 a_1^2 + 4a_3 a_2 d_2^2 + 4a_3 a_2 d_4^2 + 4a_3 a_2 d_3^2 + 6a_3^2 a_2^2 + 4a_3 a_2^3 + 4a_3^3 a_2 + 2a_1^2 d_2^2$$

$$-2a_1^2 a_2^2 + 2a_1^2 d_4^2 - 2a_1^2 a_3^2 - 2a_1^2 d_3^2 + 2d_2^2 a_2^2 + 6d_2^2 d_4^2 + 2d_2^2 a_3^2 + 2d_2^2 d_3^2$$

$$+2a_2^2 d_4^2 + 2a_2^2 d_3^2 + 2d_4^2 a_3^2 + 2d_4^2 d_3^2 + 2a_3^2 d_3^2 - 2a_1^2 r^2 - 2d_4^2 r^2 - 2a_3^2 r^2$$

$$-2d_3^2 r^2 + a_1^4 + d_2^4 + a_2^4 + d_4^4 + a_3^4 + d_3^4 - 4a_3 a_2 r^2 + 2a_1^2 d_2^2 \cos\alpha_1^2 - 8a_1^2 z d_2 \cos\alpha_1$$

$$-4a_1^2 \cos\alpha_1^2 d_4 \sin\alpha_3 \sin\alpha_2 d_2 + 4a_1^2 \cos\alpha_1^2 d_3 \cos\alpha_2 d_2$$

$$-8a_1^2 z \cos\alpha_1 d_3 \cos\alpha_2 + 4a_1^2 \cos\alpha_1^2 d_4 \cos\alpha_2 \cos\alpha_3 d_2$$

$$+8a_1^2 z \cos\alpha_1 d_4 \sin\alpha_3 \sin\alpha_2 - 4a_2^2 d_2 d_4 \cos\alpha_2 \cos\alpha_3 \cos\alpha_1^2$$

$$-4a_1^2 d_2 d_4 \sin\alpha_3 \sin\alpha_2 - 4d_2^3 d_3 \cos\alpha_2 \cos\alpha_1^2 - 4d_4^3 \cos\alpha_3 d_3 \cos\alpha_1^2$$

$$+4a_1^2 d_2 d_3 \cos\alpha_2 - 4a_1^2 d_4 \cos\alpha_3 d_3 + 4d_2^3 d_4 \cos\alpha_2 \cos\alpha_3 + 4d_2^2 d_4 \cos\alpha_3 d_3$$

$$+4d_2 d_3 \cos\alpha_2 a_2^2 + 4d_2 d_3 \cos\alpha_2 d_4^2 + 4d_2 d_3 \cos\alpha_2 a_3^2 + 4a_2^2 d_4 \cos\alpha_3 d_3$$

$$+8d_2^2 d_4^2 \cos\alpha_2^2 \cos\alpha_3^2 + 4d_2 d_4^3 \cos\alpha_2 \cos\alpha_3 - 4d_2 d_3 \cos\alpha_2 r^2$$

$$-4d_4^3 d_2 \sin\alpha_3 \sin\alpha_2 - 4d_2^2 d_4^2 \cos\alpha_3^2 - 4a_1^2 d_4^2 \cos\alpha_2^2$$

$$-6a_3^2 a_2^2 \cos\alpha_1^2 + 4d_2^3 d_3 \cos\alpha_2 - 4a_2^2 d_2 d_4 \sin\alpha_3 \sin\alpha_2$$

$$-4d_2 d_4 \cos\alpha_2 \cos\alpha_3 r^2 - 8d_2^2 d_4^2 \cos\alpha_2 \cos\alpha_3 \sin\alpha_3 \sin\alpha_2$$

$$
\begin{aligned}
&-4d_3^2d_2d_4\sin\alpha_3\sin\alpha_2 - 8d_4^2\cos\alpha_3d_3d_2\sin\alpha_3\sin\alpha_2 \\
&+8a_1^2d_4^2\cos\alpha_3^2\cos\alpha_2^2 - 4d_2d_3^3\cos\alpha_2\cos\alpha_1^2 \\
&-4d_4^2\cos\alpha_3^2d_3^2\cos\alpha_1^2 + 4a_3a_2r^2\cos\alpha_1^2 + 4a_3a_2a_1^2\cos\alpha_1^2 \\
&-4a_3a_2d_2^2\cos\alpha_1^2 - 4a_3a_2d_4^2\cos\alpha_1^2 - 4a_3a_2d_3^2\cos\alpha_1^2 \\
&-4d_2^2d_3^2\cos\alpha_2^2\cos\alpha_1^2 - 4d_3^3d_4\cos\alpha_3\cos\alpha_1^2 \\
&+4d_2^2d_4^2\cos\alpha_1^2\cos\alpha_2^2 + 4d_2^2d_4^2\cos\alpha_1^2\cos\alpha_3^2 \\
&+4d_4\cos\alpha_3d_3r^2\cos\alpha_1^2 - 8d_2d_4^2\cos\alpha_2\cos\alpha_3^2d_3\cos\alpha_1^2 \\
&-4d_2^3d_4\cos\alpha_2\cos\alpha_3\cos\alpha_1^2 - 2a_3^2d_3^2\cos\alpha_1^2 - 6d_2^2d_4^2\cos\alpha_1^2 \\
&-2d_4^2d_3^2\cos\alpha_1^2 + 2a_1^2d_3^2\cos\alpha_1^2 - 2a_2^2d_3^2\cos\alpha_1^2 + 2d_4^2r^2\cos\alpha_1^2 \\
&+2a_1^2a_3^2\cos\alpha_1^2 + 2a_1^2d_4^2\cos\alpha_1^2 - 2a_2^2d_4^2\cos\alpha_1^2 + 2d_3^2r^2\cos\alpha_1^2 \\
&+2d_2^2r^2\cos\alpha_1^2 + 2a_1^2r^2\cos\alpha_1^2 - 2d_2^2a_3^2\cos\alpha_1^2 + 2a_2^2r^2\cos\alpha_1^2 \\
&-2d_2^2a_2^2\cos\alpha_1^2 - 2d_4^2a_3^2\cos\alpha_1^2 - 4a_3a_2^3\cos\alpha_1^2 + 2a_3^2r^2\cos\alpha_1^2 \\
&-2d_2^2d_3^2\cos\alpha_1^2 + 2a_1^2a_2^2\cos\alpha_1^2 - 4a_3^3a_2\cos\alpha_1^2 - 4a_1^2d_4^2\cos\alpha_3^2 \\
&+4a_3^2d_2d_4\sin\alpha_3\sin\alpha_2\cos\alpha_1^2 + 4d_2d_3\cos\alpha_2r^2\cos\alpha_1^2 \\
&-4a_2^2d_4\cos\alpha_3d_3\cos\alpha_1^2 - 8d_2^2d_3\cos\alpha_2^2d_4\cos\alpha_3\cos\alpha_1^2 \\
&+4d_2^3d_4\sin\alpha_3\sin\alpha_2\cos\alpha_1^2 - 4d_2d_3\cos\alpha_2a_3^2\cos\alpha_1^2 \\
&-8a_3a_2d_2d_3\cos\alpha_2\cos\alpha_1^2 - 4a_3^2d_4\cos\alpha_3d_3\cos\alpha_1^2 \\
&+8a_3a_2d_2d_4\sin\alpha_3\sin\alpha_2\cos\alpha_1^2 - 4d_2d_4^3\cos\alpha_2\cos\alpha_3\cos\alpha_1^2 \\
&+8d_2^2d_3\cos\alpha_2d_4\sin\alpha_3\sin\alpha_2\cos\alpha_1^2 - 8a_3a_2d_2d_4\cos\alpha_2\cos\alpha_3\cos\alpha_1^2 \\
&+8d_2^2d_4^2\cos\alpha_2\cos\alpha_3\sin\alpha_3\sin\alpha_2\cos\alpha_1^2 \\
&+8d_4^2\cos\alpha_3d_3d_2\sin\alpha_3\sin\alpha_2\cos\alpha_1^2 - 8a_1^2d_4\sin\alpha_3\cos\alpha_2d_3\sin\alpha_2 = 0. \qquad \text{(A.21)}
\end{aligned}
$$

In the solution of θ_2, either Equation 2.40 or Equation 2.41 is used. For Case 1, Equation 2.41 is used to solve for θ_2, and for Case 2, Equation 2.40. For Case 3, either

can be used. So, the solution of Equation 2.40 corresponds to the quadratic,

$$
\begin{aligned}
&(-r^2 + a_3^2 + a_1^2 + d_2^2 + d_4^2 + a_2^2 + d_3^2 - 2a_1a_3\cos\theta_3 + 2a_2a_3\cos\theta_3 + 2d_2d_3\cos\alpha_2 \\
&+2d_3d_4\cos\alpha_3 - 2a_1a_2 + 2a_2d_4\sin\alpha_3\sin\theta_3 + 2d_2a_3\sin\alpha_2\sin\theta_3 \\
&+2d_2d_4\cos\alpha_2\cos\alpha 3 - 2a_1d_4\sin\alpha_3\sin\theta_3 - 2d_2d_4\sin\alpha_3\sin\alpha_2\cos\theta 3)u^2 \\
&+(4a_1d_3\sin\alpha_2 - 4a_1a_3\cos\alpha_2\sin\theta_3 + 4a_1d_4\sin\alpha_3\cos\alpha_2\cos\theta_3 \\
&+4a_1d_4\sin\alpha_2\cos\alpha_3)u + 2d_2d_3\cos\alpha_2 + 2d_3d_4\cos\alpha_3 \\
&+2a_2a_3\cos\theta_3 - r^2 + 2a_1a_3\cos\theta_3 + 2a_1a_2 + 2a_1d_4\sin\alpha_3\sin\theta_3 + a_3^2 \\
&+a_1^2 + d_2^2 + d_4^2 + 2a_2d_4\sin\alpha_3\sin\theta_3 + 2d_2a_3\sin\alpha_2\sin\theta_3 \\
&+2d_2d_4\cos\alpha_2\cos\alpha_3 + a_2^2 - 2d_2d_4\sin\alpha_3\sin\alpha_2\cos\theta_3 + d_3^2 = 0,
\end{aligned}
\tag{A.22}
$$

and the solution of Equation 2.41 corresponds to the quadratic,

$$
\begin{aligned}
&(-r^2 + a_3^2 + a_1^2 + d_2^2 + d_4^2 + a_2^2 + d_3^2 - 2a_1a_3\cos\theta_3 + 2a_3\cos\theta_3a_2 + 2d_2d_3\cos\alpha_2 \\
&+2d_4\cos\alpha_3d_3 - 2a_1a_2 + 2d_4\sin\alpha_3\sin\theta_3a_2 + 2d_2a_3\sin\alpha_2\sin\theta_3 \\
&+2d_2d_4\cos\alpha_2\cos\alpha_3 - 2a_1d_4\sin\alpha_3\sin\theta_3 - 2d_2d_4\sin\alpha_3\sin\alpha_2\cos\theta_3)u^2 \\
&+(4a_1d_3\sin\alpha_2 - 4a_1a_3\cos\alpha_2\sin\theta_3 + 4a_1d_4\sin\alpha_3\cos\alpha_2\cos\theta_3 \\
&+4a_1d_4\sin\alpha_2\cos\alpha_3)u + 2d_2d_3\cos\alpha_2 + 2d_4\cos\alpha_3d_3 + 2a_3\cos\theta_3a_2 \\
&-r^2 + 2a_1a_3\cos\theta_3 + 2a_1a_2 + 2a_1d_4\sin\alpha_3\sin\theta_3 + a_3^2 + a_1^2 + d_2^2 + d_4 \\
&+2d_4\sin\alpha_3\sin\theta_3a_2 + 2d_2a_3\sin\alpha_2\sin\theta_3 + 2d_2d_4\cos\alpha_2\cos\alpha_3 \\
&+a_2^2 - 2d_2d_4\sin\alpha_3\sin\alpha_2\cos\theta_3 + d_3^2 = 0.
\end{aligned}
\tag{A.23}
$$

In the solution of θ_1, one equation is used for all three cases:

$$
-(\sin\theta_2d_4\sin\alpha_3\cos\alpha_2\cos\theta_3 - \cos\theta_2a_2 + \sin\theta_2a_3\cos\alpha_2\sin\theta_3 - \sin\theta_2d_3\sin\alpha_2
$$

$$
\begin{aligned}
&-\cos\theta_2 a_3\cos\theta_3-\cos\theta_2 d_4\sin\alpha_3\sin\theta_3-a_1-x-\sin\theta_2 d_4\sin\alpha_2\cos\alpha_3)u^2\\
&+(2\cos\theta_2\cos\alpha_1 d_3\sin\alpha_2-2\sin\theta_2\cos\alpha_1 d_4\sin\alpha_3\sin\theta_3\\
&-2\cos\theta_2\cos\alpha_1 a_3\cos\alpha_2\sin\theta_3+2\cos\theta_2\cos\alpha_1 d_4\sin\alpha_3\cos\alpha_2\cos\theta_3\\
&+2\sin\alpha_1 d_3\cos\alpha_2+2\sin\alpha_1 d_4\cos\alpha_2\cos\alpha_3+2\cos\theta_2\cos\alpha_1 d_4\sin\alpha_2\cos\alpha_3\\
&+2\sin\alpha_1 a_3\sin\alpha_2\sin\theta_3-2\sin\theta_2\cos\alpha_1 a_3\cos\theta_3+2d_2\sin\alpha_1\\
&-2\sin\alpha_1 d_4\sin\alpha_3\sin\alpha_2\cos\theta_3-2\sin\theta_2\cos\alpha_1 a_2)u-x+\sin\theta_2 d_3\sin\alpha_2\\
&+\cos\theta_2 a_3\cos\theta_3+a_1+\cos\theta_2 a_2+\sin\theta_2 d_4\sin\alpha_3\cos\alpha_2\cos\theta_3\\
&-\sin\theta_2 a_3\cos\alpha_2\sin\theta_3+\sin\theta_2 d_4\sin\alpha_2\cos\alpha_3+\cos\theta_2 d_4\sin\alpha_3\sin\theta_3. \qquad \text{(A.24)}
\end{aligned}
$$

As noted in Chapter 2, the last three joint angles are obtained by computing the *Z-Y-Z* Euler angles. When θ_4 is set to zero in Equation 2.45, and the first three joint angles now known, the orientation of the robot, up to its wrist, can be computed. By employing Euler angles, the difference between this matrix and the known orientation matrix of the desired pose can be used to solve for the last three joint angles.

Please refer to the Maple worksheets: *Pieper Case I*, *Pieper Case II* and *Pieper Case III* for the steps involved in obtaining these general polynomials. Currently, the m-file *RMCPieper* performs the inverse kinematics when the MDH parameters are used.

Appendix B

DH Inverse Kinematic Solution

The following inverse kinematic solution method [35] is based on successively pre-multiplying the inverses of the ${}^{i}\mathbf{T}_{i+1}$ with the ${}^{0}\mathbf{T}_{6}$ matrix. ${}^{0}\mathbf{T}_{6}$ has numerical entries, based on a desired pose, and corresponds to the forward kinematic equations. The joint angles are isolated by selecting suitable elements, where the left-hand side of the equation comprises of the numerical entries and known geometry, and the right-hand side is dependent on only one joint angle. To begin, the analytical kinematic equations,of the ${}^{0}\mathbf{T}_{6}$ matrix are determined. The solution presented here is for the A465, but the same approach is taken for the KR 15/2. The only difference is that the KR 15/2 has a slightly more complicated geometry.

The transformation matrices, based on the DH parameters of the A465, listed in Table 2.1, are as follows:

$$ {}^{0}\mathbf{T}_{1} = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{B.1} $$

$$ {}^{1}\mathbf{T}_{2} = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{B.2} $$

$$ {}^{2}\mathbf{T}_{3} = \begin{bmatrix} \cos\theta_3 & 0 & \sin\theta_3 & 0 \\ \sin\theta_3 & 0 & -\cos\theta_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{B.3} $$

$$ {}^{3}\mathbf{T}_{4} = \begin{bmatrix} \cos\theta_4 & 0 & -\sin\theta_4 & 0 \\ \sin\theta_4 & 0 & \cos\theta_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{B.4} $$

$$ {}^{4}\mathbf{T}_{5} = \begin{bmatrix} \cos\theta_5 & 0 & \sin\theta_5 & 0 \\ \sin\theta_5 & 0 & -\cos\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{B.5} $$

$$ {}^{5}\mathbf{T}_{6} = \begin{bmatrix} \cos\theta_6 & -\sin\theta_6 & 0 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{B.6} $$

The tool flange dimension, d_6, has been purposely removed so that the last frame originates at the wrist centre-point. The position of the wrist-centre is then dependent on only θ_1, θ_2, and θ_3. The orientation of the robot's hand would then be strictly dependent on the last three joint angles, θ_3, θ_5, and θ_6. The forward kinematics of the A465 can be

computed with:

$$ {}^{0}\mathbf{T}_1 \; {}^{1}\mathbf{T}_2 \; {}^{2}\mathbf{T}_3 \; {}^{3}\mathbf{T}_4 \; {}^{4}\mathbf{T}_5 \; {}^{5}\mathbf{T}_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{B.7} $$

The elements of ${}^{0}\mathbf{T}_6$ are known. Each of these elements can also be analytically obtained through the kinematic equations resulting from the matrix multiplication of the individual transformation matrices. It is a matter of isolating the joint variables, θ_i, by pre-multiplying the transformation inverses, and performing a comparison.

The inverse of the first transformation matrix is:

$$ {}^{0}\mathbf{T}_1^{-1} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ \sin\theta_1 & -\cos\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{B.8} $$

When matrix multiplied with ${}^{0}\mathbf{T}_6$, two matrices are obtained from the left and right sides of the equation. One corresponding to the ${}^{1}\mathbf{T}_6$,

$$ {}^{0}\mathbf{T}_1^{-1} \; {}^{0}\mathbf{T}_6 = {}^{0}\mathbf{T}_1^{-1} \; {}^{0}\mathbf{T}_1 \; {}^{1}\mathbf{T}_2 \; {}^{2}\mathbf{T}_3 \; {}^{3}\mathbf{T}_4 \; {}^{4}\mathbf{T}_5 \; {}^{5}\mathbf{T}_6, \tag{B.9} $$

which contains known quantities, and the other from analytical expressions dependent on

one variable:

$$ {}^{1}\mathbf{T}_6 = \begin{bmatrix} \cos\theta_1 n_x + \sin\theta_1 n_y & \cos\theta_1 o_x + \sin\theta_1 o_y & \cos\theta_1 a_x + \sin\theta_1 a_y & \cos\theta_1 p_x + \sin\theta_1 p_y \\ n_z & o_z & a_z & p_z - d_1 \\ \sin\theta_1 n_x - \cos\theta_1 n_y & \sin\theta_1 o_x - \cos\theta_1 o_y & \sin\theta_1 a_x - \cos\theta_1 a_y & \sin\theta_1 p_x - \cos\theta_1 p_y \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad \text{(B.10)} $$

Due to the size of the kinematic equations corresponding to each element, they will not be reported here. However, by finding a suitable element to exploit, a joint angle can be identified. For θ_1, element $(3,4)$ of ${}^{1}\mathbf{T}_6$ is used. Thus,

$$ \sin\theta_1 p_x - \cos\theta_1 p_y = 0, \quad \text{(B.11)} $$

which can be re-written to obtain an expression for θ_1. So,

$$ \theta_1 = atan2(p_y, p_x), \quad \text{(B.12)} $$

which is exactly as it appears as in Equation 2.48. Through successive matrix multiplications, the remainder of the joint angles can be computed.

For θ_2, the roots of a polynomial expression must be determined. After the matrix multiplication of the second inverse, elements (1,4) and (2,4) of ${}^{2}\mathbf{T}_6$ and the trigonometric identity of Equation 2.38 are used to form a transcendental equation dependent on θ_2. Once the geometric substitutions of Equation 2.44 are applied, a polynomial expression in u_2 is generated. The elements of ${}^{2}\mathbf{T}_6$ are as follows:

$$ \begin{aligned} {}^{2}\mathbf{T}_6(1,4) &= p_x \cos\theta_2 \cos\theta_1 + p_y \cos\theta_2 \sin\theta_1 + p_z \sin\theta_2 - d_1 \sin\theta_2 - a_2 \\ &= d_4 \sin\theta_3, \quad \text{(B.13)} \\ {}^{2}\mathbf{T}_6(2,4) &= -p_x \sin\theta_2 \cos\theta_1 - p_y \sin\theta_2 \sin\theta_1 + p_z \cos\theta_2 - d_1 \cos\theta_2 \end{aligned} $$

$$= -d_4 \cos\theta_3. \tag{B.14}$$

By first squaring the two equations, and then adding them together, θ_3 is eliminated from the right-hand side. Therefore, the transcendental equation is as follows:

$$\begin{aligned}&(p_x^2 \cos\theta_1^2 + 2p_x p_y \cos\theta_1 \sin\theta_1 - 2p_x a_2 \cos\theta_1 \cos\theta_2 + p_z^2 + p_y^2 - 2p_z d_1 - p_y^2 \cos\theta_1^2 \\ &-2p_y a_2 \sin\theta_1 \cos\theta_2 - 2p_z a_2 \sin\theta_2 + 2a_2 d_1 \sin\theta_2 + d_1^2 + a_2^2 - d_4^2)/d_4^2 = 0.\end{aligned} \tag{B.15}$$

The solution of θ_2 is then carried out with Equation 2.50, where u_2 is one of the roots of Equation 2.49. For θ_3, the (1,4) element of is used:

$$\begin{aligned}{}^3\mathbf{T}_6(1,4) &= p_x \cos\theta_3 \cos\theta_2 \cos\theta_1 + p_y \cos\theta_3 \cos\theta_2 \sin\theta_1 + p_z \cos\theta_3 \sin\theta_2 \\ &\quad -d_1 \cos\theta_3 \sin\theta_2 - a_2 \cos\theta_3 - p_x \sin\theta_3 \sin\theta_2 \cos\theta_1 \\ &\quad -p_y \sin\theta_3 \sin\theta_2 \sin\theta_1 + p_z \sin\theta_3 \cos\theta_2 - d_1 \sin\theta_3 \cos\theta_2 \\ &= 0.\end{aligned} \tag{B.16}$$

When rearranged, Equation 2.51 is obtained. Two of the last three joint angles, θ_4 and θ_6, can be obtained from the rotation matrix of ${}^3\mathbf{T}_6$. θ_5 is obtained from ${}^4\mathbf{T}_6$. From elements (1,3) and (2,3) of ${}^3\mathbf{T}_6$,

$$\begin{aligned}{}^3\mathbf{T}_6(1,3) &= a_x \cos\theta_3 \cos\theta_2 \cos\theta_1 + a_y \cos\theta_3 \cos\theta_2 \sin\theta_1 + a_z \cos\theta_3 \sin\theta_2 \\ &\quad -a_x \sin\theta_3 \sin\theta_2 \cos\theta_1 - a_y \sin\theta_3 \sin\theta_2 \sin\theta_1 + a_z \sin\theta_3 \cos\theta_2 \\ &= \cos\theta_4 \sin\theta_5,\end{aligned} \tag{B.17}$$

$$\begin{aligned}{}^3\mathbf{T}_6(2,3) &= a_x \sin\theta_1 - a_y \cos\theta_1 \\ &= \sin\theta_5 \sin\theta_4,\end{aligned} \tag{B.18}$$

respectively, θ_4 can be determined.

By eliminating $\sin\theta_5$, Equation 2.52 can be solved with y_4 and x_4 given in Equations 2.54 and 2.54. From (1,3) and (2,3) of ${}^4\mathbf{T}_6$,

$$\begin{aligned}
{}^4\mathbf{T}_6(1,3) &= a_x\cos\theta_4\cos\theta_3\cos\theta_2\cos\theta_1 + a_y\cos\theta_4\cos\theta_3\cos\theta_2\sin\theta_1 \\
&\quad + a_z\cos\theta_4\cos\theta_3\sin\theta_2 - a_x\cos\theta_4\sin\theta_3\sin\theta_2\cos\theta_1 \\
&\quad - a_y\cos\theta_4\sin\theta_3\sin\theta_2\sin\theta_1 + a_z\cos\theta_4\sin\theta_3\cos\theta_2 \\
&\quad + a_x\sin\theta_4\sin\theta_1 - a_y\sin\theta_4\cos\theta_1 \\
&= \sin\theta_5, \qquad \text{(B.19)} \\
{}^4\mathbf{T}_6(2,3) &= -a_x\sin\theta_3\cos\theta_2\cos\theta_1 - a_y\sin\theta_3\cos\theta_2\sin\theta_1 - a_z\sin\theta_3\sin\theta_2 \\
&\quad - a_x\cos\theta_3\sin\theta_2\cos\theta_1 - a_y\cos\theta_3\sin\theta_2\sin\theta_1 + a_z\cos\theta_3\cos\theta_2 \\
&= -\cos\theta_5, \qquad \text{(B.20)}
\end{aligned}$$

respectively, θ_5 can be identified.

Upon rearrangement, θ_5 can be solved with Equation 2.55, where y_5 and x_5 are given in Equations 2.57 and 2.57. The final angle, θ_6, is obtained from elements (3,1) and (3,2) of ${}^3\mathbf{T}_6$:

$$\begin{aligned}
{}^3\mathbf{T}_6(3,1) &= n_x\sin\theta_3\cos\theta_2\cos\theta_1 + n_y\sin\theta_3\cos\theta_2\sin\theta_1 + n_z\sin\theta_3\sin\theta_2 \\
&\quad + n_x\cos\theta_3\sin\theta_2\cos\theta_1 + n_y\cos\theta_3\sin\theta_2\sin\theta_1 - n_z\cos\theta_3\cos\theta_2 \\
&= -\cos\theta 6\sin\theta_5, \qquad \text{(B.21)} \\
{}^3\mathbf{T}_6(3,2) &= o_x\sin\theta_3\cos\theta_2\cos\theta_1 + o_y\sin\theta_3\cos\theta_2\sin\theta_1 + o_z\sin\theta_3\sin\theta_2 \\
&\quad + o_x\cos\theta_3\sin\theta_2\cos\theta_1 + o_y\cos\theta_3\sin\theta_2\sin\theta_1 - o_z\cos\theta_3\cos\theta_2 \\
&= \sin\theta 6\sin\theta_5, \qquad \text{(B.22)}
\end{aligned}$$

respectively.

By eliminating θ_5, the final angle can be computed via Equation 2.58, where y_6 and x_6 are given in Equations 2.60 and 2.60.

As with the inverse kinematic solution for the KR 15/2, this solution is only valid when using the nominal geometry of the A465. The numeric values for the link lengths and offsets can change, but not if they were initially zero.

Please refer to the Maple worksheets *Thermo CRS A465 Inverse Kinematics* and *Thermo CRS A465 Forward Kinematics and Pre-Multiplication* for the analytical solution and Matlab code generation for the inverse kinematics of the A465. *KUKA KR 15-2 Inverse Kinematics* and *KUKA KR 15-2 Forward Kinematics and Pre - Multiplication* contains the analytical solution and Matlab code generation for the inverse kinematics of the KR 15/2. In the relative simulation, the m-files *ThermoCRSA465InverseKinematics* and *KUKAKR152InverseKinematics*, provided the necessary inputs, compute the six joint angles.

Appendix C

Absolute Simulation Code

This section outlines the Matlab version 6.5 code that was developed to kinematically calibrate six degree-of-freedom serial manipulators, following the convention established in Section 3.1. Figure C.1 illustrates a map of all inputs, outputs, and subsidiary programs required by the shell program.

RMCAbsoluteSimulation.m**

- Shell program that attends to: initialization of robot geometry, specification of parameter deviations, pose generation, measurement acquisition, measurement noise, Jacobian matrix assembly, SVD, parameter identification, and the output of results.
- The program has gone through several iterations that accommodate various improvements. RMCAbsoluteSimulation06.m is the latest version.
- Optional input of joint angles from RMCDataAcquisition.m
- On-screen output of identified parameter deviations, optional figure generation illustrating measurement poses.

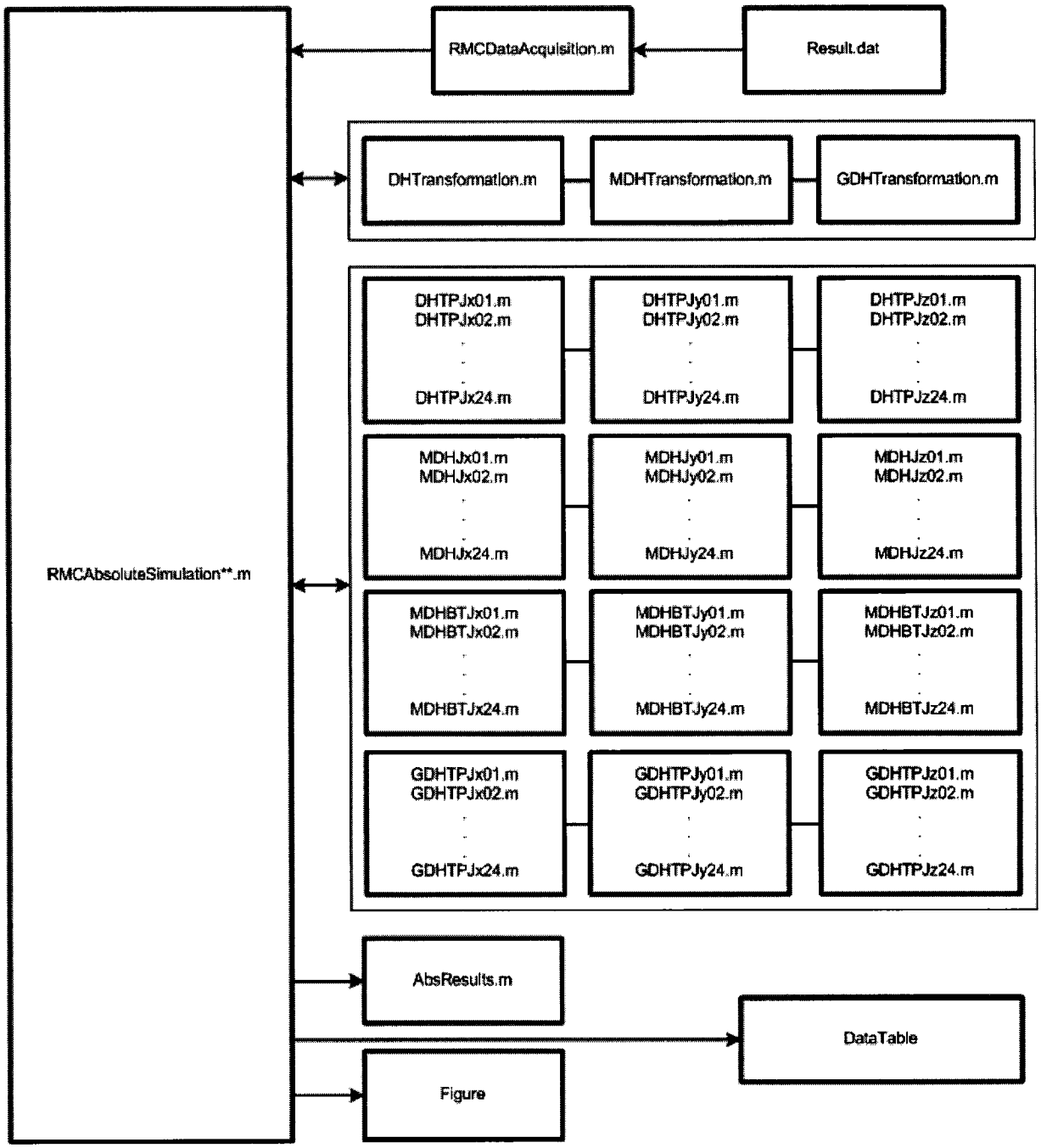

Figure C.1: Program map for the absolute simulation.

RMCDataAcquisition.m

- Provides joint angles to RMCAbsoluteSimulation**.m in a single $m \times 6$ matrix.

- Requires input file Result.dat

DHTransformation.m

- Accepts inputs θ_{i+1}, d_{i+1}, a_{i+1}, and α_{i+1} and computes the transformation matrix.
- Outputs ${}^{i}\mathbf{T}_{i+1}$, a 4×4 homogeneous transformation matrix.

MDHTransformation.m

- Accepts inputs θ_{i+1}, d_{i+1}, a_i, and α_i and computes the transformation matrix.
- Outputs ${}^{i}\mathbf{T}_{i+1}$, a 4×4 homogeneous transformation matrix.

GDHTransformation.m

- Accepts inputs θ_{i+1}, d_{i+1}, a_{i+1}, α_{i+1}, and β_{i+1} and computes the transformation matrix.
- Outputs ${}^{i}\mathbf{T}_{i+1}$, a 4×4 homogeneous transformation matrix.

DHTPJ$***$.m

- Accepts all nominal robot parameters, tool tip transformation components, and and a set of joint angles and computes the appropriate x, y, or z direction Jacobian element.
- Parameters θ_1 to θ_6 correspond to 01-06, a_1 to a_6 with 07-12, d_1 to d_6 with 13-18, and α_1 to α_6 with 19-24.
- Outputs the numeric value of the Jacobian element.

MDHJ* * *.m

- Accepts all nominal robot parameters and a set of joint angles and computes the appropriate x, y, or z direction Jacobian element.
- Parameters θ_1 to θ_6 correspond to 01-06, a_1 to a_6 with 07-12, d_0 to d_5 with 13-18, and α_0 to α_0 with 19-24.
- Outputs the numeric value of the Jacobian element.

MDHBTJ* * *.m

- Accepts all nominal robot parameters, the base and tool flange dimensions, and a set of joint angles and computes the appropriate x, y, or z direction Jacobian element.
- Parameters θ_1 to θ_6 correspond to 01-06, a_1 to a_6 with 07-12, d_0 to d_5 with 13-18, α_0 to α_0 with 19-24, d_B with 25, and $d_{T_F}$with 26.
- Outputs the numeric value of the Jacobian element.

GDHJ* * *.m

- Accepts all nominal robot parameters and a set of joint angles and computes the appropriate x, y, or z direction Jacobian element.
- Parameters θ_1 to θ_6 correspond to 01-06, a_1 to a_6 with 07-12, d_0 to d_5 with 13-18, α_0 to α_0 with 19-24, and β_3 with 25.
- Outputs the numeric value of the Jacobian element.

AbsResults.m

- Output file that contains the absolute error data.

Appendix D

Relative Simulation Code

This section outlines the Matlab version 6.5 code that was developed to kinematically calibrate six degree-of-freedom serial manipulators, using the RMC defined in Section 3.2. Figure E.1 illustrates a map of all inputs, outputs, and subsidiary programs required by the shell program.

RMCSimulation.m**

- Shell program that attends to: initialization of robot geometry, specification of parameter deviations, pose generation, measurement acquisition, Jacobian matrix assembly, SVD, parameter identification, and the output of results.
- The program has gone through several iterations that accommodate various improvements. RMCAbsoluteSimulation13.m is the latest version.
- On-screen output of identified parameter deviations.

DHTransformation.m

- Accepts inputs θ_{i+1}, d_{i+1}, a_{i+1}, and α_{i+1} and computes the transformation matrix.

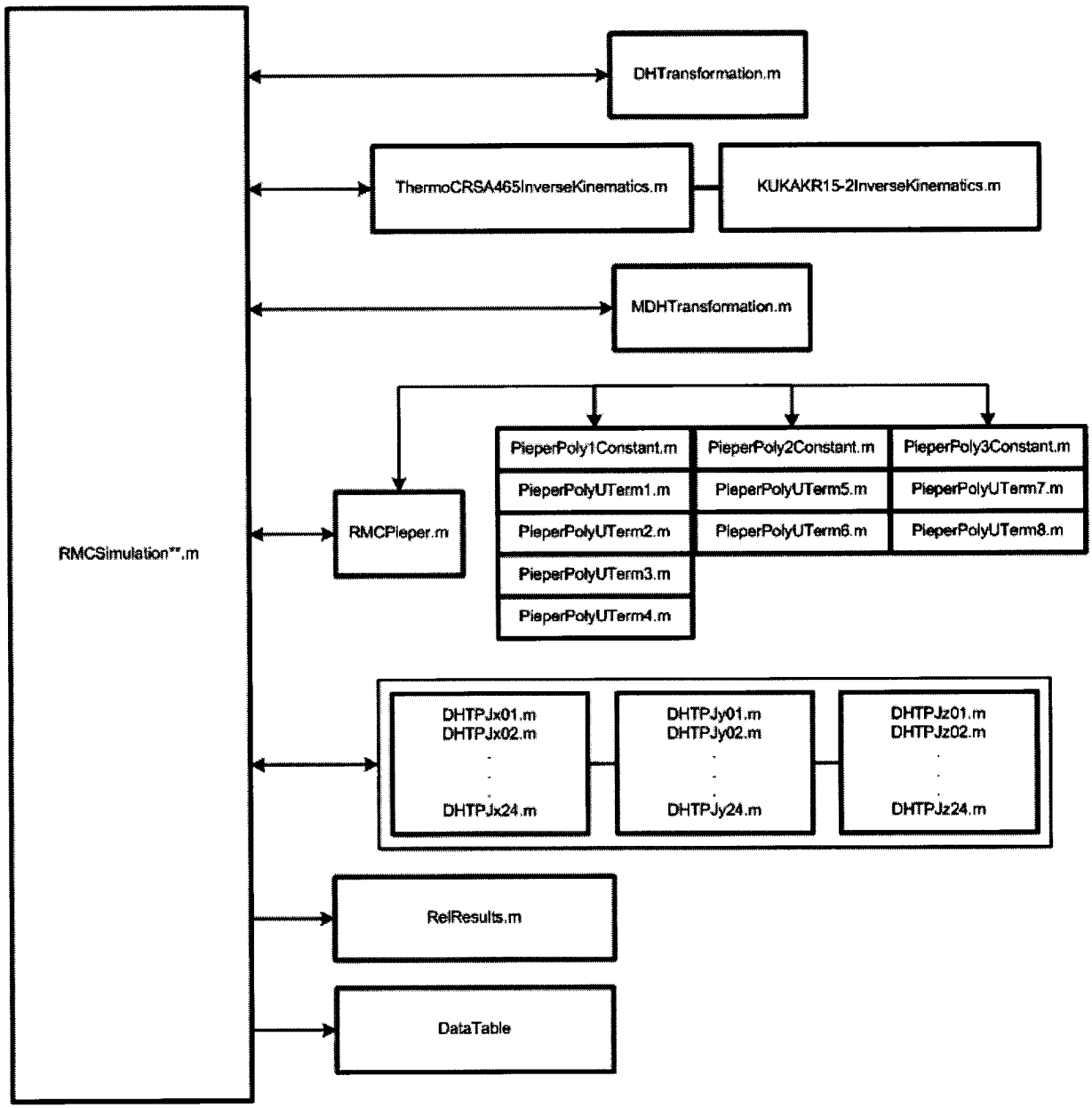

Figure D.1: Program map for the relative simulation.

- Outputs ${}^{i}\mathbf{T}_{i+1}$, a 4×4 homogeneous transformation matrix.

ThermoCRSA465InverseKinematics.m

- Accepts all nominal robot parameters, a joint angle set, and the pertinent entries of the transformation matrix for the desired pose.

- Outputs joint angle set necessary for the desired pose.

KUKAKR15-2InverseKinematics.m

- Accepts all nominal robot parameters, a joint angle set, and the pertinent entries of the transformation matrix for the desired pose.
- Outputs joint angle set necessary for the desired pose.

MDHTransformation.m

- Accepts inputs θ_{i+1}, d_{i+1}, a_i, and α_i and computes the transformation matrix.
- Outputs ${}^{i}\mathbf{T}_{i+1}$, a 4×4 homogeneous transformation matrix.

RMCPieper.m

- With earlier versions of the simulation, was used to compute the inverse kinematics for any general serial robot meeting the requirements of Pieper's method.
- Accepts specific nominal robot parameters, a joint angle set, and the pertinent entries of the transformation matrix for the desired pose.
- Outputs joint angle set necessary for the desired pose.

PieperPoly∗Constant.m

- Evaluates the value of the constant term in the polynomial expression.
- PieperPoly1Constant.m corresponds to the solution of θ_3, PieperPoly2Constant.m to θ_2, and PieperPoly3Constant.m to θ_1.

PieperPolyUTerm∗.m

- Evaluates the value of the coefficients for the powers of u in the polynomial expression.
- PieperPolyUTerm1.m to PieperPolyUTerm4.m corresponds to the solution of θ_3, PieperPolyUTerm5.m and PieperPolyUTerm6.m to θ_2, and PieperPolyUTerm7.m and PieperPolyUTerm8.m to θ_1.

DHTPJ∗ ∗ ∗.m

- Accepts all nominal robot parameters, tool tip transformation components, and and a set of joint angles and computes the appropriate x, y, or z direction Jacobian element.
- Parameters θ_1 to θ_6 correspond to 01-06, a_1 to a_6 with 07-12, d_1 to d_6 with 13-18, and α_1 to α_6 with 19-24.
- Outputs the numeric value of the Jacobian element.

RelResults.m

- Output file that contains the relative error data.

Appendix E

Experimental Calibration Code

This section outlines the Matlab version 6.5 code that was developed to kinematically calibrate the six degree-of-freedom Thermo CRS A465 serial manipulator, using the RMC defined in Section 3.2. Figure E.1 illustrates a map of all inputs, outputs, and subsidiary programs required by the shell program.

RMCExperiment01.m

- Shell program that attends to: initialization of robot geometry, forward kinematics of poses, Jacobian matrix assembly, SVD, parameter identification, and the output of results.
- Requires input of the joint angles and associated pose errors.
- On-screen output of identified parameter deviations, optional figure of relative error data.

RMCExperimentJointAngles.m

- Provides joint angles to RMCExperiment01.m in a single $m \times 6$ matrix.

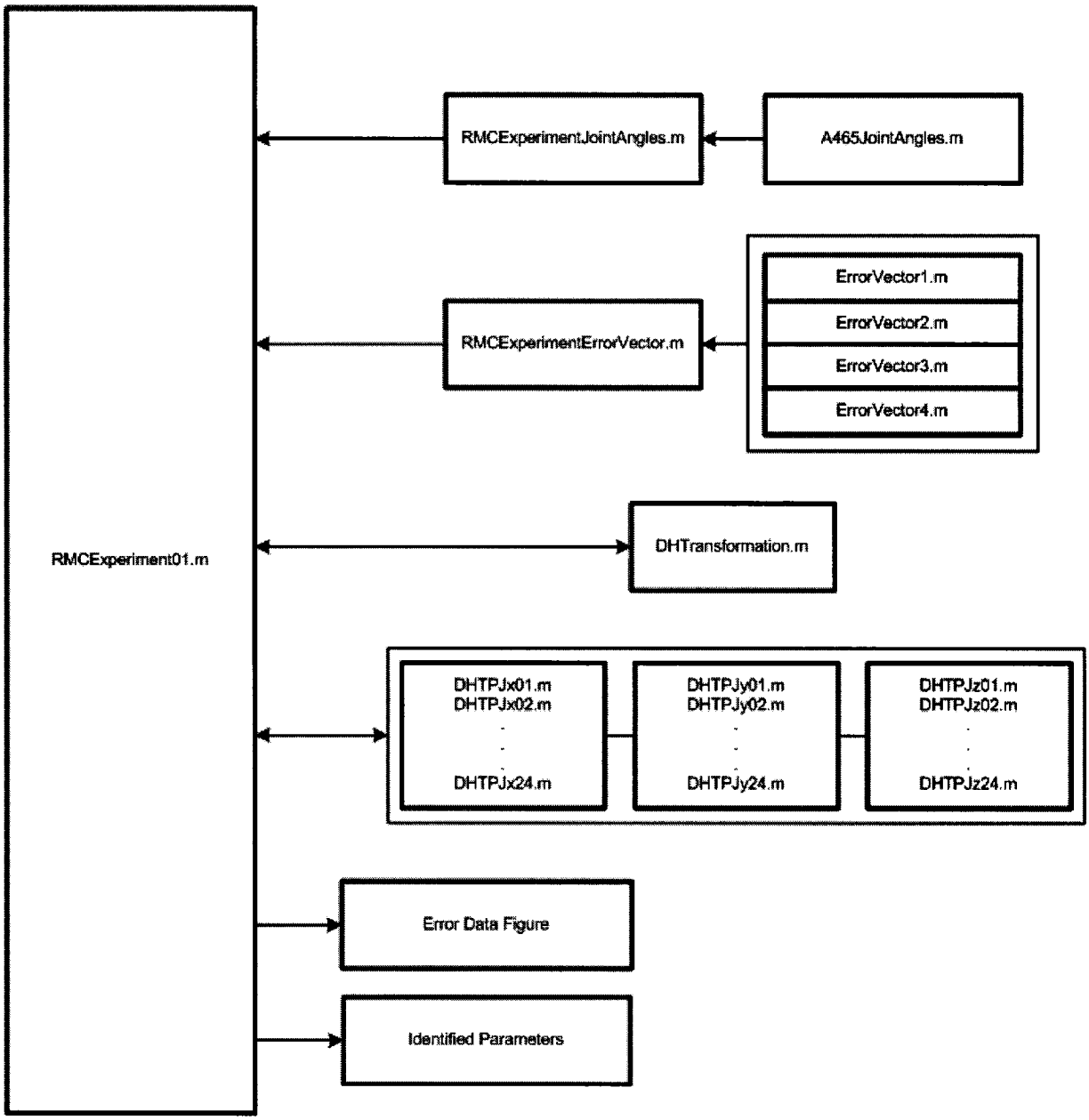

Figure E.1: Program map for experimental calibration program.

- Requires input file A465JointAngles.m

A465JointAngles.m

- Sample of output text file from the autocal.r program.

- Contains 80 joint angle sets corresponding to the poses of the first experiment.

RMCExperimentErrorVector.m

- Provides a stacked column-vector, of error data relative to the end-effector, to RMCExperiment01.m.
- Requires input file ErrorVector**.m.

ErrorVector.m**

- Contains two columns of positional error data obtained through digital image processing.
- Several files exist due to the need for the analysis of different parts of the entire data set.

DHTransformation.m

- Accepts inputs θ_{i+1}, d_{i+1}, a_{i+1}, and α_{i+1} and computes the transformation matrix.
- Outputs ${}^{i}\mathbf{T}_{i+1}$, a 4×4 homogeneous transformation matrix.

DHTPJ*.m**

- Accepts all nominal robot parameters, tool tip transformation components, and and a set of joint angles and computes the appropriate x, y, or z direction Jacobian element.
- Parameters θ_1 to θ_6 correspond to 01-06, a_1 to a_6 with 07-12, d_1 to d_6 with 13-18, and α_1 to α_6 with 19-24.
- Outputs the numeric value of the Jacobian element.

Appendix F

Automated Image Acquisition

This section outlines the the Labview Virtual Instrument and RAPL-III program that was developed to automatically acquire measurement images with a Thermo CRS A465 serial manipulator. Figure F.1 illustrates a flow diagram for the linked process. Serial communication has been established between the C500C controller and the image acquisition development computer. A message-based approach was implemented where text messages, signifying different actions, are sent through the data line. Thus far, only two codes are needed, one to signify continuance and the other to signify program completion.

As seen in Figure F.2, only a minimal amount of information is required by the virtual instrument. Essentially, which communication port to access and the location of the images. By default, the images are bitmap files named Image* * **.bmp, where a four-digit number is assigned. Both the image name and the number of digits can be altered within the program. Also, the starting place can be altered so that a measurement set can be continued.

To automatically acquire images, the virtual instrument would be activated first. The number of serial access attempts will incrementally increase over time. Next the RAPL-III program autocal.r would be started. The two will communicate via the serial connection

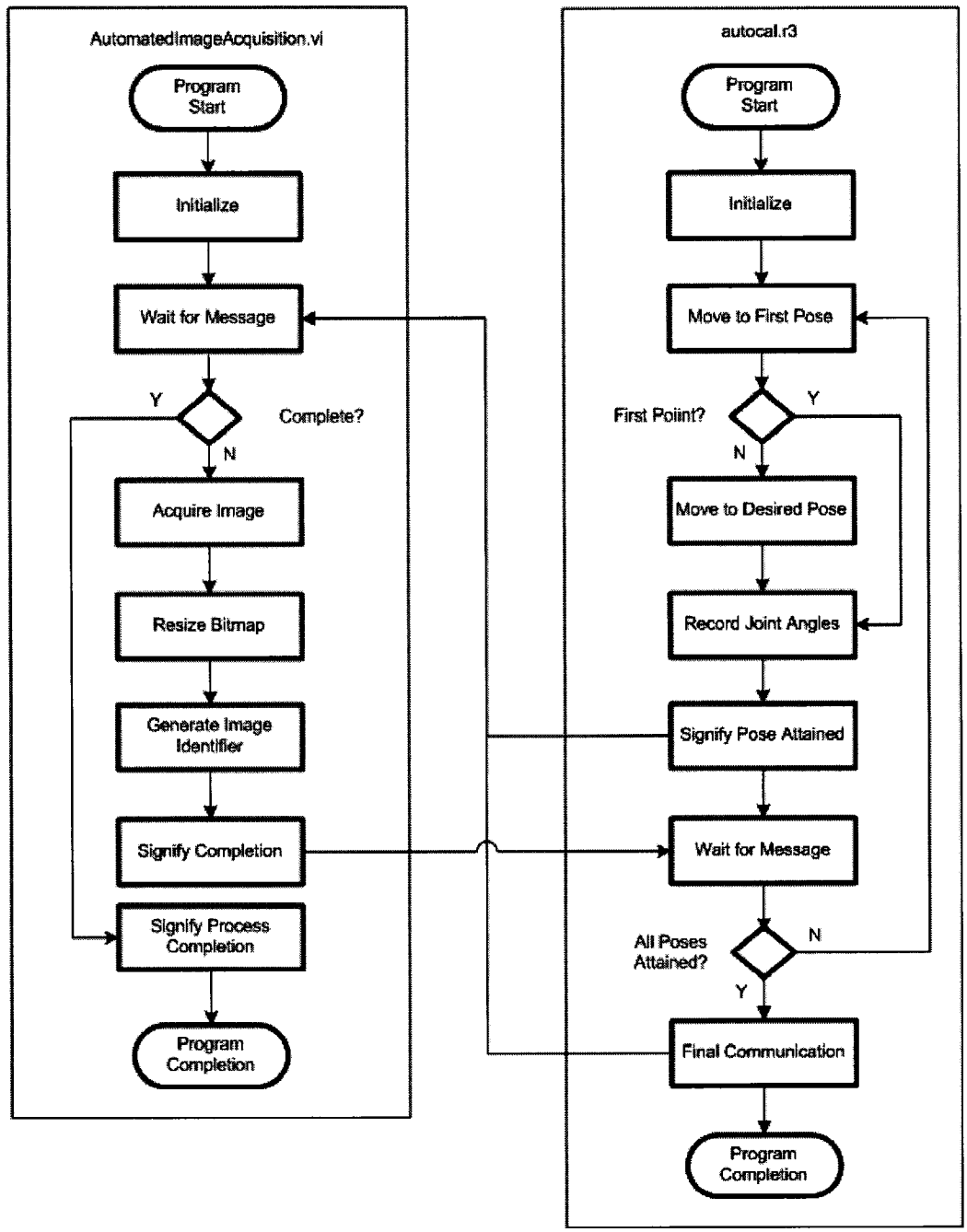

Figure F.1: Flow diagram for the automated image acquisition process.

to signify that their respective processes are complete. Once the program is complete a different code is sent through and both programs signify completion. If the process has to be stopped for any reason, the virtual instrument can just be stopped via a button in the

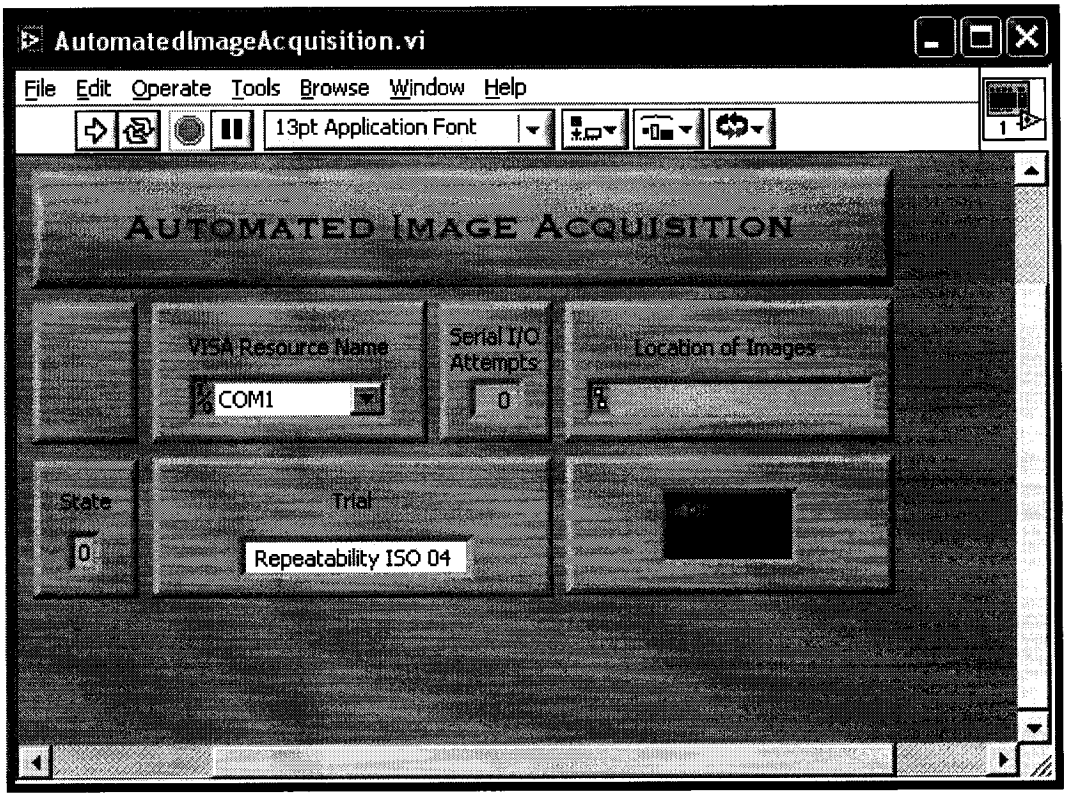

Figure F.2: Front panel for the AutomatedImageAcquisition.vi virtual instrument.

window. Unfortunately, the capability to signify a desired exit has not been implemented yet. Thus, an emergency stop must be triggered with the robotic system.

The image files are currently stored in a folder of choice within an already specified parent directory. For the joint angles, the file transfer utility of Robcomm3 must be used to removed it from the controller's memory.