Optimization of Robotic Drilling Operations by Leveraging Functional Kinematic Redundancy to Minimize Joint Reversals

by

Jasper Arthur

A thesis submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

Department of Mechanical Engineering University of Manitoba Winnipeg

© Jasper Arthur, 2021

## Abstract

Industrial robots used in manufacturing suffer from static friction (stiction) and backlash in their joints during joint reversals. The dynamic controller response to these stick-slip and dead-band phenomena leads to error at the end effector, which is especially pronounced and undesirable in precision applications such as aerospace composite drilling. The tolerance requirements for aerospace components are typically smaller than 0.2mm, which is on the boundary of what typical industrial robots can achieve. During robotic drilling operations, even small errors may result in unacceptable tolerances. For this reason, drilling using robots has not been as widely adopted in this sector. Many methods exist to optimally stiffen a robot's posture, compensate for the anticipated error, or actuate an independently stabilized drilling tool. But these methods do not address the source of the error and often do not result in satisfactory performance.

In this thesis, it is shown that by leveraging the functional kinematic redundancy inherent to drilling, the robot can reduce or even completely eliminate joint reversals while achieving the same plunge and retract motions to drill a hole. The rotation about the tool's redundant work axis is characterized at the start, target, and end positions. The parameter space is searched using Particle Swarm Optimization to converge on the best combination of input parameters which minimize reversals. The proposed methodology is applied to a KUKA KR 6 R700-2 robot with a sample drilling tool, and the performance is analyzed using internal joint position and torque measurements, as well as tool tip position. A reduction in the envelope of the drilling motion of 40% is observed, and the hysteresis commonly seen in robotic drilling motions is significantly reduced.

# **Table of Contents**

Abstra	ct	ii
List of	Гables	<b>v</b>
List of	Figures	vi
Acknow	vledgement	X
1 Int	roduction	
1.1	Background	1
1.2	Problem Statement	3
1.3	Thesis Objectives and Contributions	
2 Ba	ckground and Literature Review	5
2.1	Backlash and Stiction in Industrial Robots	5
2.2	Optimization of Robot Pose and Workpiece Placement	7
2.3	Redundancy Resolution	8
2.4	Local and Global Optimization	
2.5	Summary	
3 Ki	nematic Modeling and Motion Planning	14
3.1	Workspace Definitions	
3.2	Assumptions and Conventions	
3.3	Denavit-Hartenburg Parameter Identification	
3.4	Forward Kinematics of the KUKA KR 6 R700-2	
3.5	Inverse Kinematics of the KUKA KR 6 R700-2	
3.6	Redundancy Resolution in the Drilling Motion	
3.7	Summary	
4 Op	timization of Drilling Motion	27
4.1	Particle Swarm Optimization Algorithm	
4.2	Application of PSO for Minimizing Joint Reversals	
4.3	Objective Function and Optimization Parameters	
4.4	Simulation of a Sample Drilling Path	
4.5	Performance Tuning	
4.6	Workspace Mapping	
4.7	Parameter Mapping	
4.8	Summary	
5 Ex	perimental Validation	65
5.1	Design of Experiments and Experimental Implementation	
5.2	Repeatability Studies	
5.3	Experimental Results	
5.4	Alternate Optimization Locations	77
5.4	.1 Workspace Position 200, 400, 600 (Position 2)	77
5.4	.2 Workspace Position 200, 200, 1000 (Position 3)	
5.5	Discussion of Other Effects	
5.6	Summary	
6 Co	nclusion and Future Work	90
6.1	Conclusion	

6.2	Future Work	.91
Append	ix A	.93
Annend	ix B	.95
Rihling	ranhv	99
PIPIPIP	арлу шинин на	. , ,

# List of Tables

51
zero reversals on 53
-

# List of Figures

Figure 1. Typical 6DOF industrial manipulator robot, the KUKA KR 6 R700-2, located at the University of Manitoba Intelligent Digital Manufacturing Lab (IDML)
Figure 2. Coordinate systems within the robot workspace; start, target, and end points of a sample drilling motion, and the $\psi$ angle used to describe rotation about the drilling spindle's redundant degree of freedom. 15
Figure 3. Frame assignments to a KUKA KR 6 R700-2 according to the modified DH convention, as well as the TCP offset used for the sample drilling tool
Figure 4. Sample drilling end effector with associated coordinate frame and dimensions 22
Figure 5. Visualization of PSO behavior from initial chaotic motion, to searching the workspace, to finally converging near the global minimum
Figure 6. Sample visualization of local minimum, global minimum, and poor behavior near region of interest
Figure 7. Visualization of particle movement with different weights. A high cognitive factor $(WPB)$ will keep a particle near its personal best, while a high social factor $(WGB)$ will convince a particle to move to the swarm's best
Figure 8. Visualization of psi angle with the sample drilling tool
Figure 9. Flow chart of PSO algorithm implemented within the context of minimizing joint reversals for a drilling motion
Figure 10. Location of particles within the PSO algorithm at iterations 0, 10, 25, and 50 39
Figure 11. Score distribution of 100 particles after 50 iterations of PSO algorithm
Figure 12. Trial 1 particle convergence and performance
Figure 13. Trial 2 particle convergence and performance
Figure 14. Trial 3 particle convergence and performance
Figure 15. Trial 4 particle convergence and performance
Figure 16. Trial 5 particle convergence and performance
Figure 17. Trial 6 particle convergence and performance
Figure 18. Trial 7 particle convergence and performance
Figure 19. Trial 8 particle convergence and performance
Figure 20. Mapping of maximum plunge depth to achieve the zero-reversal case at X=500mm

Figure 22. Workspace map of maximum plunge depth capable of achieving zero reversals viewed from the underside of the XY plane
Figure 23. Side (XZ) profile of workspace map with heights/colors representing the maximum plunge depth still capable of achieving zero reversals when drilling in the positive X direction
Figure 24. Side (XZ) profile of workspace map at planes X=-100, 0, and 100, with heights/colors representing the maximum plunge depth still capable of achieving zero reversals when drilling in the positive X direction
Figure 25. Location of experimental trial positions within the robot workspace
Figure 26. Objective function map at (500, 300, 800) plotted according to start and end psi angles, with color showing the objective function value achieved
Figure 27. Representation of the objective function with the $\psi t$ parameter ±1.992 [rad].
Error! Bookmark not defined.
Figure 28. Robot at Position 1 starting location with simple drilling end effector with colored rectangular prism for rotation visualization
Figure 29. Joint torque comparison in short-term repeatability study
Figure 30. Joint torque comparison in long-term repeatability study
Figure 31. Robot at Position 1 target71
Figure 32. Drilling motion of robot at start, target, and end positions
Figure 33. Comparison of joint angles in baseline, optimized, and simulated optimized motions
Figure 34. Visualization of path distance as a function of plunge and retract distance73
Figure 35. Comparison of joint torques in the baseline and optimized motions
Figure 36. Horizontal and vertical profile comparison of TCP position between baseline and optimized motions at Position 1
Figure 37. Comparison of contour error from mean in the baseline and optimized cases 76
Figure 38. Robot at Position 2 target, [200,400,600]
Figure 39. Position 2 joint angles for the baseline and optimized motions
Figure 40. Position 2 joint torque for the baseline and optimized motions
Figure 41. Position 2 TCP cartesian position throughout baseline and optimized motions. 82
Figure 42. Robot at Position 3, [200, 200, 1000]
Figure 43. Position 3 joint angles for the baseline and optimized motions
Figure 44. Position 3 joint torque for baseline and optimized motions

Figure 45. Position 3 TCP cartesian position throughout baseline and optimized motions. 87

#### Nomenclature

# Symbols

$a_{i-1}$	DH parameter of joint i, angle
$B_{fault}$	Boolean, fault detected in joint history
$B_{jump}$	Boolean, jump detected in joint history
$B_{limit}$	Boolean, joint limit detected in joint history
С	Compliance matrix
$c_{ heta}$	Cosine of angle $\theta$
$C_{tt}$	Translational component of the compliance matrix
$d_i$	DH parameter of joint i, offset
$D_p$	Plunge depth
$D_w$	Wrist position in GCS coordinates
$EE_a$	End effector height
$EE_b$	End effector length
$f(\mathbf{x})$	Objective function
J	Jacobian matrix
k <sub>c</sub>	Compliance condition number
$k_{\theta}$	Stiffness of joint $\theta$
$K_{\theta}$	Joint stiffness matrix
N	Number of steps
$N_p$	Number of particles
N <sub>rev</sub>	Number of reversals in a drilling motion
P	Point in GCS coordinates
$p_{(x,y,z)w}$	X, Y, or Z coordinate of wrist
r	Random value between 0 and 1 in particle velocity expression
$R_{p1}$	Plunge depth lower boundary
$R_{p2}$	Plunge depth upper boundary
S	Random value between 0 and 1 in particle velocity expression
s <sub>e</sub>	Sin of angle $\theta$
$_{i}^{J}T$	Homogenous transformation matrix from 1 to j
V	Particle velocity in parameter space
V <sub>p</sub>	Plunge vector
V <sub>r</sub>	Retract vector
$V_{p+r}$	Drilling motion vector
$W_{in}$	Inertial weight in particle velocity expression
$W_{GB}$	Social factor (Global Best) in particle velocity expression
$W_{PB}$	Cognitive factor (Personal Best) in particle velocity expression

Weight of reversals in objective function
Weight of joint travel in objective function
Particle position in parameter space
DH parameter of joint i, length
Relative difference for stopping criterion
Distance along drilling path
End effector displacement
DH parameter of joint i, joint angle
Total joint travel
$\theta_3 - 90^{\circ}$
Twist about $Z_{TCS}$ at point i
Set of $\psi_s$ , $\psi_t$ , $\psi_e$ for some motion
Optimal set of psi angles for some motion

# Acronyms

APO.CDIS	Approximate Positioning – Cartesian Distance
CAD	Computer Aided Design
CNC	Computer Numerical Control
DH	Denavit-Hartenberg
DOF	Degree(s) Of Freedom
FK	Forward Kinematics
GA	Genetic Algorithm
GCS	Global Coordinate System
IDML	Intelligent Digital Manufacturing Laboratory
IK	Inverse Kinematics
KRL	KUKA Robot Language
LHS	Left Hand Side
PLA	Polylactic Acid
PSO	Particle Swarm Optimization
PTP	Point-To-Point
RHS	Right Hand Side
RPM	Revolutions Per Minute
RRT	Rapidly Exploring Random Tree
RSI	Robot Sensor Interface
SRC	Source File
Stiction	Static Friction
ТСР	Tool Center Point
TCS	Tool Coordinate System

## Acknowledgement

I would like to thank Dr. Matt Khoshdarregi, without whom I would have never discovered a passion for robotics research. His impeccable work ethic and push for excellence elevates everyone who has the fortune of calling him an instructor, research advisor, or friend.

A well-deserved gratitude is due to all of my friends at the Intelligent Digital Manufacturing Lab. Both to those who have patiently taught me, and to those whose probing questions cause me to reflect and reassess what I know. Thank you to all who have built me up, and to all who kept me grounded.

This research was financially supported by the MITACS Entrepreneurship Accelerate program as well as Natural Sciences and Engineering Research Council (NSERC) of Canada. Their support ensures students of all stripes have a chance to thrive.

Finally, the most heartfelt appreciation must be extended to my wife. She is steadfast in her support, optimistic in all her outlooks, and caring in every capacity. It is my privilege to have her by my side in the life we share together.

# **1** Introduction

Industrial robots which perform machining operations suffer from a relatively low stiffness arising from their structure and internal mechanisms. During drilling specifically, the motion commands typically executed will cause all joints to start moving in one direction then reverse, passing through the stiction region. This in turn excites controller dynamics and increases error at the end effector, resulting in a hole which can be off center, oblong, or have delamination at the edges. While research has been done to mitigate the effects that these errors have by increasing the stiffness of the robot, this thesis proposes to avoid the cause altogether. This is accomplished using the robot's redundant degree of freedom with the drilling tool, and by optimally selecting tool orientations throughout the drilling motion to accomplish the same task while minimizing individual joint reversals.

Section 1.1 introduces background knowledge related to industrial robots and machining. Section 1.2 discusses the specific problems faced by drilling robots. Section 1.3 outlines the solution to this problem and the contributions of this thesis to the field.

# 1.1 Background

Industrial robots are typified by six degree of freedom (DOF) manipulators comprising revolute or linear axes in a serial kinematic configuration, such as the one shown in Figure 1. These robotic arms mimic the functionality of the human arm, and similar terminology can be applied. The first two joints (starting from the base) constitute the "shoulder", while the third joint is associated with the "elbow". Together, these three joints control the overall position of the robot in its workspace. The last three joints form the "wrist" which controls fine positioning and orientation. Each of these joints is driven by a servo motor coupled to a reduction gearbox, which can turn some finite amount in either direction before hitting a mechanical joint limit. The end of the last link of the robot is called the flange, where different tools can be mounted depending on the desired operation. These tools are typically denoted as "end effectors." The point of interest on the tool for the

operation (e.g. the tip of a spindle, the nozzle of a glue gun) is called the Tool Centre Point (TCP).



Figure 1. Typical 6DOF industrial manipulator robot, the KUKA KR 6 R700-2, located at the University of Manitoba Intelligent Digital Manufacturing Lab (IDML).

Industrial manipulation robots were initially purposed for pick and place applications within the manufacturing industry. This required the robot to simply pick up parts from a known location and move them to another. However, thanks to their flexibility and cost effectiveness, they have been adapted to further roles including welding, gluing, inspection, and most significantly for this thesis, material removal [1].

Machining with robots, be it milling, drilling, or other material removal techniques, has advantages and disadvantages when compared to traditional CNC machining. For their

#### Chapter 1 | Introduction

cost, industrial robots can operate on a much larger workpiece since they are placed at some point and can reach around, rather than needing to fully enclose the workpiece like a CNC machine. The structure of a robot is also much more compact than other machines, allowing them to be placed in difficult locations or orientations, or bunched together in close proximity. The most significant disadvantage compared to CNC machines, however, is their reduced overall stiffness [2]. Since they have a serial kinematic configuration, the location of each joint is dependent on the angle of the previous joints. This means that even a small deviation at the base of the robot will propagate through the entire structure to result in a large error at the TCP. Reduced joint stiffness, especially pronounced when excited by large forces or controller dynamics, has limited the application of robots in certain industries; most notably aerospace and precision manufacturing. The tight tolerances mean that robots are not quite able to achieve the required results. However, with drilling upwards of a million holes in a commercial aircraft, the adoption of robots in these sectors could be significantly increased. Besides improving the stiffness of a robot, identifying and correcting the source of errors could lead to improved performance and more widespread adoption of robots.

# **1.2 Problem Statement**

As automation in the manufacturing sector increases, robots are being called upon to perform a wider variety of tasks beyond pick and place. Namely material removal tasks such as milling and drilling have become more popular in recent years. However, the stringent tolerance requirements of certain industries, especially aerospace, precludes robots from being more widely adopted in these applications. The most significant barriers are the low stiffness and propagation of errors along a robot's serial kinematic chain, so that even a relatively small deviation at the base joint of the robot results in unacceptably high error at the end effector. A major source of these joint errors is brought on by static friction and backlash during joint reversals. This is especially prevalent in drilling motions since the typically programmed instructions will linearly translate the end effector with no change in orientation of the end effector. These commands cause all joints to reverse once the target depth has been reached. Since the joints must break stiction and experience backlash each

#### Chapter 1 | Introduction

time they reverse, there is a sudden inconsistency in the force profile demanded of the motors at this point. The controller response in turn causes deviation at the end effector; exacerbated by the poor overall stiffness of industrial robots when compared to other machining platforms. To perform more precise drilling, the errors associated with these joint reversals must be mitigated or avoided.

# **1.3 Thesis Objectives and Contributions**

In this research, the proposed solution to the problem statement is to avoid joint reversals altogether during the drilling motion. A rotating spindle is used during drilling, creating a redundant DOF which allows for infinite possible tool orientations at any point in the path. Leveraging this kinematic redundancy allows for additional possible end effector rotation throughout the drilling motion while still drilling in the same location. Particle Swarm Optimization (PSO) is used to select the best orientations throughout the drilling motion with the global objective of minimizing joint reversals. A standard drilling motion would require all six joints to reverse as the drill bit is removed from the workpiece. However, strategically picking the orientation of the tool at the start, target, and end points of the drilling motion will allow the robot to accomplish a drilling task with fewer or even zero reversals. This does not inherently add stiffness to the robot or change the dynamic properties, but rather avoids the phenomena which lead to forces that excite dynamic errors in the robot. While stiffness (and many other metrics) could also be included as optimization goals, the objective of this thesis is to demonstrate that the principle of joint reversal avoidance is applicable to typical industrial manipulator robots. The proposed methodology allows a robot integrator to find, for arbitrary drilling motions, some rotation about the redundant degree of freedom which can be planned throughout the motion to reduce joint reversals. The implementation and implications of this strategy are also explored, showing a significant improvement in the precision of the drilling motion. The ability of the robot to achieve a zero-reversal drilling profile throughout the workspace is also explored.

# 2 Background and Literature Review

This chapter reviews the core afflictions causing robotic machining errors and the different approaches that researchers have proposed to address such errors. Section 2.1 reviews the detrimental effects of backlash and friction in industrial robots. Section 2.2 discusses the optimization of robot pose and workpiece placement for enhanced stiffness and accuracy. Section 2.3 reviews the methods researchers have used to leverage an additional DOF to further optimize for a range of criteria. Section 2.4 discusses optimization constraints and methods to highlight the most suitable approach for drilling.

# 2.1 Backlash and Stiction in Industrial Robots

Static friction, or stiction, represents the resistance that must be overcome to begin motion of a body at rest against a surface. Once moving, this resistance to motion will sharply decrease, then follow a linear change as a function of the relative velocity. In industrial robots, stiction is encountered when starting the movement of joints from rest, or when joints reverse and momentarily have an angular velocity of zero. In both cases, the rapidly changing resistance causes overshoot and undershoot of the demanded torque to execute the desired motion. This, along with backlash, causes eccentricity or glitches in the motion at locations corresponding to joint reversals, as documented by Oh [3]. Backlash was found to be a significant contribution to the overall error of a robot, along with overall joint compliance, according to Schneider et al. [4].

Friction is notoriously difficult to model [5]. If the friction profile throughout a motion can be predicted accurately, the controller may be able to compensate the resultant errors. This has been attempted in two ways. The first is to execute the desired motion and record the positional error, then offset each point in the motion path according to this error in the opposite direction [6]. While this can significantly improve the accuracy of a robot, it is a not practical approach as it requires extensive measurements and trials for any desired path. In addition, these measurements must be repeated often to account for variations due to

changes in temperature, lubrication, and wear conditions. The second method for mitigating friction errors is to use an analytical model to predict and compensate such errors. Friction identification algorithms are run on the robot to identify the friction behaviour in each joint, then when a new motion is executed, the controller will adjust the commanded torque to better overcome regions of expected high friction [7]. While the above-mentioned methods can reduce the errors to some extent, they fail to reliably compensate friction errors in general arbitrary motions. This is mainly due to the complexity, nonlinearity, and inconsistency of friction. Friction compensation methods are still highly dependent on the trajectory and specific task being performed [8]. There are a large number of factors that affect friction, such as lubrication composition, particulates in the lubricants, wear in mechanical connections, temperature of all parts in contact, and material properties. Even if perfectly predicted and compensated, friction is not the only cause of error during joint reversals, and backlash still causes significant problems.

Backlash refers to the space between internal meshing components which must be eliminated for mating to occur. In a case where a gear must switch direction, its teeth will not be perfectly meshed with the other gear at the moment of reversal and must traverse some short distance before mating begins again. This causes a jerk in the output motion which again leads to error at the robot end effector. Many robots are now being designed with harmonic gear drives, which due to their design can achieve zero or very low backlash. However, not all joints on all robots use this system, with larger robots and certain joint geometries favoring planetary gear systems or a belt drive. Moreover, additional joint and metrology components can suffer from wear and assembly errors which will introduce backlash even in an ideal gear drive. Again, techniques have been implemented to anticipate and compensate for these errors, but as of yet still cannot completely eliminate them. Because of this, researchers have tried to improve performance through other means, namely optimization of the robot pose and workpiece placement.

## 2.2 Optimization of Robot Pose and Workpiece Placement

Industrial robots are lauded for their adaptability and working range. A single robot can perform multiple operations within a work cell. This also means that when selecting the placement of a part to be operated on, integrators will typically have a wide array of options for the placement and orientation of the workpiece within the reach of the robot. Leveraging this, researchers have proposed techniques to strategically select the best placement of the workpiece relative to the robot such that performance is enhanced [9][10]. This can be, in the case of drilling operations, for the stiffest overall posture of the robot. The stiffness can be quantified for any pose similarly to the dexterity, then the part placed such that any holes which need to be drilled can be done so when the robot is in a relatively stiff posture [11] [12]. This reduces the error caused by machining forces, inertial forces, as well as the glitches introduced by stiction and backlash. The compliance condition number is commonly used for the inclusion of stiffness as a metric in optimization, however there are others which are preferred depending on the operation [13][14]. While this has the advantage of stiffnesing the entire system, thereby reducing the effects of many errors, it does not address the cause of these errors, only the symptoms.

Optimization of the workpiece placement has also been explored for operations such as milling [15]. The stiffness and natural frequencies of the robot change as a function of pose, but can be predicted [16]. Optimal workpiece placement can be selected to ensure the robot flexibilities do not contribute to chatter or other undesirable machining harmonics.

Common to both milling and drilling operations is a rotating tool which is mounted on a spindle at the end effector. Since the exact rotation angle of the tool does not affect the motion planning, general robotic machining operations require only five degrees of freedom. Performing machining operations with common six-axis industrial robots introduces a redundant degree of freedom which can be leveraged for further optimization of the process. While well documented in the literature, this is not something always explored in industry. When robots are utilized for machining applications, the initial programming is still carried out by a human operator. Typically, the robot will be jogged to some position, then a linear

translation forward then backwards will be commanded. Sometimes, more advanced software is used which is capable of integrating CAD data into robot work instructions and can allow a user to generate a toolpath based on the part's CAD geometry. However, both these methods suffer from the preconceptions and limitations in imagination of the programmer. For a simple drilling motion, the intuitive approach is to maintain the drill in the same orientation and with the robot at some upright posture. But thanks to the redundant degree of freedom, this is not necessary. By evaluating different tool orientations similarly to how posture stiffness optimization is done, performance differences in the drilling motion arise and the best can be selected.

# 2.3 Redundancy Resolution

When the number of degrees of freedom of a kinematic chain is greater than the degrees of freedom which need to be defined in the workspace, the kinematics are said to be redundant. The inverse kinematics for a typical six axis robot in three-dimensional space require all six degrees of freedom (X, Y, Z, A, B, C) to be defined. But under certain conditions, one or more of these degrees of freedom is undefined. This can be the case when an arm is designed with additional degrees of freedom intentionally to allow greater flexibility (such as the KUKA iiwa), or else for six-axis industrial robots, when the axisymmetric performance of a tool about its work axis creates the ambiguity. For example, a marker will draw the same line despite any rotation about its longitudinal axis. In the same way, since a drill bit will rotate during normal operation, there will be no kinematic impact on the operation being performed due to the tool's rotation about this axis (provided coordinate systems are properly calibrated). In both cases, some redundancy resolution technique is required to decide on the orientation to take, and fully define the inverse kinematics at a given position. Broadly, redundancy resolution is defining the null space of the robot such that a secondary task is executed while also executing the primary task (moving the end effector in some way). Since robot properties can vary dramatically with pose, it follows that resolving the redundancy can be done strategically to obtain desirable outcomes, much like the optimization of workpiece placement.

In early redundancy resolution literature, the focus was on utilizing the weighted pseudo-inverse Jacobian to select the joint velocities at any incremental time step which would have some beneficial effect on an optimization criterion while still fulfilling the desired translation of the end effector [17]. This method still sees use in applications where the redundancy is described as distributing individual joint motion in the null space [18], however requires a starting configuration and is not always the most suitable approach when the system is functionally, rather than intrinsically redundant. This technique has the system began from a defined position and tasked with moving a specified distance in some direction. Since the instructions do not fully define the final position (or if the robot was operating in a lower dimensional space than its own degrees of freedom), the robot could accomplish the task in an infinite number of ways. But by quantifying the overall performance of the robot with respect to the motion of each joint, the robot could decide on a method of attaining the new position by the motion of joints which provide the best increase in performance. The most common criterion for this performance was (and still is often used alongside other metrics) avoiding joint singularities, such as in [19]. Singularities are described in detail by Aboaf and Paul [20], but in general can be thought of as multiple joints (such as joints 4 and 6 in the wrist) aligning in such a way as to make a small end effector movement require excessive joint motion. A characteristic of singularities is impossibly high joint velocities (imagine flipping your wrist from a thumbs down to a thumbs up position instantaneously) and so configurations aimed to minimize the joint velocities. As secondary criteria, joint or acceleration limits could also be included to ensure the robot avoided moving any joint near to the angular limits imposed by hardware limitations or to reduce torque demands [21]. The main disadvantage with this technique, however, is that it is a local optimization method which only considers the benefits at one time step ahead of the current position. This makes it susceptible to being trapped at local minima. Always taking a step downhill does not necessarily help you descend a mountain if you are in the crater of a volcano. For the goal of minimizing reversals throughout the entire path, a global approach must be taken.

Techniques such as a Rapidly Expanding Random Tree (RRT) have been explored to efficiently search the closest local options to achieve global convergence [22]. However, in the case of robot manipulators, it often suffers from the computational requirements necessary for calculating the pseudo-inverse Jacobian which becomes increasingly complex as additional degrees of freedom are added to the robot. Furthermore, it is important to make the distinction between intrinsic and functional redundancy [23]. An intrinsically redundant manipulator has more degrees of freedom than could ever be defined by the space it is in – for example, a 4 joint planar arm in 2D space, or a 7DOF manipulator in 3D space. Functionally redundant manipulators are those which are "artificially" redundant, in that they have a number of degrees of freedom which could be fully defined by their work dimension but the operation to be executed does not define one or more of them. This is the case when drilling in the physical world with a 6DOF manipulator since the drill bit is already rotating. When functional, rather than intrinsic, redundancy must be resolved, the Jacobian becomes nonsingular square which is unsolvable without the use of artificial additions to resolve the discrepancies [24].

Even assuming that the weighted pseudo-inverse Jacobian could be efficiently calculated for the functionally redundant case at hand and characterized in a meaningful way throughout the entire path simultaneously, the optimization criteria becomes another barrier. The optimization criteria, as has been discussed, is to avoid joint reversals. Mathematically, the metric to avoid joint reversals is to apply the limitation that no joint velocities can equal zero, since slowing to a halt then accelerating from that point would physically represent a joint falling into the static friction region, with or without a reversal. Besides the difficulties in setting an inequality constraint through the entire set of differential equations according to traditional intrinsic redundancy resolution, there are further mathematical issues. If the constraint is set to maximize the absolute joint velocity (thereby avoiding joint velocities of zero) this comes in direct conflict with the singularity avoidance criterion of minimizing absolute joint velocity. Optimizing in this way would push the robot into a singular position, and while the kinematic path may be valid, it would fail upon implementation onto a physical robot. Objective functions exist which can moderate

joint speed between two extremes. However, the upper limits are variable depending on the robot, geometry, and specific joint, increasing the complexity of the problem.

In light of all these difficulties, it becomes more straightforward and efficient to simply apply a metaheuristic optimization technique to a large number of fully resolved potential configurations for a given path, then select the best of all the options. This is fundamentally different and should not be confused with the approach detailed by Rokbani [25], wherein the inverse kinematics themselves are unknown, and solved using a PSO algorithm. The research presented in this thesis has instead deterministically calculated the inverse kinematics at any point, but uses metaheuristics to select a globally optimal solution of inverse kinematics which achieve the path motion.

Studies have already applied metaheuristic optimization techniques to evaluate and select the best possible tool orientation for a given task, according to some metrics [26]. Now, beyond just optimizing the positioning of the workpiece within the workspace, the rotation of the tool about its redundant axis can be included as an additional optimization parameter. Or, for scenarios where the workpiece must be fixed in a certain position relative to the robot, there still exists a parameter which can be optimized.

However, the studies which take advantage of this additional degree of freedom typically do so only at one position relative to the workpiece. Some rotation is applied, then the motion is executed as normal. Most (but not all) of these studies focus on optimizing some aspect of the dynamic properties of a robot arm at a single point, usually the target. This is entirely valid, since these properties will change throughout the robot workspace, but there is usually not a significant change throughout the relatively short drilling motion. For the case of drilling, this means that the tool's rotation about the tool's work axis is constant throughout the motion. But these are local optimization methods, the ideal case for a single point, and there are relatively few cases of this optimization being applied globally to a whole path, and none to the author's knowledge of any which incorporate joint reversals as a criterion. In this thesis, it is shown that by taking a global optimization approach which considers the rotation about the redundant degree of freedom at several points through a

motion, the additional optimization criteria of reversal minimization can be applied to further improve the performance. The optimization methods have been well described and commonly used for similar tasks in literature. However, the author acknowledges that the fields of path planning, optimization, and redundancy resolution are extensive, and there may exist different methods for optimizing the task presented.

# 2.4 Local and Global Optimization

Optimization is a broad topic, and the ideal method to optimize some tasks is highly dependent on the task itself. The task presented here requires that the entire motion be optimized. This implies that the search method used will be global, rather than local. In local searches, only the next step is considered, and there is a movement in the direction which, at the current point, leads to the most desirable score. However, this method is susceptible to being trapped at local minima. Additionally, these search methods require a continuous function to evaluate as the objective. This poses a problem in cases when the function should switch cases or a discontinuity is imposed (such as with joint limits). Global search methods, on the other hand, are much more applicable to tasks which require the evaluation of some combination of input parameters to find the best candidate. In this case, since the performance of the entire path is evaluated simultaneously, some overall score is assigned to the set of parameters which yield that specific path. By looking at the entire path holistically, further refinement can be imposed than would be possible for an algorithm which only evaluates some set of joint positions or velocities.

For example, assume that some combination of joint velocities is perfectly valid during the plunging motion so long as they continue to all move in the same direction during the retract motion. However, that same combination of joint velocities may no longer be desirable (and therefore should get a worse score) if they require that one or more of the joints switch direction once the target is reached. A local method cannot look ahead to see the future implications of a current decision, and so only a static score for that combination of joint velocities can be defined. Some method is required to refine the score beyond simply a function of instantaneous joint angles or velocities. So, the method selected must evaluate

the full path, and therefore must have some method to generate a fully defined path from a combination of input parameters. With this knowledge, it can be seen that this is not a redundancy resolution technique which aims to specify the inverse kinematics at each point. Rather, it is a method for selecting the best way to fully define a functionally redundant path to achieve some additional goal besides the desired tool path.

These requirements fit the description of a metaheuristic optimization technique. That is, one which is combinatorial, checking a set of input parameters then refining the subsequent selection of parameters to achieve better results. After enough iterations and with some robust strategy for finding better input parameters, the technique can be reasonably confident in achieving the goal. Many of these metaheuristic techniques and variations upon them exist, but PSO was selected for its robustness, efficiency, and ease of implementation [27][28]. Further details on PSO and comparisons to other metaheuristics are provided in Section 4.1.

# 2.5 Summary

Robotic machining has been plagued by the errors in industrial robots, in part excited by stiction and backlash during joint reversals. With the additional DOF afforded during milling and drilling, researchers have optimized the positioning of the workpiece as well as the posture of the robot to stiffen the system as much as possible. This optimization can be done in a number of ways, but especially well-suited for the defined problem is a metaheuristic optimization technique. This thesis will demonstrate the application of PSO with the goal of minimizing joint reversals during a drilling motion to avoid their negative effects altogether and as a result improve performance.

### **3** Kinematic Modeling and Motion Planning

This chapter derives the mathematical model of robotic drilling motions as a function of robot kinematics and path parameters. Sections 3.1 and 3.2 define necessary terminology and assumptions about the robot and workspace. Most industrial robots are comprised of six revolute joints, and so the rotation of each joint propagates throughout the robot until the final position of the end effector. Kinematically modeling this propagation is done according to the Denavit-Hartenburg (DH) convention in Section 3.3. Likewise, to achieve some point in space, the kinematic chain which positions the end effector at the desired position and orientation must be solved. These are the forward (Section 3.4) and inverse (Section 3.5) kinematics of the robot, in this case a KUKA KR 6 R700-2. However, the drilling motion having a redundant degree of freedom renders it underdefined. Section 3.6 demonstrates how to resolve a single solution from the infinite options available.

The kinematic methodology described in Section 3.3 – 3.5 is well understood and not a novel contribution. However, no sufficiently similar model which could be adapted to the robot in question was found, and so a new model was developed. This is an application of textbook methods to a new platform, and shown here in the hopes that it can be useful to other researchers using a similar robot.

# 3.1 Workspace Definitions

Figure 2 shows a schematic of the six-axis KUKA KR6 R700-2 robot used in this study. The origin of the global coordinate system (GCS) is positioned coincident with the origin of the robot such that  $X_{GCS} = Y_{GCS} = Z_{GCS} = 0$  with the front of the robot oriented along the positive  $X_{GCS}$  axis,  $Z_{GCS}$  aligned vertically and towards the top of the robot, and  $Y_{GCS}$  positioned according to the right-hand convention as shown in Figure 2. Point coordinates are given in the GCS as  $[X, Y, Z]_{GCS}$ , with units of millimeters (mm) throughout this thesis.



Figure 2. Coordinate systems within the robot workspace; start, target, and end points of a sample drilling motion, and the  $\psi$  angle used to describe rotation about the drilling spindle's redundant degree of freedom.

For any drilling motion, 3 points must be defined: the start of the motion, the target and the end of the motion. These are labeled  $P_s$ ,  $P_t$ , and  $P_e$  respectively and are composed of the X, Y, and Z coordinates at each point. The drilling motion consists of two distinct motions called the plunge and retract. The plunge denotes movement from  $P_s$  to  $P_t$  and the retract denotes movement from  $P_t$  to  $P_e$ . These motions are defined by plunge and retract vectors,  $V_P$  and  $V_r$  respectively such that

#### Chapter 3 | Kinematic Modeling and Motion Planning

$$\boldsymbol{V}_{\boldsymbol{p}} = \boldsymbol{P}_{\boldsymbol{t}} - \boldsymbol{P}_{\boldsymbol{s}} = \begin{bmatrix} \boldsymbol{x}_{t} - \boldsymbol{x}_{s} \\ \boldsymbol{y}_{t} - \boldsymbol{y}_{s} \\ \boldsymbol{z}_{t} - \boldsymbol{z}_{s} \end{bmatrix} = -\boldsymbol{V}_{\boldsymbol{r}}, \tag{3.1}$$

and a complete drilling motion consisting of a plunge followed by a retract is denoted  $V_{p+r}$ .

The orientation of the drill is partially defined by  $V_p$  which points from  $P_s$  to  $P_t$ . In this case, the tool's work axis is  $Z_{TCS}$  and is aligned such that negative motion along  $Z_{TCS}$ corresponds to the drill plunging into the material. The ambiguous orientation about  $Z_{TCS}$  is the redundant degree of freedom and is labelled as  $\psi$  with an arbitrary definition of  $\psi = 0^\circ$ when  $Y_{TCS}$  and  $Y_{GCS}$  are parallel and positive in the same direction as shown in Figure 2. This aligns the GCS with the TCS nicely for this case of drilling direction. However, for drilling in any general direction,  $\psi = 0^\circ$  may be designated as the tool orientation which brings the robot flange closest to the robot base. If a workpiece is being drilled, it may be more helpful to align  $\psi = 0^\circ$  with some axis of the workpiece coordinate system. In any case, positive rotation about  $Z_{TCS}$  according to the right-hand rule corresponds to an increase in  $\psi$ . A set of angles which consists of some  $\psi$  at each of the start, target, and end points is denoted as

$$\boldsymbol{\psi} = [\psi_s \quad \psi_t \quad \psi_e]. \tag{3.2}$$

This  $\boldsymbol{\psi}$  fully defines the orientation at each of the three points, and by linear interpolation the orientation at all intermediate points.

#### 3.2 Assumptions and Conventions

The methodology laid out in this thesis is trajectory agnostic; that is, the same results can be applied regardless of the velocity profile used in the movement of the robot. Since the velocity profile of a robot is not always known, or there is some delay between real-time sensing and compensation, it is not always practical to create methods which rely on precise timing of a path. The robot controller is responsible for planning the trajectory and following the ideal input from the programmed instructions. As such, there will be some discrepancy

#### Chapter 3 | Kinematic Modeling and Motion Planning

between the joint paths caused by the controller dynamic response, but these are assumed not to induce any additional reversals (Section 5.3 verifies this assumption).

The plunge direction for all simulations and experiments was chosen as  $V_p = [1, 0, 0]_{GCS}$  for visualization purposes. However, the proposed methodology is general and can be applied to more complex plunge motions which include translation in all the GCS axes.

The performance of the proposed method and ability to achieve zero reversals was assumed to be symmetrical about the *XZ* plane. This was verified with simulation in Section 4.6, however only to an extent due to the nature of numerical optimization reaching a minimum. For subsequent mapping of workspace performance, the results were obtained on one side of  $Y_{GCS}$  only.

Unless otherwise stated, all units of distance are in the industry standard of millimeters. Units of angular displacement are in degrees. In all cases, the right-hand rule applies as the convention when determining frame placements, which is especially relevant when identifying the locations and orientations of frames for finding DH parameters.

# 3.3 Denavit-Hartenburg Parameter Identification

The KR 6 R700-2 was a new addition to the IDML at the time of the research being carried out for this thesis. The kinematics for this robot were not publicly available but were necessary for the optimization model to examine valid robot paths offline. Therefore, the forward and inverse kinematic models were fully derived, troubleshot, and tested, and will serve as a resource for future students to carry out kinematic examination of the robot. Schematics and 3D models of the KR 6 R700-2 are publicly available from KUKA, and were used to assign frames to each axis of the robot according to the modified DH convention [29]. The neutral position assumed the robot fully extended horizontally, as shown in Figure 3. However, the default controller home position of the KR 6 R700-2 is with shoulder upright and the elbow bent at 90°, forming a  $\exists$  shaped robot posture. Table 1 shows the four DH parameters  $\alpha$ , a, d, and  $\theta$  which were found for each of the six joints.



Figure 3. Frame assignments to a KUKA KR 6 R700-2 according to the modified DH convention, as well as the TCP offset used for the sample drilling tool.

Joint	α <sub>i-1</sub> (deg)	<b>a</b> i-1 <b>(mm)</b>	<b>d</b> i (mm)	θi (deg)
1	180	0	-400	0
2	90	25	0	0
3	0	335	0	-90
4	90	25	-365	0
5	-90	0	0	0
6	90	0	-90	0

 Table 1. Modified DH convention parameters for a KR 6 R700-2.

Additionally, for visualization of the robot using Peter Corke's Robotics Toolbox [30], parameters were found according to the standard DH convention, and are given in Table 2.

Table 2. Standard DH convention parameters for a KR 6 R700-2.

Link	θi <b>(deg)</b>	di (mm)	a <sub>i</sub> (mm)	αi (deg)
1	0	0	25	90
2	0	0	335	180
3	-90	0	-25	90
4	0	-365	0	90
5	0	0	0	-90
6	180	-90	0	0

### 3.4 Forward Kinematics of the KUKA KR 6 R700-2

Forward kinematics (FK) translate known joint angles to an end effector position and orientation. Following the DH convention, the FKs can be found by successively multiplying transform matrices from the base of the robot to the end effector. The resultant matrix has as unknowns  $\theta_1, \theta_2, ..., \theta_6$ . The transform matrix associated with the mDH, representing the rotation and translation

$${}_{i-1}{}^{i}T = \begin{bmatrix} \cos(\theta_{i}) & -\sin(\theta_{i}) & 0 & a_{i-1} \\ \sin(\theta_{i})\cos(\alpha_{i-1}) & \cos(\theta_{i})\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -d_{i}\sin(\alpha_{i-1}) \\ \sin(\theta_{i})\sin(\alpha_{i-1}) & \cos(\theta_{i})\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & -d_{i}\cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
(3.3)

which, when applied for each set of DH parameters yields the transformation matrices  ${}_{0}^{1}T$ ,  ${}_{1}^{2}T$ ,  ${}_{2}^{3}T$ ,  ${}_{3}^{4}T$ , and  ${}_{5}^{6}T$ . The product of all these matrices creates the matrix

$${}_{0}^{6}T = {}_{0}^{1}T \times {}_{1}^{2}T \times ... \times {}_{5}^{6}T, \qquad (3.4)$$

which describes the transformation from the base to the flange as a function of the DH parameters. One additional transformation is usually necessary for practical applications,  ${}_{6}^{E}T$  relating the geometry of the end effector. The matrix  ${}_{0}^{E}T = {}_{0}^{6}T \times {}_{6}^{E}T$  then gives the position and orientation of the TCP in the global coordinate frame.

To verify this against the physical robot, the robot was jogged to some non-trivial pose and the joint angles along with TCP position recorded. When the associated joint angles were input into  ${}_{6}^{E}T$ , the TCP coordinates agreed.

# 3.5 Inverse Kinematics of the KUKA KR 6 R700-2

The inverse kinematics are analytical solutions to the problem which requires the robot end effector to attain some desired position and orientation. The processes followed in this section are common for a 6DOF spherical wrist robot, however the equations presented are simplified, and not generalized for any robot. The equations are derived by

#### Chapter 3 | Kinematic Modeling and Motion Planning

following the process of successive premultiplication but are unique to the specific arrangement of axis frames assigned according to the DH convention.

Typically, the inverse kinematics for a spherical wrist robot, such as the one analyzed, will result in 8 sets of discrete joint angle solutions arising from the use of dual solution identities (A.1) to (A.7) shown in Appendix A. These can generally be categorized as elbow up or down, elbow left or right, and wrist normal or flipped. Not all solutions are necessarily feasible in the physical robot due to joint limits, however at this step no solutions are immediately discarded.

The functions *cos* and *sin* will be abbreviated as *c* and *s* respectively. The subscript *i* represents the associated  $\theta_i$ . For example,  $s_{12}$  denotes  $\sin(\theta_1 + \theta_2)$ . To align the frame of joint 3 with the zero-position marked on the physical robot, an offset of 90° was added such that  $\phi_3 = \theta_3 - 90^\circ$ . Throughout this section  $\phi_3$  is used for brevity.

An expression relating the end effector position and orientation to each joint angle can be found by simply multiplying the forward kinematic transformation matrices  ${}_{0}^{1}T \times {}_{1}^{2}T \times ... \times {}_{5}^{6}T$  and taking the inverse. However, this becomes extremely complicated to solve analytically for a general case. Instead, by taking advantage of the spherical wrist nature of the robot, the problem can be broken into two parts: position and orientation. The position component is solved with  $\theta_{1}$ ,  $\theta_{2}$ , and  $\phi_{3}$  while the orientation component is solved by  $\theta_{4}$ ,  $\theta_{5}$ , and  $\theta_{6}$ .

First, from a known target position and orientation in the workspace  $P = (X_{GCS}, Y_{GCS}, Z_{GCS})$ , the location of the wrist can be obtained as a function of the geometry of the wrist and the end effector. The relative displacement from the TCP to the spherical wrist intersection is constant and known based on the geometry of the end effector.

The wrist position, or  $D_w$ , are the cartesian coordinates of the intersection of the axes of joints 4, 5, and 6 in the global coordinate system. This is denoted as

$$D_{w} = \begin{bmatrix} p_{xw} \\ p_{yw} \\ p_{zw} \\ 1 \end{bmatrix}.$$
(3.5)

From the forward kinematics, this point can be represented as the translational component (last, fourth column) of  ${}_{0}^{4}T$ ,  ${}_{0}^{4}T_{4}$ 

$$= \begin{bmatrix} \cos(\theta_1)(a_2 + a_4\cos(\phi_3 + \theta_2) + d_4\sin(\phi_3 + \theta_2) + a_3\cos(\theta_2) \\ -\sin(\theta_1)(a_2 + a_4\cos(\phi_3 + \theta_2) + d_4\sin(\phi_3 + \theta_2) + a_3\cos(\theta_2) \\ d_1 + d_4\cos(\phi_3 + \theta_2) - a_4\sin(\phi_3 + \theta_2) - a_3\sin(\theta_2) \\ 1 \end{bmatrix} = D_w.$$

$$(3.6)$$

Then, the technique of successive pre-multiplication can be used to increase the number of available equations. Both  ${}_{0}^{4}T_{4}$  and  $D_{w}$  are multiplied by  $[{}_{0}^{1}T]^{-1}$ , or  ${}_{1}^{0}T$ . The first step yields

$${}^{0}_{1}T^{4}_{0}T_{4} = {}^{4}_{1}T_{4} = \begin{bmatrix} a_{2} + a_{4}c_{32} + d_{4}s_{32} + a_{3}c_{2} \\ 0 \\ a_{4}s_{32} - d_{4}c_{32} + a_{3}s_{2} \\ 1 \end{bmatrix} = \begin{bmatrix} p_{xw}c_{1} - p_{yw}s_{1} \\ -p_{yw}c_{1} - p_{xw}s_{1} \\ d_{1} - p_{zw} \\ 1 \end{bmatrix} = {}^{0}_{1}TD_{w}.$$
(3.7)

The following pre-multiplication is done with  $\begin{bmatrix} 2\\ 1 \end{bmatrix}^{-1}$ , or  $\frac{1}{2}T$  to obtain

$${}^{4}_{2}T_{4} = \begin{bmatrix} a_{3} + a_{4}c_{3} + d_{4}s_{3} \\ a_{4}s_{3} - d_{4}c_{3} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} d_{1}s_{2} - a_{2}c_{2} - p_{zw}s_{2} + p_{xw}c_{1}c_{2} - p_{yw}c_{2}s_{1} \\ d_{1}c_{2} - p_{zw}c_{2} + a_{2}s_{2} - p_{xw}c_{1}s_{2} + p_{yw}s_{1}s_{2} \\ p_{yw}c_{1} + p_{xw}s_{1} \\ 1 \end{bmatrix} = {}^{0}_{1}T_{2}^{1}TD_{w}.$$
(3.8)

And finally, pre-multiplying both sides by  $[\frac{3}{2}T]^{-1}$  gives

$${}^{4}_{3}T_{4} = \begin{bmatrix} a_{4} \\ -d_{4} \\ 0 \\ 1 \end{bmatrix} = {}^{0}_{1}T_{2}^{1}T_{3}^{2}TD_{w}$$

$$= \begin{bmatrix} \frac{p_{xw}}{2} (c_{(3-12)} - c_{312}) + \frac{p_{yw}}{2} (s_{(3-12)} - s_{312}) - a_{2}c_{32} + d_{1}s_{32} - p_{zw}s_{32} - a_{3}c_{3} \\ \frac{p_{yw}}{2} (c_{(3-12)} - c_{312}) - \frac{p_{xw}}{2} (s_{(3-12)} + s_{312}) + a_{2}s_{32} + d_{1}c_{32} - p_{zw}c_{32} + a_{3}s_{3} \\ p_{yw}c_{1} + p_{xw}s_{1} \\ 1 \end{bmatrix}.$$

$$(3.9)$$

These steps provide sufficient equations to solve for  $\theta_1$ ,  $\theta_2$ , and  $\phi_3$ . To solve  $\theta_4$ ,  $\theta_5$ , and  $\theta_6$  requires a set of equations relating the wrist position to the target. For this, the geometry of the end effector will have significant impact. The transformation from joint 6 to the TCP of the end effector is denoted  $\frac{E}{6}T$ . For the end effector used in this thesis

$${}^{E}_{6}T = \begin{bmatrix} 0 & 0 & 1 & -EEa \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -EEb \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
(3.10)

where *EEa* and *EEb* are 65mm and 100mm respectively, as shown in Figure 4.



Figure 4. Sample drilling end effector with associated coordinate frame and dimensions.

Now given this transformation, the relationship between the wrist and the target can be described either as  ${}_{3}^{E}T$ , or else  $\left({}_{0}^{3}T\right)^{-1}P$  where

$${}_{3}^{E}T = {}_{3}^{4}T_{5}^{5}T_{5}^{6}T_{6}^{E}T = \left({}_{0}^{1}T_{1}^{2}T_{2}^{3}T\right)^{-1}P, \qquad (3.11)$$

arising from the knowledge that the target point **P** must equal to the product of all frame transformations  ${}_{0}^{1}T_{1}^{2}T \dots {}_{6}^{E}T$ . A number of trigonometric identities and simplifications will be used, they are listed in Appendix A.

Given a target matrix  $P_{4\times4}$  the wrist position  $D_w$  of the robot is determined by the specific geometry of the end effector. If there is no end effector, it is given by

$$p_{xw} = p_x - d_6 \times \boldsymbol{P}_{1,3},$$

$$p_{yw} = p_y - d_6 \times \boldsymbol{P}_{2,3},$$

$$p_{zw} = p_z - d_6 \times \boldsymbol{P}_{3,3}.$$
(3.12)

With an end effector, the wrist position is determined by the specific geometry in the relative sliding, approach, and normal directions along with the associated rotations.

Then, from row 2 of (3.7) an expression for  $\theta_1$  in the form of (A.4) is found, where

$$a = p_{yw}, \tag{3.13}$$
$$b = p_{xw}.$$

Leading to the two known possibilities of  $\theta_1$ . With this variable solved,  $\theta_3$  is isolated in rows 1 and 2 of (3.8). Where, after manipulation of the equations through the use of (A.8) and (A.9),  $\phi_3$  takes the form of  $\theta$  in (A.7), while  $a = a_4$ ,  $b = d_4$ , and

$$c = \frac{(-d_1 - p_{zw})^2 \mp 2a_2\sqrt{p_{xw}^2 + p_{yw}^2} + p_{xw}^2 + p_{yw}^2 + a_2^2 - a_3^2 - a_4^2 - d_4^2}{2a_3}.$$
 (3.14)

 $\theta_2$  is then solved by substituting the corresponding  $\theta_1$  and  $\phi_3$  into rows 1 and 2 of (3.8), and applying (A.8) and (A.9) where

$$a = -d_{1} - p_{zw},$$

$$b = p_{xw}c_{1} - p_{yw}s_{1} - a_{2},$$

$$c = a_{4}s_{3} - d_{4}c_{3},$$

$$d = a_{4}c_{3} + d_{4}s_{3} + a_{3},$$
(3.15)

necessitating the condition

$$c^2 > a^2 + b^2. (3.16)$$

With this, the wrist position is given for any target as a function of  $\theta_1$ ,  $\theta_2$ , and  $\phi_3$ . The wrist kinematics are specific to the end effector geometry, so care should be taken if adapting these results to other applications.

Column 1 of the LHS of (3.11) takes the form of (A.5), solving for  $\theta_4$  with

$$a = -n_y c_1 - n_x s_1,$$

$$b = -n_x c_{23} c_1 + n_z s_{23} + n_y c_{23} s_1.$$
(3.17)

And the supplementary condition in (A.6) where

$$c = -n_x s_{23} c_1 - n_z c_{23} + n_y s_{23} s_1, ag{3.18}$$

giving  $\theta_5$ . Care must be taken to assign the correct configurations with the associated solutions. The flipped wrist solutions can alternatively be found with  $\theta_4 + \pi$  and  $-\theta_5$ .

Finally, from columns 2 & 3 in row 2 of the LHS of (3.11),  $\theta_6$  is found with (A.5) where

$$a = -S_z c_{23} - S_x s_{23} c_1 + S_y s_{23} s_1,$$
  

$$b = -a_z c_{23} - a_x s_{23} c_1 + a_y s_{23} s_1.$$
(3.19)

Note that *S* represents the sliding X, X, or Z vector associated with the end effector.

With all joint angles determined, the results can be used as joint commands for the robot to validate the inverse kinematics. This process may not be what is used by the controller, as that process is proprietary KUKA information, but the method shown was successful in predicted the required joint angles which could put a robot at a specific location and orientation in the workspace.

# 3.6 Redundancy Resolution in the Drilling Motion

The inverse kinematics give the capability of solving the set of joint angles for any point within the robot's workspace, but for the drilling motion in question the orientation is still undefined. The tool will perform equally well rotated at any angle about the spindle, and so some strategy is required for deciding the orientation of the tool throughout the motion. In this case, there are no additional factors to be concerned with (such as keeping a gravityfed reservoir relatively upright) and so any orientation about the tool Z axis is considered viable. Orientations are assigned at three points in the drilling operation: at the start of the motion, the target position, and at the end of the motion. These orientations will be optimized. As a matter of simplicity, the bounds of each of these orientation angles is set at  $\pm 180^{\circ}$ . As discussed in Section 2.3, there are many ways of resolving this redundancy, and more elegant methods which make use of the null space and weighted-pseudo inverse Jacobian may be formulated in the future. Additionally, if extended to include more degrees of freedom or additional optimization criteria, more input parameters may be necessary to reduce the dimension of the null space enough for an optimal solution to be found. The method used here involves using a few input parameters to fully define the path, then evaluating that path and comparing it to others according to some optimization criteria. In this sense, the problem of resolving redundant inverse kinematics is avoided because any point in the parameter space evaluated is fully defined, and it is only the motion itself which is initially undefined.

The chosen method to fully define all points along the path is as follows: First, identify  $\boldsymbol{\psi}$  for some  $\boldsymbol{P}_s, \boldsymbol{P}_t$ , and  $\boldsymbol{P}_e$  which define the target and plunge depth  $D_p = ||\boldsymbol{V}_p||$ . Then, for any distance  $\delta$  along the path, the associated  $\psi_{\delta}$  is linearly interpolated between  $\psi_s$  and  $\psi_t$  (if during the plunge motion) or  $\psi_t$  and  $\psi_e$  (if during the retract)

$$\psi_{\delta} = \begin{cases} \psi_{s} + (\psi_{t} - \psi_{s}) \left(\frac{\delta}{D_{p}}\right), & \delta < D_{p} \\ \psi_{t} + (\psi_{e} - \psi_{t}) \left(\frac{\delta - D_{p}}{D_{p}}\right), & \delta \ge D_{p}, \end{cases}$$
(3.20)

which fully defines the target position and orientation at any point along the drilling path.

# 3.7 Summary

A model is required before attempting optimization, which has been done in this chapter by identifying the DH parameters of the experimental platform to be used, deriving the forward and inverse kinematics associated with it, and finding methods to define the orientation from an underdefined target at any point within the workspace. However, these steps do not optimally resolve the redundancy, and so requires the use of combinatorial optimization techniques to evaluate and select the best parameters.
# **4** Optimization of Drilling Motion

This chapter presents the methodology for the optimization of drilling motions by leveraging the kinematic redundancies of robots. In this work, Particle Swarm Optimization (PSO) has been employed to determine the optimal robot pose and motion strategy during a drilling operation to avoid joint reversals. Section 4.1, describes the behavior and selection of the Particle Swarm Optimization algorithm. Section 4.2 outlines the application of this optimization to the formulated drilling case. Further detail on the objective function and optimization parameters is given in Section 4.3. The simulated result of the optimization process to the formulated problem with the robot model is shown in Section 4.4. Then, in Section 4.5, the weights and settings used in the optimization model are tweaked to ensure good performance as the process is repeated at multiple locations throughout the robot workspace in Section 4.6. Finally, Section 4.7 focuses on a single point in the robot workspace and seeks insight into the nature of the objective function by using brute force to map the output from every set of input parameters.

# 4.1 Particle Swarm Optimization Algorithm

First and foremost, Particle Swarm Optimization is not being presented here as the only or best method for solving the problem at hand. It is an efficient, simple, and robust algorithm which returned good results in this application. However, for more complex implementations with additional DOF or parameters creating high dimension optimization, PSO may not be the best option.

PSO takes inspiration from nature and the behavior of organisms (particles) in a group (swarm). Groups of animals can solve complex behavioral tasks very quickly, such as schools of fish which seem to move as one cohesive unit, even when unexpected events such as a predator attack occur. In reality, each individual is following some simple set of rules in response to whatever stimulus it observes or receives. The behavioral task set in PSO is the optimization problem (getting to the unknown minimum value of some objective function)

with a specific set of input parameters representing the "position" of an individual particle in the parameter space. How much, and in which direction, each particle moves from one iteration to the next is called the "velocity." This velocity is calculated individually for each particle at each iteration according to a "velocity expression." The motion of the particles is governed by the simple rules (the velocity expression) each particle follows. While these rules can vary between species and situation in nature, the set of rules which are given to particles in PSO is to favor positions closer to their personal best as well as the swarm's global best. After multiple iterations, this results in overall swarm movement towards the global minimum. The general steps and motion of particles in PSO is visualized in Figure 5.



Figure 5. Visualization of PSO behavior from initial chaotic motion, to searching the workspace, to finally converging near the global minimum.

The basic structure of PSO is as follows: generate a population of particles each with some set of optimization parameters. Then, evaluate their performance at their current position using the objective function. Apply the velocity expression to determine the subsequent motion of particles, then move particles to their new positions. Repeat until some stopping criterion is reached. A detailed description of the application of PSO is given in Section 4.2.

PSO and metaheuristic searching algorithms in general are especially well suited for the task at hand, i.e. robotic drilling, due to a number of factors. First, they do not require a

differentiable objective function. Analytical optimization techniques which guarantee convergence at some minimum typically require the derivative or gradient to be defined for every point. However, due to the selected objective function (see Section 4.3), there may be sudden discontinuities and sharp curves (as shown in Figure 6), rather than a smooth and well-behaved function. Metaheuristic searching algorithms are a suitable approach for dealing with discontinuous objective functions such as the one defined in this work.



*Figure 6. Sample visualization of local minimum, global minimum, and poor behavior near region of interest.* 

Second, a population-based optimization model can cover a large region of the parameter space in initial iterations compared to algorithms which only search using a single point. Since it is of interest here to better understand the objective function, being able to map a larger area quickly (and remembering points of interest while doing so) is more desirable than necessarily zeroing in on the single best point as quickly as possible. Again, since the objective function behavior is unknown and possibly poorly conditioned, a large population can more easily avoid being trapped in local minima while searching the parameter space.

Additionally, while there are a number of these metaheuristic population methods, the selection of PSO was primarily driven by the difficulty in conceptualizing the objective function, and the relatively small set of input parameters. A highly random initial search strategy was desired to broadly identify regions of potential minima. Once certain candidate

zones are identified, there is no guarantee that they will create a slope towards the minima, and there may be regions around them which are discontinuous. Importantly, these discontinuous regions may be very close to a minimum, and so the region should not be wholly disregarded if a single invalid point is found. For this reason, there is not necessarily an advantage to breeding nearby fit particles, as in Genetic Algorithm (GA). Besides, the number of input parameters is so small that introducing a single mutation in a GA entity causes an extremely significant relative deviation from the unmutated version and including particle and swarm bests gives the PSO a memory component which GA lacks [31].

Finally, the population should eventually converge and perform many pseudorandom searches in the best identified region to try and achieve zero reversals. So, a method which can adapt throughout the search or naturally exhibit this convergence behavior quickly is desirable. However, it could still be beneficial to converge at a number of local minima, especially when trying to analyze the behavior of a new objective function. At the outset, it is unknown if there is only a very narrow set of parameters which achieves zero reversals, or if there are many. PSO achieves this with the help of adjustable weights which can affect a particle's inertia (momentum) and tendency to favor personal best over the global best (clustering at local minima). This behavior is illustrated in Figure 7.



Figure 7. Visualization of particle movement with different weights. A high cognitive factor  $(W_{PB})$  will keep a particle near its personal best, while a high social factor  $(W_{GB})$  will convince a particle to move to the swarm's best.

PSO fits all the aforementioned criteria and is relatively efficient computation-wise compared to other metaheuristic algorithms [32]. PSO has been used in a number of studies of posture optimization with good results [33] and proved effective in initial trials of the process outlined here so was selected to continue. Countless variations of the PSO algorithm have been proposed [34], most of which claim improvement in some way or another under certain conditions or with certain objective functions. While a basic implementation of PSO proved effective for this thesis, further refining the process and exploring other models for this objective is an encouraging domain of future study.

# 4.2 Application of PSO for Minimizing Joint Reversals

Like all metaheuristic optimization algorithms, PSO requires some set of input parameters to iterate as well as an objective function to evaluate performance. For this case, there are three optimization parameters:  $\psi_s$ ,  $\psi_t$ , and  $\psi_e$ , representing the twist about the work axis at  $P_s$ ,  $P_t$ , and  $P_e$  respectively (see Figure 8).



Figure 8. Visualization of psi angle with the sample drilling tool.

The objective function essentially counts the number of reversals during a given drilling motion, with the option to include other parameters such as joint travel or stiffness. Since there are three parameters, it is possible to visualize the movement of the swarm in 3D space. With more than three parameters visualization becomes difficult as not all input parameters can be mapped to an axis in space. The coordinates or "position" (*X*) of a particle at iteration *i* is given by a set of  $\psi$  parameters

$$X_i = \boldsymbol{\psi} = [\psi_s, \psi_t, \psi_e].$$

The movement of a particle in the parameter space is its "velocity" and described as the change in position; the difference between coordinates from one iteration to the next

$$V_{i-1} = X_i - X_{i-1}.$$
(4.2)

Random initial velocities are generated and assigned to each particle. A starting guess may be used, however since it is still difficult to predict where the zero-reversal case will be achieved this is not normally accurate. But including a starting guess and limiting the velocity may prove to be effective at reducing the convergence time of the algorithm in future, when the location of a minimum for any given position can be reasonably estimated. A flowchart of the PSO algorithm and implementation in this context is shown in Figure 9.

(4.1)

(1 2)



Figure 9. Flow chart of PSO algorithm implemented within the context of minimizing joint reversals for a drilling motion.

Before the first iteration, the PSO algorithm first defines values to use throughout, such as term weights (*W*) and stopping criteria. Then, a number of particles are generated

according to the population limit, each somewhere within the parameter space defined by the limits of  $\psi$  and given random initial velocities.

Each iteration begins by updating the new particle positions according to their current position and velocity. Two stopping criteria are normally used to limit the search, the first is a check on the number of iterations which have elapsed. The second is to check if at least 95% of the number particles ( $N_p$ ) have a less than 1% change in their location from the previous iteration,

$$\frac{|X_i - X_{i-1}|}{X_i} < 1\% \text{ for } N_p > 95\%.$$
(4.3)

If most particles are relatively stationary, there is a high likelihood that an acceptable minimum has been found and if not, the subsequent iterations are not likely to find any better positions. A third stopping criteria was used for populating workspace performance maps, in this case the algorithm stopped as soon as the first zero-reversal case was found. If any stopping criteria are reached, the algorithm outputs the current global best score along with the associated  $\boldsymbol{\psi}$ .

Provided the stopping criteria is not reached, the objective function analyzes each particle in the swarm. For each of these particles, the path  $V_{p+r}$  is constructed according to the particle input parameters. The path is then divided into N interpolation points. For each interpolation point (n = 1, 2, ..., N), the inverse kinematics yield 8 sets of joint angles

$$\boldsymbol{Q}_{\boldsymbol{n},\boldsymbol{k}} = [\theta_1, \theta_2, \dots \theta_6]_{\boldsymbol{n},\boldsymbol{k}}^T, \tag{4.4}$$

where k = 1, 2, ..., 8. For every configuration k, the set of joint angles fully describing the drilling motion is concatenated into a joint angle history

$$[\boldsymbol{H}_{k}]_{6xN} = [\boldsymbol{Q}_{1,k}, \boldsymbol{Q}_{2,k}, \dots, \boldsymbol{Q}_{N,k}],$$
(4.5)

This solution  $H_k$  describes a complete drilling motion in the joint space of the robot and is what will be evaluated with the objective function during optimization. The number

of discrete points is significant since if they are too widely spaced, there may be an intermediate point which is invalid and so cannot be executed by a physical robot. If the number of points is too large, the computation time becomes excessive. In this case 200 steps were used for the entire motion while searching. When the best score is found, a check is run with up to 40,000 steps to ensure convergence. For a 50mm plunge, this corresponds to a linear resolution of around 1 micron. This is likely far beyond what is necessary to ensure no intermediate points caused an error, but the added computation time for a single case, even with so many steps, is negligible.

After finding the joint angles at each point along the path, the joint angle history  $H_k$  showing the movement of each joint throughout the motion is passed into the objective function. The entire joint angle history is evaluated to find the number of reversals, the total travel, and other Boolean conditions, and a score is given between 0 and 1.

Once particles have been assigned a score, it is compared first to their personal best. If a better score has been achieved, this is saved as the new personal best. If a new personal best is achieved, that score is also compared against the swarm's global best, which is updated to the new score if it is better. When any bests are stored, both the score and location within the parameter space is saved. Finally, the velocity of the particles for the subsequent iteration is determined according to the velocity expression, which is given as

$$\boldsymbol{V}_{i+1} = r \cdot (\boldsymbol{X}_{PB} - \boldsymbol{X}_{i+1}) \cdot \boldsymbol{W}_{PB} + s \cdot (\boldsymbol{X}_{GB} - \boldsymbol{X}_{i+1}) \cdot \boldsymbol{W}_{GB} + \boldsymbol{V}_i \cdot \boldsymbol{W}_{in},$$
(4.6)

where  $W_{PB}$ ,  $W_{GB}$ , and  $W_{in}$  are weights associated with the personal best, global best, and inertia terms, respectively.  $W_{PB}$  was selected as 0.2,  $W_{GB}$  was selected as 0.1, and  $W_{in}$  was randomized at each iteration between 0.5 and 1 according to a uniform distribution. Typically,  $W_{PB} = W_{GB} > W_{in}$ . However, since the optimization was constrained, there was no concern for particles shooting away too far so a higher inertial weight allowed further exploration of the parameter space.  $W_{PB} > W_{PB}$  was done since, after algorithm execution, it

allowed for better visualization of locations of local minima clusters. For a detailed analysis of the possible weight distributions in PSO and their effects, see [28]. The variables *r* and *s* are randomly generated at each iteration between 0 and 1 to represent a particle's adherence to each influence. This random weight scheme improved the convergence rate, while the higher personal best weight ensured particles clustered at numerous minima as opposed to all seeking the first global best.

This optimization methodology is trajectory agnostic, which is beneficial since it allows for the same method to be applied to any robot controller without needing knowledge or control over the trajectory. Often times, control algorithms which improve performance require an override of the standard control architecture within a given robot controller. This makes it difficult to implement in an industry setting where the expertise, resources, and time required are prohibitive. Instead, the optimization methodology proposed has results which are easily translated into robot instructions. A point-to-point move can locate the robot at the first position, followed by a linear translation  $V_p$  of the plunge depth value with some associated rotation about  $Z_{TCS}$  corresponding to  $\psi_t - \psi_s$  followed by another linear motion along  $V_r$  of  $\psi_e - \psi_t$ .

# 4.3 Objective Function and Optimization Parameters

The optimization input parameters are  $\psi_s$ ,  $\psi_t$ , and  $\psi_e$ . Since all  $\psi$  parameters are bounded between ±180°, this creates the parameter space, or the volume which will contain all the particles. This is the only explicit constraint on the input parameters. In addition to these limits, a velocity restriction can be put in place which limits the maximum displacement of a particle between two iterations. This is useful when refining the search near a specific point (with some initial guess) however was not used when searching the entire parameter space. The optimization parameters (particle locations) are initially distributed throughout the parameter space semi-randomly.  $\psi_s$  and  $\psi_t$  are randomly selected, then  $\psi_e$  is randomly selected within the remaining volume of the workspace which will ensure there is no inflection. That is,  $\psi$  throughout the motion is either monotonically

increasing or decreasing. This was done since a reversal in  $\psi$  was overwhelmingly likely to introduce a reversal in the robot joints, usually the wrist.

The objective function which evaluates the performance of the particles must not only evaluate some given robot position, but the entire drilling motion constructed from the set of  $\boldsymbol{\psi}$ . All positions are known, and only the orientation is undefined. Therefore, when given the orientation at the three known points, the entire motion is fully defined by assuming a constant rate of rotation from one position to the next.  $\boldsymbol{\psi}$  at any point along the path is given in (3.20). After the IKs are applied to each point in the path, the resulting joint angles for the entire motion create six smooth profiles, together called the joint angle history  $H_k$ .

As an objective, this thesis aims to minimize joint reversals first. Then, within solutions with the minimum number of reversals, discriminate between solutions by minimizing total joint travel. Therefore, the weight on the number of reversals,  $W_{rev}$ , was set as 6/7 while the weight for joint travel,  $W_{trv}$ , was set as 1/7. This ensured that a solution, no matter how much joint travel it had, would be favored over any other solution with more joint reversals.

The final resulting objective function is

$$f(\boldsymbol{H}_{\boldsymbol{k}}) = \left(\frac{N_{rev}}{6} \cdot W_{rev}\right) + \left(\frac{\theta_{trv}}{6 \times 360^{\circ}} \cdot W_{trv}\right) + B_{jump} + B_{limit} + B_{fault},$$

$$(4.7)$$

where  $H_k$  is some generated joint angle history,  $N_{rev}$  is the number of reversals,  $\theta_{trv}$  is the total joint travel in  $H_k$  and the remaining terms are Boolean conditions which excluded invalid solutions. The maximum possible normalized value of the first two terms in (4.7) is 1. Therefore, any Boolean conditions which returned True as 1 automatically achieved the worst possible score and were excluded. These terms are another way of constraining viable input parameters. It is difficult to predict and relate from just the values of  $\psi_s$ ,  $\psi_t$ , and  $\psi_e$  where the inverse kinematics will be unsolvable, violate joint limits, etc. (and of the 8

solutions some may be viable while others aren't). Therefore, it becomes more practical to parse the invalid results after evaluating the path generated by  $\psi_s$ ,  $\psi_t$ , and  $\psi_e$  than to determine which combinations to preemptively exclude.  $B_{jump}$ ,  $B_{limit}$ , and  $B_{fault}$  were assigned if there were any discontinuities in the generated path, if any joints exceeded their hardware limits, or if there was no valid inverse kinematic solution found, respectively.

If additional criteria were desired, such as including stiffness or minimizing torque (and thereby power consumption), these would simply be added as another term to the objective function. Whatever metric used should be normalized between 0 and 1, then term weights adjusted as desired.

In addition, no joints in the plunge motion can be stationary as the goal is to avoid stiction at the target position. This would not happen if a joint remained stationary during the plunge then began motion during the retract. Physically, this issue would still be present at very small velocities as well, so for best practice the absolute velocity should be maintained above the static friction threshold. Since the methodology is trajectory agnostic (velocity is not included) this cannot be specified as an angular velocity, but a minimum angular difference between adjacent steps was set as 0.005°/step.

# 4.4 Simulation of a Sample Drilling Path

A target was selected for trials with the target position  $P_t = [500,300,800]_{GCS}$ . This position in the workspace is one where the manipulator is relatively dexterous, and represented a fairly typical drilling pose and direction. Dexterity refers to positions which are far from potential obstacles, singularities, joint limits, edge of the workspace, and can generally rotate freely about the target point in multiple axes. This metric can be quantified through use of the Jacobian but is not required here. A plunge vector  $V_p$  of  $[50,0,0]_{GCS}$  was selected, and N = 200 steps were used in the interpolation over the entire path distance  $||V_{p+r}|| = 100mm$ . The optimal solution was verified at N = 40,000 steps to ensure numerical approximation did not smooth over invalid points. The optimal solution  $\psi_{op} =$ 

 $[14.16^{\circ}, -5.94^{\circ}, -30.46^{\circ}]$  was output after 280 seconds on an i5 3.5 GHz CPU. The convergence of particles is shown in Figure 10. At iteration 0, all 100 particles are distributed pseudo-randomly throughout the parameter space and their initial scores are evaluated. After 25 iterations, the particles have begun to cluster near regions which have shown good scores. After 50 iterations the algorithm stopped, and the majority of particles are clustered around a few local minima. The current global best is shown in red, and the rest of the particles are colored according to their current score (scale in Figure 11).



Figure 10. Location of particles within the PSO algorithm at iterations 0, 10, 25, and 50.

At iteration 0 the distribution of particles is not uniform throughout the entire parameter space. There is a noticeable skew to place particles near the diagonal  $\psi_t = \psi_e$ 

plane. This is because when generating the initial particle positions first  $\psi_s$  is randomized (explaining the uniform distribution along the  $\psi_s$  axis). Then,  $\psi_t$  is also randomly distributed (again, along the entire axis between the bounds). However,  $\psi_e$  is then generated between  $\psi_t$  and the upper or lower bounds of the parameter space such that

$$sign(\psi_t - \psi_s) = sign(\psi_e - \psi_t), \tag{4.8}$$

continuing the trend from  $\psi_t$  to  $\psi_s$ . This ensures the direction of rotation of the tool is consistent throughout the motion. Initial trials found that solutions which exhibited a reversal in  $\psi$  were far more likely to have a joint reversal, especially in the wrist. Therefore, in the interest of obtaining more feasible initial guesses, these cases were excluded. However, there is no restriction preventing any particle from moving into a region of the parameter space where (4.8) is not true in subsequent iterations, as is evident from the distribution of particles at iteration 10 in Figure 10.

The final scores of all particles are plotted according to ascending scores in Figure 11.



#### Figure 11. Score distribution of 100 particles after 50 iterations of PSO algorithm.

The width of a single-colored bar shows how many particles achieved the number of reversals corresponding to that score step. In this test, of the 100 particles, 40 were able to find a set of parameters with zero reversals in the iterations allotted. This figure helps illustrate the echelons of the objective function, as well as the general performance of the optimization model. The number of reversals corresponds to the large steps, while the gradual incline in each is the different amount of total joint travel for each particle which distinguishes solutions within a given strata. This technique of visualizing the swarm at the beginning, midpoint, and end of algorithm execution then graphing the performance of particles will be used in Section 4.5 to tune the parameters of the optimization to ensure consistent results could be achieved.

# 4.5 Performance Tuning

PSO has a number of parameters which can be tuned to influence the ability of the algorithm to obtain the global minimum. This section will detail the tests carried out to find values for these parameters which ensured efficient convergence of the algorithm for subsequent tests.

The purpose of this tuning was not to obtain the best possible set of parameters with respect to convergence rate and computational efficiency. A much more in-depth evaluation of parameters and optimization algorithms would be required for that goal. The optimization methodology presented here is a means to the result of a drilling motion with zero reversals. It is not intended to be taken as the exclusive or optimal approach. A sufficiently robust algorithm will be able to find zero-reversal cases somewhat efficiently, but it is not a real-time control strategy. A faster and more accurate approach may be found with further research into the specific objective function, alternative optimization algorithms, and parameter tuning. Nevertheless, the process of selecting parameter settings is shown to demonstrate some of the considerations which went into selecting these values.

During initial testing of the algorithm, it was recognized that a constant inertial scheme was not suitable for this application. While the implementation of PSO does not guarantee convergence, tests with a constant inertial scheme often converged at local minima with greater than zero reversals. Multiple trials with the same parameters were required to definitively say that a zero-reversal case could not be achieved. Furthermore, particles often became stuck in a repetitive motion from iteration to iteration, either oscillating about some point or consistently stepping in the same direction at the velocity limit. There are numerous other inertial weight schemes for PSO detailed by Bansal [35]. As a conclusion, they recommend that a chaotic inertia weight be selected for improved accuracy, and a random inertia weight for computational efficiency. Other schemes offered certain advantages, and so were tested as well against the benchmark implementation of the

random inertia weight (Figure 12). The trials which were conducted, and the associated tuned parameters, are shown in Table 3.

Trial #	Pop. Size	Iterations	Inertia Scheme	$W_{PB}$	$W_{GB}$
1	200	25	$W_{in} = 0.5 + \frac{rand()}{2}$	0.2	0.1
2	200	25	$W_{in}(t) = 0.95 \times W_{in}(t-1),$ $W_{in}(0) = 0.7$	0.2	0.1
3	100	50	$W_{in} = 0.5 + \frac{rand()}{2}$	0.2	0.1
4	50	200	$W_{in} = 0.5 + \frac{rand()}{2}$	0.2	0.1
5	100	50	$W_{in}(t) = 1.05 \times W_{in}(t-1),$ $W_{in}(0) = 0.1$	0.2	0.1
6	100	50	$W_{in} = 0.5 + \frac{rand()}{2}$	0.1	0.2
7	100	50	$W_{in} = 0.1 + \frac{rand()}{2}$	0.2	0.4
8	100	50	$W_{in}(t) = 0.7 - 0.7 \times \frac{t}{50}$	0.2	0.1

# Table 3. Summary of PSO parameter values during tuning trials.

Trial 1 yielded promising results, with a significant number of particles achieving the zero-reversal case, while having still searched a large overall portion of the workspace. However, no clear clusters of local minima emerged. The sheer number of individuals meant that a large region of the parameter space was saturated with particles. This can be seen as rather computationally inefficient, despite having the potential to find very small regions of local minima.



Figure 12. Trial 1 particle convergence and performance.

Implementing a decaying inertia weight scheme (Figure 13) resulted in a similarly sized portion of the workspace being searched, however most particles failed to converge in the allotted number of maximum iterations, with some even failing to find a single valid solution even with six reversals (the portion of the six-reversal bar with a score of 1). Again, no clear local minima emerged and having many particles close to one another is inefficient.



Figure 13. Trial 2 particle convergence and performance.

While a larger population size was desired to search a larger workspace, through these trials and visualization of the objective function with set parameters (see Section 4.7) it became evident that well performing regions were not so small as to necessitate this population size. It was sufficiently likely that, given the typical size of the zero-reversal region, fewer numbers were needed to eventually find a minimum. Trial 3 (Figure 14) reverted to a random weight scheme and instead reduced the number of particles to 100 and increased the maximum number of iterations to 50. With this, the plurality of particles found a zero-reversal case.



Figure 14. Trial 3 particle convergence and performance.

To further stretch this trend, only 25 particles were used over 200 iterations in Trial 4 (Figure 15). In this case, particles either found a zero-reversal case or became noticeably trapped at local minima. Also, the initial distribution of the swarm with so few particles creates a rather sparse coverage of the parameter space. This makes this scheme more susceptible to unfortunate initial starting locations which could lead the swarm towards a local minimum. The 100 particle 50 iteration combination of Trial 3 seemed to strike a good balance between computation time, area searched, and percentage of particles which found a zero-reversal case. For subsequent trials, the Trial 3 combination was used.



Figure 15. Trial 4 particle convergence and performance.

An unconventional increasing weight scheme was tested in Trial 5 (Figure 16). The theory behind this method was to have the first iterations heavily affected by the particle's initial random positions. Since the particles began distributed throughout the workspace, each particle would spend more iterations searching their local region before converging towards the best. However, this offered no significant benefit over the new best performing candidate.



Figure 16. Trial 5 particle convergence and performance.

Trial 6 (Figure 17) tested the results of having a larger social bias than cognitive bias. This encouraged particles to move towards the global best and predictably resulted in most of the particles converging immediately towards the best without significant searching of the workspace. While the result is promising in this trial, for those cases where the feasible zeroreversal region may be smaller or have additional local minima traps nearby this behavior could be deleterious. For this reason, it was not favored over those schemes which had a larger cognitive bias.



Figure 17. Trial 6 particle convergence and performance.

Trial 7 (Figure 18) investigated decreasing the ratio of inertial weight to best location weights. This meant particles tended to change direction quickly towards the global best. PSO traditionally applies this weight scheme, and while this strategy produced a significant number of particles achieving zero reversals, it had not searched a large region of the workspace. This scheme could be useful when implementing objective functions with additional criteria, such as minimized compliance. Since the first particle to find zero reversals will greatly attract all others, there will be more searching near that zero-reversal case to find a solution which minimizes secondary objectives. However, this scheme is not promising for searching an objective function where many local minima not at zero reversals could trap the swarm.



Figure 18. Trial 7 particle convergence and performance.

Finally, a decaying inertial scheme was tested in Trial 8 (Figure 19). This had the advantage of exploring the workspace widely during initial iterations and converging on the best locations later. However, after exploration the particles did not sufficiently converge, and so only a small number achieved the zero-reversal case.



Figure 19. Trial 8 particle convergence and performance.

After consideration of the available weight schemes, testing options, and taking recommendations from literature, the weight scheme associated with Trial 3 was selected for use in the optimization algorithm for future tests.

Trial #	Pop. Size	Iterations	Inertia Scheme	W <sub>PB</sub>	W <sub>GB</sub>
3	100	50	$W_{in} = 0.5 + \frac{rand()}{2}$	0.2	0.1

#### Table 4. Weight scheme used for subsequent tests.

This was efficient to run, searched a large region of the workspace, and resulted in a large number of particles converging on a zero-reversal minimum. Moving forward, this scheme was used to search numerous points throughout the robot workspace and create a map of locations where zero-reversals could be achieved.

# 4.6 Workspace Mapping

In Section 4.4, the scope was limited to only finding the best solution at a single position in the workspace. However, it is also beneficial to take a broad view and understand how the position of the target and the plunge depth in the workspace affects the number of reversals which can be achieved. To this end, the same optimization was performed throughout the robot workspace, with decreasing plunge depths. Once it was realized that plunge depth would affect the minimum number of reversals achieved, the next logical question was to find the longest plunge depth at each location which still allowed for zero reversals. So, at each location the plunge depth was iterated according to a bisection method bounded between 0-200mm. After each iteration, the plunge depth  $D_p$  was changed according to

$$D_p = R_{p1} + \frac{R_{p2} - R_{p1}}{2},\tag{4.9}$$

where  $R_{p1}$  and  $R_{p2}$  refer to the lower and upper bounds of the plunge depth. Before computation of the plunge depth for the subsequent iteration, the bounds were replaced according to the previous best result. If zero reversals had been achieved, then  $R_{p1} = D_p$ , else  $R_{p2} = D_p$ . To achieve a resolution of at most  $\epsilon = 1mm$ , this process was iterated 8 times according to

$$n \ge \frac{\log\left(\frac{R_{p2} - R_{p1}}{\epsilon}\right)}{\log(2)} = 7.64.$$

$$(4.10)$$

The PSO algorithm followed the technique outlined in Section 4.2, however an additional stopping criterion halted the program when any viable set of parameters was found which obtained zero reversals. This significantly reduced computation time, yet to obtain data for a grid of 81 different locations in the workspace still required 2 days of processing time on an i5 3.5GHz CPU.

The results of the first mapping at X = 500mm are shown in Figure 20, and detailed in Table 5. The results of this search were more coarse owing to a simpler plunge depth searching algorithm which simply decremented the plunge depth by 5mm at each iteration. As can be seen, the results are symmetrical about Y = 0 within  $\pm 10$ mm, owing to the potential to miss an instance of a 0-reversal minimum since PSO does not guarantee global convergence. However, with the knowledge that the performance seems to be axisymmetric about the base, further trials utilized only one half of the workspace. This makes sense intuitively, since we would expect the mirrored version of the motion to achieve the same result provided there are no additional obstacles or limits on the robot.

		$Y_{GCS}(mm)$								
		400	300	200	100	0	-100	-200	-300	-400
$Z_{GCS}(mm)$	1200	0	0	0	0	0	0	0	0	0
	1100	0	0	0	0	15	0	0	0	0
	1000	0	40	70	60	50	60	75	40	0
	900	75	120	85	65	45	70	95	120	70
	800	105	90	80	75	65	85	95	100	105
	700	80	110	100	90	70	95	110	110	95
	600	60	45	35	25	30	30	30	40	70
	500	40	25	20	25	20	10	20	30	40
	400	40	50	35	15	0	20	35	45	40

Table 5. Map of maximum plunge depth (in mm) still capable of achieving zeroreversals on the X = 500mm plane.

Figure 20 shows a subsequent map of this same plane. The surface is offset in the X direction by the maximum plunge depth amount at that point to visualize the topology, and colored according to this amount. A wireframe representation of the robot at the home position is also shown to demonstrate orientation.



Figure 20. Mapping of maximum plunge depth to achieve the zero-reversal case at X=500mm.

As seen in Figure 20, the regions which could still obtain zero reversals despite a long plunge depth are those in the upper corners of the workspace. The region directly ahead of the robot saw poorer performance, indicating that installing the robot with an offset, rather than directly in front of the workpiece, could prove beneficial in practical applications. Furthermore, there is an abrupt decrease in performance above Z = 1000mm, near the upper boundary of the tested area, while there is a more gradual decline in performance along the lower bounds.

Figure 21 shows the same mapping procedure results repeated for the planes X = 100 mm, 200 mm, 300 mm, 400 mm, and 500 mm. The same colormap is used for all plots, however the same color (maximum plunge depth which achieved zero reversals) on one plot does not necessarily correspond to the same color on another plot. This relative rather than absolute coloring helps distinguish finer features which may be missed otherwise.



Figure 21. Workspace maps showing a visual representation of the maximum plunge depths still able to achieve zero reversals for planes from X = 100mm to X = 600mm.

When comparing the performance along the plane X = 300mm against the previous X = 500mm case, the 300mm case the performance changes much more gradually towards outer edges of workspace. Additionally, the choppy peaks and valleys from before are replaced by a much smoother function with the best performance near the top and middle of the workspace, rather than off to the sides. This trend seems to be consistent, with the tapering off of the plunge depth to the sides of the workspace being more pronounced at X = 600mm and more gradual moving towards X = 200mm. This is especially evident when all planes are plotted simultaneously, and viewed from the bottom as seen in Figure 22. From this perspective, the robot wireframe runs along the top of the figure and the planes are all situated to the left and ahead of the robot base. In this case the colormap is absolute.



Figure 22. Workspace map of maximum plunge depth capable of achieving zero reversals viewed from the underside of the XY plane.

When viewing all planes in the workspace map from the side of the robot as in Figure 23, another trend becomes apparent. The region of best performance shifts downwards relative to the robot as the target moves farther away. For reference, the best zero-reversal result of 125mm was achieved at [500, 300, 900], marked with a red circle.



Figure 23. Side (XZ) profile of workspace map with heights/colors representing the maximum plunge depth still capable of achieving zero reversals when drilling in the positive X direction.

From Figure 23 it is apparent that the region of the workspace searched for these tests does not completely enclose the viable zero-reversal region. A more complete map of the workspace could glean further insight into performance trends. However, the region searched still represents the commonly applied area for robotic drilling (forward, into a workpiece).

Evident from both Figure 22 and Figure 23 is the odd performance at X = 100mm. Rather than continuing any previous trend, instead the performance is exceedingly

discontinuous and sparse throughout the plane. It is unclear exactly what causes the performance to change so abruptly. The drilling direction is still the same (positive X direction) but at a more awkward posture. By repeating the mapping process for additional planes X = 0mm and X = -100mm in Figure 24 it is evident that the performance is smoothed out when the target is moved even farther back. It is likely a quirk of the inverse kinematics where the drilling motion is difficult to perform when directly beside the robot.



#### Figure 24. Side (XZ) profile of workspace map at planes X=-100, 0, and 100, with heights/colors representing the maximum plunge depth still capable of achieving zero reversals when drilling in the positive X direction.

Beyond the previously simulated sample location (Position 1), two other locations were selected as candidates for experimental trials. These were located on the X = 200mm plane. Position 1 represented a target where the robot reached forward, and Position 2 represented a point "beside" the robot and Position 3 a point where the robot reached "up" (while both still being ahead slightly). All of these trial points/positions are represented in

Figure 25. There are certainly other locations of interest which should prove fruitful in future investigation, such as the very long plunge depths in X = 500mm plane, the individual peaks in the X = 200mm plane. But for the purposes of this thesis, points were selected to establish some baseline for performance improvement and an understanding of the steps for implementation. For this, more representative points not at any extremes but at mundane yet sufficiently diverse locations were preferred.



Figure 25. Location of experimental trial positions within the robot workspace.

While it is of great interest to quantify the performance of this robot with respect to the potential to achieve zero reversals, the analysis done here represents only the case of horizontal drilling with an end effector of the specified dimensions. However, by rotating about the base, a skewed drilling motion becomes functionally identical to an ideal  $\Delta X_{GCS}$  plunge at some point beside the robot, making it suitable for many practical drilling applications. With that said, a different robot, with different end effectors and different kinematics will no doubt vary in its ability to achieve zero reversals throughout the workspace and so the trends presented here should not be applied to other cases

indiscriminately. However, by analyzing these trends insight can be gained into the nature of the problem and the findings can be applied more deliberately. For example, when creating a robot cell, it could be worthwhile to adjust the relative position of the workpiece to the robot to ensure that the maximum plunge depth the robot will use can still achieve zero reversals. This could potentially be done by moving the workpiece down as little as 200mm in some cases to go from no solutions with zero reversals to solutions with zero reversals at a plunge depth of 120mm. Furthermore, the multiple robots could be positioned in a cell such that their workspaces overlap. In this way, the poor performing regions of one robot could be covered by the well performing region of another robot. These robots could drill alternating sections of a long series of holes so that they are always operating in their most capable regions.

# 4.7 Parameter Mapping

Section 4.6 took a view which was broad in scope of the workspace (possible target locations) but narrow in only showing the best result of the optimization model at each point. This section will instead take a narrow view of the workspace (selecting a single target point) but show broadly how the objective function behaves according to the possible combinations of input parameters. This will check the suitability of the selected optimization model, as well as offer insight into future improvements which can be made.

The motivation for selecting PSO was in part due to the unknown and likely irregular behavior of the objective function. And up to this point, the exact behavior of the objective function is still relatively unknown. To remedy this, the objective function was mapped for a specific point in the workspace, (500, 300, 800), with a plunge depth of 50mm. A 100x100 grid of points for  $\psi_s = \pm \pi$  and  $\psi_e = \pm \pi$  was created for each of 100 points between  $\psi_t =$  $\pm \pi$ . The objective function was evaluated at each combination of parameters for the 8 configurations and the single best score of those selected for representation. While still difficult to visualize, a cross section at  $\psi_t = 0$  gives insight into the behavior of the objective function. Figure 26 shows the objective function results for a fixed target point, plunge depth, and  $\psi_t$ , with horizontal and vertical axes corresponding to  $\psi_s$  and  $\psi_e$  respectively.



Figure 26. Objective function map at (500, 300, 800) plotted according to start and end psi angles, with color showing the objective function value achieved.

The performance appears mostly symmetrical about the  $\psi_s = \psi_e$  diagonal, apart from a missing "block" above  $\psi_e = -1$ . This means that, for the most part, some set of parameters performs equally well if the twist angles at the start and end are switched. This follows intuitively – if some path has no reversals in one direction, it should have no reversals when mirrored. However, there is clearly a region wherein this is not true. The cause of this is the joint limits imposed by the hardware of the robot. For the case where  $\boldsymbol{\psi} = [-1,0,1]$  one of the eight possible configurations are valid and produces a solution with 3 reversals. However, for the case of  $\boldsymbol{\psi} = [1,0,-1]$  the previously valid configuration is eliminated

because it exceeds joint limits. The reasons for this come down to the kinematics of the robot, the cyclical nature of revolute joints, and how a given configuration selects a starting set of joint angles. In this implementation of the inverse kinematics, each point in the path is calculated independently. Therefore, there is no knowledge of the subsequent joint angles. Because some joint angle  $\theta$  can be expressed as

$$\theta = \theta \pm 360^{\circ} \times k, k \in \mathbb{Z}, \tag{4.11}$$

then the same set of joint angle histories which are functionally identical can be expressed in a way in which one is valid while the other violates joint angle limits.

If a motion calls for  $\theta$  to rotate  $-200^\circ$ , it is better to begin the motion with the value of  $\theta$  as high as possible, since given some range of  $[-185^\circ, 185^\circ]$  (such as the case for joint 4 of the robot in question) starting below 0 will certainly result in the motion exceeding joint limits. Conversely, in the case where the motion calls for a rotation of  $+200^\circ$  in  $\theta$ , it would be better to start as low as possible to avoid exceeding joint limits. But when calculating the initial point, the kinematics do not know what the rest of the motion will be and so cannot compensate accordingly. A wrapping function is in place to keep joint angles close to zero, but there is no predictive ability of the inverse kinematics to strategically select the expression of an initial joint angle which will keep the subsequent motion within joint limits. This is an interesting aspect of motion planning, and may be in place within the controllers of industrial robots. However, since it was not the main focus of this research and the existing inverse kinematic implementation was performing well, time was not dedicated to changing the procedure. When manually verifying points within the "blocked out" region, identifying which joints exceeded limits and shifting these joints by  $\pm 360^\circ$  produced identical results to the symmetrical point such that

$$f(\boldsymbol{H}(\psi_s,\psi_t,\psi_e)) = f(\boldsymbol{H}(\psi_e,\psi_t,\psi_s)).$$
(4.12)

But if this manual shifting were preemptively applied to the inverse kinematics, invariably the mirrored point would instead exceed joint limits.
### Chapter 4 | Optimization of Drilling Motion

The specifics of inverse kinematics and motion planning aside, a robot can be assumed to exhibit this symmetrical property for the objective function defined if more robust path planning is applied in the robot controller. This property could be exploited in two ways. The first is to significantly reduce the viable parameter space to search. The valid bounds of  $\psi_s$  and  $\psi_e$  could be adaptively altered to, for example, only examine cases where  $\psi_s \leq \psi_e$ . This could significantly reduce computation time and improve convergence by eliminating competing local minima on the opposite side of the plane of symmetry. The competing local minima are especially evident in **Error! Reference source not found.**. The second way that the symmetry could be exploited is in chaining motions together. For example, if a series of nearby holes must be drilled in quick succession. As has been discussed it is feasible to use the same  $\psi$  for neighboring points  $P_1$  and  $P_2$ . Then, it is more efficient to begin each subsequent motion with the same orientation as the end of the previous motion such that

$$\boldsymbol{\psi}_{P_1} = [\psi_{s_1}, \psi_{t_1}, \psi_{e_1}],$$

$$\boldsymbol{\psi}_{P_2} = [\psi_{e_1}, \psi_{t_1}, \psi_{s_1}].$$

$$(4.13)$$

This could be exploited to reduce computation time, but also to select when in the motion more rotational movement is desired. If

$$|\psi_t - \psi_s| \gg |\psi_e - \psi_t|, \tag{4.14}$$

then for that drilling motion the change in orientation is greater during the plunge. If this proved disadvantageous, such as by increasing the actual RPM of the drill bit beyond a desirable threshold, then instead  $\psi_t$  could be replaced by  $-\psi_t$  resulting in

$$|\psi_t - \psi_s| \ll |\psi_e - \psi_t|. \tag{4.15}$$

This would keep the tool in a more fixed orientation while drilling the hole, and allow the necessary tool rotation to occur during the retract motion when no additional material is being removed.

### Chapter 4 | Optimization of Drilling Motion

While further insight would surely be gained from analyzing the objective function at many points throughout the workspace, it is simply time prohibitive to examine every combination of target location, plunge depth, and  $\psi$ . This is a possible future area of research. A limitation with this approach, of course, is its applicability to other manipulators. Since the results are not being rigorously proven, they cannot be universally extrapolated to other robots, especially those with significantly different geometries.

## 4.8 Summary

This chapter has shown how an appropriate optimization algorithm can be applied to an objective function with the goal of minimizing the number of joint reversals during a drilling motion. In addition, the effect of the target position in the workspace and plunge depth were explored and showed significant pose dependency on the achievability of the zero-reversal case. Finally, for a given target point the objective function was mapped for combinations of  $\psi_s$ ,  $\psi_t$ , and  $\psi_e$  throughout the parameter space.

## **5** Experimental Validation

Following the simulation of numerous drilling scenarios throughout the workspace, tests were conducted to ensure instructions could be translated easily, were achievable, and produced the desired outcome with beneficial results. Section 5.1 will discuss the design of these experiments, Section 5.2 will demonstrate the repeatability of the robot, and Section 5.3 will show the results at the first trial point. Section 5.4 will detail and discuss the results of experiments carried out at other trial locations, and the chapter will conclude in Section 5.5 with a discussion of other effects of the optimization not directly measured.

## 5.1 Design of Experiments and Experimental Implementation

The primary objective of the experimental tests was to determine if all assumptions and modelling carried out in the simulation were sufficient to accurately reflect the physical system. A secondary objective of the tests was to acquire data on the performance of the robot so that cases with and without optimization could be compared. Specifically, the angular position and torque of each joint, as well as the GCS position of the end effector.

The experimental validation was carried out on a KR 6 R700-2 robot with KR C4 Compact controller. Instructions were programmed as .SRC files and uploaded to the controller through WorkVisual after being written using KUKA Robot Language (KRL). A sample drilling end effector was 3D printed using PLA on a Creality CR-10S. For better visualization of the twist of the tool, a rectangular prism with alternately colored faces replaced the typical cylindrical drill bit.

No physical part was drilled as the additional forces and dynamics in this process are beyond the scope of this investigation. Additionally, the feed rate, RPM, bit selection, and material are all likely to affect the overall performance. A generalized approach was taken for which results can be reasonably extrapolated to more specific scenarios.



Figure 27. Robot at Position 1 starting location with simple drilling end effector with colored rectangular prism for rotation visualization.

Data were recorded through the KUKA option package Robot Sensor Interface (RSI) and KUKA Trace Recording were used to store data from the controller for later analysis. Actual and target values for joint and cartesian position, velocity, and acceleration were obtained. The recorded data were imported into MATLAB, then parsed and plotted for visualization and analysis purposes.

All tests in any comparison experiments were carried out within a short time frame of each other, usually immediately following, to reduce the impact of other factors on the results, such as the state of the robot internal mechanisms and ambient environment.

To initiate the tests, first the simulated results output a set of  $\boldsymbol{\psi}$  for each trial location which corresponded to the start, target, and end angles for the drilling motion. The joint values which corresponded to the start position were used as the first point to point (PTP) instruction. Then, two linear commands (LIN\_REL) moved the robot to the target, then end positions. During these commands, the plunge depth was used as the translation components in  $Z_{TCP}$ . The rotational component about  $Z_{TCP}$  was  $\psi_t - \psi_s$  for the plunge motion, and  $\psi_e$  –

 $\psi_t$  for the retract. After coming to a rest, the robot was reset at the starting position. Appendix A provides a KRL .SRC file which was used for the experiments. All trial motions were programmed in this script, and the trial to run was selected by changing the TrialPicker and ModePicker variables. The programmed velocity was constant for all trials.

## 5.2 Repeatability Studies

Robots are well known for their excellent repeatability. Even if the accuracy of a motion is only within an error of  $\pm 0.2$ mm, the repeatability itself is usually an order of magnitude lesser ( $\pm 0.02$ mm). Nevertheless, a simple repeatability study was conducted to show that results could be compared sufficiently well without taking an average over multiple trials. The focus of these tests was to verify the stated repeatability of the robot and ensure theoretically identical trials showed good comparison. If the focus were to precisely identify repeatability then many additional tests would have been performed with numerous payload cases in a variety of positions, however for the actual purposes here two trials were sufficient.

The first test was to perform the same motion twice and compare the results. The commanded motions were not executed in the same program, but rather the program itself was executed twice. The joint torques for both motions are overlaid in Figure 28.



Figure 28. Joint torque comparison in short-term repeatability study.

The manipulator showed excellent repeatability, and the torque is shown here to acknowledge that there are still slight differences along the path, but the results are sufficiently similar with no torque difference greater than 0.001Nm. The joint angular and cartesian positions, which are not shown, are within manufacturer specifications of repeatability and show no meaningful difference at the visualization scale. A similar test in a different pose was done and the data saved for eventual comparison. This trial was repeated 4 months later, and the data compared against the previous test, and this long-term comparison again showed excellent repeatability as shown in Figure 29.



Figure 29. Joint torque comparison in long-term repeatability study.

Together, these tests showed that while not exact, the data extracted from similar tests performed either immediately or a long period apart could be compared reliably without needing to rely on a mean taken over numerous trials. However, tests immediately following each other showed less difference. The TCP position data also verified the repeatability of the robot, and future comparisons between data will show only a single trial for some given motion to reduce clutter.

## **5.3 Experimental Results**

Experimental results verified the simulated results and showed improvement in the performance of a sample drilling motion when compared to a baseline motion with fixed tool orientation.

Internal controller data were imported after tests and were aligned using the first instance of commanded torque. This value is null before program execution, but immediately obtains some value (even if zero) as soon as program execution commences. This allows for repeatable alignment of data with one another despite timing differences between releasing safety measures and starting the program.

Since there exists an unknown control structure within the robot controller, the first comparison made was the simulated motion against the actual motion being executed. This measure was necessary to ensure that the kinematic model developed was accurate and that the predicted path could be repeatably followed despite only providing start, target, and end point parameters to the controller. This comparison is shown in Figure 32, which shows there is a difference between the optimized (measured) and optimized (simulated) profiles. However, the maximum difference in joint angles is only 5%, which is accounted for in the assumptions made. First, since the simulated motion is entirely kinematic, there is no allowance for the rise time of joints to achieve the desired path. In other words, simulated acceleration is instant when in actuality there must be some delay. Another difference is the roundness at the midpoint of the motion, representing when the robot is at  $P_t$ . During robot programming, the motion to move to the target point was assigned an approximate positioning value of \$APO.CDIS = 10. This means that rather than try to place the end effector exactly at the target, the controller began to execute the subsequent retract motion when the TCP was within 10mm of  $P_t$  during the plunge motion. This was necessary, since to achieve exact positioning requires the motion to momentarily stop. Stopping, however, puts all joints into the stiction region again. So even with zero reversals in the kinematic path, every joint would still experience stiction and cause an error at the end effector. But with approximate

positioning this issue was resolved, and from analysis of the actual joint position no additional reversals were introduced into any of the joint paths by the inclusion of approximate positioning.

The executed motion follows the path described by the parameters

$$\boldsymbol{\psi} = [14.16^{\circ}, -5.94^{\circ}, -30.46^{\circ}], \tag{5.1}$$

The motion began and ended at

$$P_s = P_e = \begin{bmatrix} 450\\ 300\\ 800 \end{bmatrix} \text{mm,}$$
(5.2)

with a plunge depth of 50mm. The overall posture is shown in Figure 30, and a close up of the start, target, and end positions is shown in Figure 31.



Figure 30. Robot at Position 1 target.



Figure 31. Drilling motion of robot at start, target, and end positions.

Before even executing the motion, immediately noticeable was the selection of an alternate kinematic configuration which is unintuitive to the robot programmer and would certainly not have been found using conventional programming methods. Figure 30 shows the robot in a position where joint 1 has been spun nearly 180°, with the arm reaching back towards the target. This upside-down posture serves to highlight the inherent biases apparent when programming robots manually. Humans will naturally assign a top, front, and side to the tool, workpiece, and robot which can hinder the imagination when programming a path.

Further demonstrated by the optimized selection of this motion is the advantage of *not* including joint limits as an optimization criterion. Joints 1 and 5 come within 20° and 10° of their limits respectively during this motion. The joint limits impose restrictions on the objective, but all other factors being equal, will not favor or discourage a point very near the joint limits to one at the midpoint of all joint ranges. Culling the out of bounds solutions after evaluation, rather than avoiding joint limits altogether, allows for solutions which are near the joint limits, yet still valid.



Figure 32 compares the baseline motion to the optimized motion.

Figure 32. Comparison of joint angles in baseline, optimized, and simulated optimized motions.

The baseline motion was defined as a motion with the same value of  $\psi_s$ , and where  $\psi_s = \psi_t = \psi_e$ . In this case, there is an obvious reversal of the joints when the target is reached. For this and future figures, path distance is defined as follows in Figure 33.



Figure 33. Visualization of path distance as a function of plunge and retract distance.

The next data to compare is the difference in actual joint torque between the optimized and baseline motions in Figure 34.



Figure 34. Comparison of joint torques in the baseline and optimized motions.

There is a significant change at the beginning of the motions as the joints break stiction. This is common to both motions and to be expected since both tests began from rest. Then, as can be seen the plunge follows the same general profile for both cases (although reflected about the starting torque value if the joint direction is different). Again, this is to be expected since no joints have yet undergone a reversal. However, at the point of transition between plunge and retract, the torque profile remains much more constant in the optimized motion. This shows that joints are not falling back into the stiction region during the optimized motion.

Next, the actual path followed by the TCP is examined in Figure 35. To ascertain the impact that avoiding reversals has on the end effector motion, the measured position of the baseline and optimized motions were overlaid.



Figure 35. Horizontal and vertical profile comparison of TCP position between baseline and optimized motions at Position 1.

While both motions are comparable in the XZ plane, there is a significant difference in the XY plane as seen in Figure 35. Both motions exhibit a kick when breaking stiction to commence the motion before settling, as well as when the motion is reversed. However, while the magnitude is comparable for the beginning of the plunge, the baseline motion exhibits significantly larger error at the target with the kick of the optimized motion being significantly less pronounced. The plane of the error is a function of the joints which are predominantly in motion, and changes with posture.

Another difference is that the optimized motion follows a similar path for both plunge and retract motions, whereas the baseline motion is offset to either side of the mean. This mean represents an average of of all data points throughout a drilling motion. The baseline mean is close to the ideal path (from [450,300,800] to [500,300,800]). However, the optimized mean is shifted to Y = 300.08mm. Even with this offset, the maximum error from the mean is within the error of the baseline motion. The instantaneous error from the mean is plotted in Figure 36.



*Figure 36. Comparison of contour error from mean in the baseline and optimized cases.* 

This represents the TCP deviation from the mean (which is near ideal Y=300mm for the baseline, and Y=500.08mm for the optimized motion) during the last 25mm of plunge into the material and first 25mm of retract out of the material. The result shows that the optimized motion with zero reversals has consistently less deviation from the mean than the baseline motion. The maximum deviation from the mean close to the target (between 25mm and 75mm of path distance travelled) during the baseline trial is 0.2mm, while only 0.12mm during the optimized trial. This represents a 40% reduction in TCP error.

The result of implementing this optimization strategy is an improvement in the precision of the drilling motion, while sacrificing approximately 0.08mm of accuracy. However, the accuracy loss is sufficiently offset by the increased precision to show overall improvement. This analysis was conducted at only a single location, but Section 5.4 will show the similarities and differences to this first trial when compared at two other locations in the workspace.

## **5.4 Alternate Optimization Locations**

Alternate drilling target locations were optimized in this section to demonstrate the feasibility of this methodology in other regions of the workspace. Further insight into the contribution of reversals to the overall end effector error is offered, and the relative performance benefits at each position are analyzed.

## 5.4.1 Workspace Position 200, 400, 600 (Position 2)

Position 2 was at the location [200,400,600] which represented a point closer to the base of the robot than Position 1, while still allowing good dexterity at the target. Figure 37 shows the robot at the target point. However, the maximum plunge depth found by the optimization algorithm was only 20mm rather than 50mm. The search for the Position 2 zero-reversal case was conducted with 5mm increments in plunge depth. This 20mm

maximum plunge agrees with the workspace mapping in Section 4.6 which predicts the maximum plunge depth that achieves zero reversals at this target to be 21.9mm.



Figure 37. Robot at Position 2 target, [200,400,600].

This trial illustrated an interesting phenomenon in a baseline case where no reorientation about the tool work axis occurs. If an operator happened to program a motion with constant tool orientation which allowed the motion to be executed with fewer joints moving, the performance would theoretically be better compared to case where all six joints reverse. Furthermore, the relative performance increase seen by adopting this zero-reversal optimization technique would decrease, since the default already shows some improvement in performance. As seen in Figure 38, joints 4 and 5 remain stationary throughout the baseline motion. There is also very little movement in joints 2 and 3. This helps to show what the "best case scenario" would be if programming the robot manually.



Figure 38. Position 2 joint angles for the baseline and optimized motions.

The absence of reversals is apparent in joints 4 and 5 by examining the joint torque during the motions in Figure 39.



Figure 39. Position 2 joint torque for the baseline and optimized motions.

Instead of a sharp change in torque, joints 4 and 5 exhibit relatively gradual changes which only serve to support the wrist as the loads shift throughout the motion, rather than to move the joints. This means that only four joints break stiction in the baseline case, and another two have very gradual motion throughout. However, even though joints 2 and 3 have very gradual changes, from the joint torque it is apparent that they must still break stiction and it is just as severe as in Position 1 where joints 2 and 3 moved significantly (Figure 32, Figure 34).

Here also is an opportunity to discuss the difference in torque magnitude for different joints. All joint-by-joint torque (and joint angle, for that matter) figures have been scaled to display well side-by-side. However, the actual torque as seen in the vertical axis markings is

not comparable between joints. For example, joints 1 and 6 appear very similar in shape, but the torque in joint 6 is an order of magnitude lower ( $\sim 0.05$  Nm compared to  $\sim 0.5$  Nm). This difference in torque is a function of the geometry, link mass and the commanded motion. Joint 6 must only accelerate the end effector, which is relatively easy. But each joint in the kinematic chain must move every link after it as well. For this reason, the average torque expected to execute some motion should to decrease from joint 1 to joint 6.

With fewer joints reversing in the baseline case, a worse relative performance increase is expected compared to when the baseline case has all six joints reversing, as in Position 1. However, this does not seem to be the case. Figure 40 shows the TCP position throughout the baseline and optimized drilling motions. The maximum deviation from the mean near the target in the baseline case is approximately 0.14mm (similar to Position 1), while the maximum deviation from the mean for the optimized motion is approximately 0.05mm. This represents about a 65% reduction in error (when measuring this specific metric). Recalling Position 1, when reducing a drilling motion from six reversals to zero, a 40% reduction in error is seen. But when reducing from four reversals to zero in Position 2, a 65% reduction is seen. Additionally, the optimized case shows increased oscillation during the retract motion, compared with the baseline. Clearly there are other factors beyond simply the number of reversals which influence the changes brought on by eliminating joint reversals.



Figure 40. Position 2 TCP cartesian position throughout baseline and optimized motions.

One of these factors is likely which joints are those actually reversing in the baseline case. Joints 1, 2, and 3 have the largest torque values throughout the motion, while joints 4, 5, and 6 have much less. It follows then that the overall error is likely caused predominantly by the first three joints rather than the last three, and so having joints 4 and 5 remain stationary does not significantly reduce the error. However, the magnitude of torque should be considered with respect to the mass of the physical components being moved. For example, a jump in torque of 5 Nm would have less influence on a 10 kg link than a 0.5 Nm jump would have on a 0.1 kg link, all other geometry being equal.

This relative difference of torque and link mass of each joint raises an interesting application for those cases where zero reversals cannot be achieved. If the required position and plunge depth dictate that the robot cannot achieve zero reversals, the reversal of each joint could be weighted differently in the optimization algorithm such that a reversal in one of the wrist joints is preferable to a reversal in the shoulder or elbow. Extending this concept, either through precise dynamic modeling or some manner of testing the joints which contribute most to the error through their reversals could be identified. Then, the weight of each reversal could be proportional, such that it may be more beneficial to have reversals in joints 2 and 3 than to have a single reversal in joint 1, for example. While this is less generally applicable (since these weights would likely vary between robot models and potentially even between identical models) it is an interesting area of potential further research.

Furthermore, a factor which is likely to play a role in the contribution of reversals to the end effector error is the alignment of each joint. This will be illustrated and elaborated on in Section 5.4.2.

## 5.4.2 Workspace Position 200, 200, 1000 (Position 3)

Position 3 was again in the upper left-hand quadrant of the robot's workspace, shown in Figure 41. This time, higher up such that not all  $\psi$  rotations about the tool were viable (values from approximately  $\psi = [-90^\circ, -180^\circ]$  were unobtainable). This position is less dexterous than the others. In this position, a maximum plunge depth of 32.8mm to achieve zero reversals was predicted in Section 4.6, and an optimized zero-reversal case with a plunge depth of 30mm was found here with

$$\boldsymbol{\psi} = [20.2431^{\circ}, -4.5475^{\circ}, -9.6903^{\circ}]. \tag{5.3}$$

Similar to Position 1, the robot's unintuitive posture (with the forearm upside-down and wrist flipped) is a feature which allows the robot to bridge the gap between the other postures required during the plunge and retract motions. With this optimization methodology switching configurations to the "regular" posture does not necessarily mean

the robot can still execute the motion, and as a result the motion may require additional reversals or be invalidated entirely as joint limits are exceeded.



Figure 41. Robot at Position 3, [200, 200, 1000].

Also similar to Position 1 is the baseline case having six reversals. The first significant difference is the behavior of joint 1 in the optimized motion. In this case, the joint is not quite stationary, moving only about 1° during the plunge motion before truly beginning motion during the retract. This can be seen in Figure 42. This is just above the minimum motion threshold set during the model creation (Section 4.3).



Figure 42. Position 3 joint angles for the baseline and optimized motions.

The effect this has on the joint torque should be noted, as instead of a significant jump when motion begins like all other joints, there is a gradual increase until the plunge where the torque remains relatively constant. Interestingly, there does not seem to be a point at which the joint jumps out of the stiction region, although it may be that in this case inertia is the dominant factor which must be overcome during the start of the motion and the torque required to set the entire robot in motion overshadows the stiction in the joint.

The remaining joints exhibit the familiar trend from the other two trials, with the baseline motion showing jumps at the start and midpoint and the optimized motion having a starting jump but a more constant profile as seen in Figure 43.



Figure 43. Position 3 joint torque for baseline and optimized motions.

The other prominent difference comes when comparing the TCP position for both cases in Figure 44. Now, instead of the most prominent error in the baseline motion being in the XY plane, it is now in the XZ plane. The maximum deviation from the mean at this position is 0.15mm, improving to 0.6mm during the optimized motion (60% reduction) In fact, the baseline performance in the XY plane is relatively good, but the optimized motion exhibits a more pronounced kick at the point of reversal. The maximum deviation from the mean near the target is similar in the XY plane for both cases.



Figure 44. Position 3 TCP cartesian position throughout baseline and optimized motions.

While it is difficult to pinpoint exactly the cause of the change in the plane of error, the likely culprit is a complex relationship between the orientation of the joints and the end effector throughout the motion. In Position 3, the end effector is positioned above the wrist, while in other trials the end effector is below the wrist. This will have some impact as the inertia of the wrist "pulls" the end effector in different directions upon beginning the retract motion, albeit in the same plane.

The main compliance of a robot comes from the joints themselves, rather than the links (which are in fact usually modeled as rigid beams). These joints oppose stiffness well in the directions they are not meant to rotate but have relatively lower stiffness in the plane which corresponds to their motion. The alignment of joints could be such that in some pose error in the horizontal plane is easier, while at a different pose error in the vertical plane is easier. Without having accurate stiffness values for each joint it is difficult to quantify the dominant source joints. Theoretically, the much lower relative stiffness of the wrist joints will contribute significantly, however this must be compared with the backlash and stiction forces which occur during reversals for each joint, as well as the relative mass of the robot in the kinematic chain after the joint.

Finally, there is the relative location of the target within the workspace to consider. In Position 3, the target is at the closest position of all the trials to the center of rotation of the base of the robot, at joint 1. But in other trials, the target moves farther away. Joint 1's contribution to the end effector error will be increased as this distance grows. However, this must be weighed by the increasing mass further from the center of rotation. A complete dynamic model of the robot would likely be able to quantify and accurately predict the plane of TCP error and relative contribution of each joint.

## 5.5 Discussion of Other Effects

While not tested empirically, the lifetime of some internal components is expected to increase when considering the effects of reducing joint reversals. Motor gears and bearings would be subject to fewer reversals and therefore fewer dithering motions where the direction of torque opposes the angular rotation of the gear [36]. Combined with other techniques such as torque minimization, this could theoretically lead to a significant reduction in foreign particles in the lubricant and lengthen the serviceable life of the robot. The time and expense of added initial optimization of motion planning being the only additional cost.

While many robots are still being trained through jogging and teaching, software such as RobotStudio by ABB has allowed for programming of a robot in a simulated environment. With this software automatically generating robot targets and motion instructions, it becomes even easier to integrate the proposed methodology into the robot programming workflow. With further improvement of the optimization algorithm itself, a point could be evaluated relatively quickly during programming and deployed with other robot instructions, further reducing the barrier to implementing this improvement on robot cells.

With a rotating drill bit, rotating the drilling spindle will cause an increase or decrease in the actual speed of the rotating bit, depending on the direction. Assuming a relatively significant change in  $\psi$  of 30° between adjacent points in only 0.5 seconds corresponds to

$$\frac{30^{\circ}}{0.5 \ second} \times \frac{1 \ revolution}{360^{\circ}} \times \frac{60 \ seconds}{1 \ minute} = 10 \ RPM.$$
(5.4)

For high cutting speeds (1000+ RPM) common in composite machining this is insignificant. However, for applications with lower cutting speeds the performance may vary as the actual rotational speed of the drill bit changes throughout the motion.

## 5.6 Summary

This chapter presented and analyzed the experimental tests used to validate the results predicted through the simulated model. The repeatability of the robot and data collections procedures were analyzed showing good repeatability, and result of implementing the simulated optimal case was shown. From the positions analyzed a 40%-65% reduction in the maximum TCP error during the material removal portion of a drilling motion is observed.

### 6 Conclusion and Future Work

This chapter summarizes the work and its impact in Section 6.1, along with the author's recommendations for further relevant research in Section 6.2.

## 6.1 Conclusion

Robot manipulators are increasingly being used for machining tasks but are held back from realizing their full potential by their inherent inaccuracies. The stiction and backlash in internal gears which contribute to dynamic errors have, up to this point, been handled by compensating for or minimizing their effects. However, due to the redundant degree of freedom afforded by a rotating tool during machining, additional avenues can be explored. The methodology shown in this thesis outlines a strategy for minimizing or avoiding joint reversals altogether by twisting the end effector throughout a drilling motion.

A kinematic model of the robot was first developed, along with the associated forward and inverse kinematics according to the Denavit-Hartenberg convention. The drilling path was formulated assuming a known target position and plunge depth, with the twist angles at the start, target, and end being linearly interpolated to sufficiently describe the TCP position and orientation at any point along the path. Particle Swarm Optimization was implemented to optimize the twist angles at the three points with the primary objective of minimizing joint reversals, and the secondary objective of minimizing total joint travel. The possible inclusion of additional criteria such as stiffness and power consumption to these objectives is straightforward. The workspace of the robot was mapped according to the maximum achievable plunge depth still resulting in zero reversals, and the results showed trends indicating certain regions of the workspace were more viable than others, though these should not be considered universal for all manipulators. The results from simulated cases were then applied to the physical manipulator, showing that implementation is straightforward even on closed architecture industrial controllers. Across three positions in the workspace, the performance of the drilling operation when optimized for reversal

### Chapter 6 | Conclusion and Future Work

avoidance showed an improvement of at least 40% in the precision of the motion, but with a sacrifice in accuracy. This accuracy loss, however, was less significant than the improvement in precision, and altogether the results represent a net benefit in the overall drilling motion and by extension to the quality of the drilled hole. Furthermore, the reduction in number of reversals is likely to lead to reduced wear on internal components and extend product lifetime.

The novel contribution to the domain of research is the idea of pairing pose optimization methods with the goal of avoiding joint reversals during a robotic motion. The kinematic modelling methodology is well documented, and any suitable kinematic model may be substituted. However, it is shown here since having one it is an integral component of the methodology, to demonstrate knowledge of the subject, and for future reference as no available model for this specific robot was found. The definition of the redundant degree of freedom is not a new concept, nor is defining it at several points to define a tool path. PSO is also well documented in literature and was chosen as a suitable algorithm to solve the problem, but other algorithms could be implemented instead. However, the combination of these ideas with the specified goal of avoiding robotic joint reversals is unique. In addition, the formulation of the objective function to achieve the goal of minimizing joint reversals is novel, but should not be taken as the only .

The author believes that the proposed methodology will help to push the achievable machining tolerance of industrial robots tighter to within aerospace requirements, opening many avenues for the implementation of robotized drilling in composite panels on aircraft.

## 6.2 Future Work

While the enhanced performance of the optimized drilling motion is demonstrated, further research could explore the increase in performance throughout the workspace. A simple workspace map has been created for a region of this robot, however further mapping could extend the region searched and investigate how the map changes as the drilling

### Chapter 6 | Conclusion and Future Work

direction changes. Further mapping for other manipulator geometries is also possible. A gap in knowledge as to the ability to analytically predict the minimum number of reversals of a robot manipulator for a given drilling motion was exposed. Whether this is even possible due to the global, combinatorial nature of the optimization problem is unknown. Future research could explore this problem, and if no suitable deterministic solution is found, then perhaps the performance could be related to other metrics such as manipulability. This would allow a reasonable guess at the minimum possible number of reversals to be taken without the time and computationally expensive task of optimization at each point. Regarding the optimization task itself, further research could explore alternative optimization algorithms and potentially find better adapted methods or tuned settings. Beyond this, the problem could be analyzed from a perspective where there are even more DOF such as with a hyper-redundant robot or a workpiece on turntable. Care must be taken, especially as the type of redundancy changes from functional to intrinsic and PSO may no longer perform well as the number of parameters increases. Further input parameters would be necessary to fully define the redundancies in the inverse kinematics, such as elbow position or workspace locations to avoid.

Finally, while the improvement in motion and TCP deviation due to eliminating joint reversals has been studied, the quantitative impact this has on drilled holes was not studied. A further investigation with access to the proper drilling apparatus, a variety of composite panels, and high precision circularity measurement instruments is a promising avenue to explore in the future.

Nonetheless, the research herein shows that minimizing or eliminating joint reversals through the optimization of twist throughout a drilling motion is a viable strategy which leads to reduced end effector error.

# Appendix A

The following trigonometric identities were used; however, their derivation is not covered here.

If  $\cos(\theta) = b$  then

$$\theta = atan2\left(\frac{\pm\sqrt{1-b^2}}{b}\right). \tag{A.1}$$

If  $sin(\theta) = a$  then

$$\theta = atan2\left(\frac{a}{\pm\sqrt{1-a^2}}\right).$$
(A.2)

If  $sin(\theta) = a$  and  $cos(\theta) = b$  then

$$\theta = atan2\left(\frac{a}{b}\right). \tag{A.3}$$

If  $a \cos(\theta) - b \sin(\theta) = 0$  then

$$\theta = atan2\left(\frac{\pm a}{\pm b}\right). \tag{A.4}$$

If  $\sin(\theta) \sin(\phi) = a$  and  $\cos(\theta) \sin(\phi) = b$  then

$$\theta = atan2\left(\frac{\pm a}{\pm b}\right).\tag{A.5}$$

And if also  $\cos(\phi) = c$ 

$$\phi = atan2\left(\frac{\pm\sqrt{a^2+b^2}}{c}\right). \tag{A.6}$$

where the positive case for equation (A.5) corresponds to the positive case for equation (A.6), and similarly for the negative cases.

If  $a\cos(\theta) + b\sin(\theta) = c$  then

$$\theta = atan2\left(\frac{b}{a}\right) + atan2\left(\frac{\pm\sqrt{a^2 + b^2 - c^2}}{c}\right).$$
(A.7)

If  $a\cos(\theta) - b\sin(\theta) = c$  and  $a\sin(\theta) + b\cos(\theta) = d$ 

$$\theta = atan2 \left(\frac{ad - bc}{ac - bd}\right),\tag{A.8}$$

$$a^2 + b^2 = c^2 + d^2. (A.9)$$

## **Appendix B**

This appendix contains the KRL SRC file used to control the robot during drilling motion trials.

&ACCESS RVP &REL 92 &COMMENT HANDLER on external automatic DEF NewDrillMotion( ) DECL INT i; DECL AXIS jSpaceStart DECL AXIS jSpaceTarget DECL AXIS jSpaceEnd DECL AXIS home; DECL REAL TrialPicker; DECL REAL ModePicker; EXT BAS (BAS\_COMMAND : IN, REAL : IN ) BAS (#INITMOV,0) \$APO.CDIS=1; approximate positioning \$VEL.CP=0.2; \$TOOL=TOOL\_DATA[ 2 ] ;\$BASE=BASE DATA[ 0 ] FOR i = 1 TO 3\$VEL AXIS[i] = 100 ; % of max \$ACC AXIS[i] = 100; % of max ENDFOR FOR i = 4 TO 6\$VEL\_AXIS[i] = 100 ; % of max \$ACC AXIS[i] = 100; % of max ENDFOR home = {AXIS: A1 0,A2 -90,A3 90,A4 0,A5 0,A6 0}

;PTP home;

```
;Position 1/7 = 500,300,800
   ;Position 2/8 = 200,400,600
   ;Position 3/9 = 200,200,1000
   TrialPicker = 9;
   ModePicker = 2; 1 for baseline, 2 for optimized
   IF TrialPicker==7 THEN
      ; For Trial 7
      jSpaceStart = {AXIS: A1 143.1752, A2 -107.9003, A3 -79.0638, A4 11.3008, A5 88.3939, A6
217.3562
      ;|SpaceTarget = {AXIS: A1 141.688, A2 -120.7634, A3 -61.915, A4 -4.6854, A5 96.3471, A6
217.943
      ;jSpaceEnd
                   = {AXIS: A1 132.754, A2 -121.5371, A3 -57.5418, A4 -21.63, A5 110.9973, A6
223.361
      ; Rotation about tool axis: 20.1033 degrees then 24.5212 degrees
      PTP jSpaceStart
      ;PTP jSpaceTarget
      ;PTP jSpaceEnd
      IF ModePicker == 1 THEN
        LIN REL { X 50, c 0} C DIS
        LIN REL { X -50, c 0}
      ENDIF
      IF ModePicker == 2 THEN
        LIN REL { X 50, c 20.1033} C DIS
        LIN REL { X -50, c 24.5212}
      ENDIF
      ; Baseline
      ;LIN REL { X 50, c 0} C DIS
      ;LIN_REL { X -50, c 0}
      ; Position to attain: 300 300 800 with EE
   ENDIF
   IF TrialPicker==8 THEN
      ; For Trial 8
      jSpaceStart = {AXIS: A1 -78.6901, A2 -86.9817, A3 87.6934, A4 0, A5 89.2884, A6
101.3099}
      ;jSpaceTarget = {AXIS: A1 -75.7884, A2 -87.0847, A3 87.8145, A4 -0.46682, A5 87.4284,
A6 104.225}
```

```
;jSpaceEnd
                   = {AXIS: A1 -75.3314, A2 -102.9224, A3 107.3981, A4 -11.6946, A5 49.5235,
A6 109.3832}
      ; Rotation about tool axis: 1.8999 degrees then 35.6083 degrees
      PTP jSpaceStart
      ;PTP jSpaceTarget
      ;PTP jSpaceEnd
      ;LIN REL { X 20, c 1.8999} C DIS
      ;LIN_REL { X -20, c 35.6083}
      IF ModePicker == 1 THEN
        LIN REL { X 20, c 0} C DIS
        LIN REL { X -20, c 0}
      ENDIF
      IF ModePicker == 2 THEN
        LIN_REL { X 20, c 1.8999} C_DIS
        LIN REL { X -20, c 35.6083}
      ENDIF
      ; Baseline
      ;LIN REL { X 20, c 0} C DIS
      ;LIN REL { X -20, c 0}
      ; Position to attain: 200 400 600 with EE
   ENDIF
   IF TrialPicker==9 THEN
      ; For Trial 9
      jSpaceStart = {AXIS: A1 115.5591, A2 -64.3099, A3 -87.1047, A4 12.4793, A5 43.6965, A6
236.7431}
      ;jSpaceTarget = {AXIS: A1 115.223, A2 -75.2295, A3 -81.4464, A4 -2.0503, A5 70.8028, A6
245.521}
                   = {AXIS: A1 107.2031, A2 -75.6544, A3 -80.8389, A4 -2.9439, A5 75.7758, A6
      ;jSpaceEnd
253.7521
      ; Rotation about tool axis: 24.7906 degrees then 5.1428 degrees
      PTP jSpaceStart
      ;PTP jSpaceTarget
      ;PTP jSpaceEnd
      ;LIN REL { X 30, c 24.7906} C DIS
      ;LIN_REL { X -30, c 5.1428}
```

```
IF ModePicker == 1 THEN
  LIN_REL { X 30, c 0} C_DIS
  LIN_REL { X -30, c 0}
ENDIF

IF ModePicker == 2 THEN
  LIN_REL { X 30, c 24.7906} C_DIS
  LIN_REL { X -30, c 5.1428}
ENDIF

; Baseline
;LIN_REL { X 30, c 0} C_DIS
;LIN_REL { X -30, c 0}
; Position to attain: 200 200 1000 with EE
ENDIF
```

;PTP home;

END
## **Bibliography**

- [1] A. Verl, A. Valente, S. Melkote, C. Brecher, E. Ozturk, and L. T. Tunc, "Robots in machining," *CIRP Ann.*, vol. 68, no. 2, pp. 799–822, 2019, doi: 10.1016/j.cirp.2019.05.009.
- H. N. Huynh, H. Assadi, E. Rivière-Lorphèvre, O. Verlinden, and K. Ahmadi, "Modelling the dynamics of industrial robots for milling operations," *Robot. Comput. Integr. Manuf.*, vol. 61, no. September 2019, p. 101852, 2020, doi: 10.1016/j.rcim.2019.101852.
- Y. T. Oh, "Influence of the joint angular characteristics on the accuracy of industrial robots," *Ind. Rob.*, vol. 38, no. 4, pp. 406–418, 2011, doi: 10.1108/01439911111132094.
- [4] U. Schneider *et al.*, "Improving robotic machining accuracy through experimental error investigation and modular compensation," *Int. J. Adv. Manuf. Technol.*, vol. 85, no. 1–4, pp. 3–15, 2016, doi: 10.1007/s00170-014-6021-2.
- [5] L. Ding, X. Li, Q. Li, and Y. Chao, "Nonlinear Friction and Dynamical Identification for a Robot Manipulator with Improved Cuckoo Search Algorithm," *J. Robot.*, vol. 2018, 2018, doi: 10.1155/2018/8219123.
- [6] M. Cordes and W. Hintze, "Offline simulation of path deviation due to joint compliance and hysteresis for robot machining," *Int. J. Adv. Manuf. Technol.*, vol. 90, no. 1–4, pp. 1075–1083, 2017, doi: 10.1007/s00170-016-9461-z.
- [7] D. Rijlaarsdam, P. Nuij, J. Schoukens, and M. Steinbuch, "Frequency domain based nonlinear feed forward control design for friction compensation," *Mech. Syst. Signal Process.*, vol. 27, no. 1, pp. 551–562, 2012, doi: 10.1016/j.ymssp.2011.08.008.
- [8] B. Bona and M. Indri, "Friction compensation in robotics: An overview," Proc. 44th IEEE

*Conf. Decis. Control. Eur. Control Conf. CDC-ECC '05*, vol. 2005, pp. 4360–4367, 2005, doi: 10.1109/CDC.2005.1582848.

- [9] J. R. Diaz P, P. Mukherjee, and A. Verl, "Automatic Workpiece Positioning for Automatic Close-optimal Workpiece Positioning for Robotic Robotic," *Procedia CIRP*, vol. 72, pp. 277–284, 2018, doi: 10.1016/j.procir.2018.03.142.
- [10] H. Qin, Y. Li, and X. Xiong, "Workpiece pose optimization for milling with flexible-joint robots to improve quasi-static performance," *Appl. Sci.*, vol. 9, no. 6, 2019, doi: 10.3390/app9061044.
- [11] W. Zhu, W. Qu, L. Cao, D. Yang, and Y. Ke, "An off-line programming system for robotic drilling in aerospace manufacturing," pp. 2535–2545, 2013, doi: 10.1007/s00170-013-4873-5.
- [12] B. Denkena and T. Lepper, "Enabling an Industrial Robot for Metal Cutting Operations," 2015, doi: 10.1016/j.procir.2015.08.100.
- [13] A. Raoofian, A. Taghvaeipour, and A. K. E., "On the stiffness analysis of robotic manipulators and calculation of stiffness indices," *Mech. Mach. Theory*, vol. 130, pp. 382–402, 2018, doi: 10.1016/j.mechmachtheory.2018.08.025.
- [14] J. A. Pamanes-Garcia, E. Cuan, and S. Zeghloul, "Single and multi-objective optimization of path placement for redundant robotic manipulators Optimización mono-objetivo y multi-objetivo del emplazamiento de trayectorias de robots manipuladores redundantes," *Ing. Investig. y Tecnol. IX.*, vol. 3, pp. 231–257, 2008.
- [15] Y. Guo, H. Dong, and Y. Ke, "Stiffness-oriented posture optimization in robotic machining applications," *Robot. Comput. Integr. Manuf.*, vol. 35, pp. 69–76, 2015, doi: 10.1016/j.rcim.2015.02.006.
- [16] M. Newman, K. Lu, and M. Khoshdarregi, "Suppression of robot vibrations using input shaping and learning-based structural models," *J. Intell. Mater. Syst. Struct.*, pp. 1–12,

2020, doi: 10.1177/1045389X20947166.

- [17] Y. Nakamura and H. Hanafusa, "Optimal Redundancy Control of Robot Manipulators," *Int. J. Rob. Res.*, vol. 6, no. 1, pp. 32–42, 1987, doi: 10.1177/027836498700600103.
- [18] T. Zhang and P. Ferguson, "Optimal Reaction Wheel Control with Stiction and Resonance Avoidance," no. May, pp. 2–6, 2021.
- [19] L. Huo and L. Baron, "The self-adaptation of weights for joint-limits and singularity avoidances of functionally redundant robotic-task," *Robot. Comput. Integr. Manuf.*, vol. 27, no. 2, pp. 367–376, 2011, doi: 10.1016/j.rcim.2010.08.004.
- [20] E. W. Aboaf and R. P. Paul, "Living With the Singularity of Robot Wrists.," pp. 1713– 1717, 1987, doi: 10.1109/robot.1987.1087792.
- [21] J. Woolfrey, W. Lu, and D. Liu, "A Control Method for Joint Torque Minimization of Redundant Manipulators Handling Large External Forces," pp. 3–16, 2019.
- [22] M. N. A. Wahab, S. Nefti-Meziani, and A. Atyabi, "A comparative review on mobile robot path planning: Classical or meta-heuristic methods?," *Annu. Rev. Control*, vol. 50, no. October, pp. 233–252, 2020, doi: 10.1016/j.arcontrol.2020.10.001.
- [23] L. Sciavicci and B. Siciliano, Modelling and Control of Robot Manipulators. Springer, 2001.
- [24] J. Léger and J. Angeles, "Off-line programming of six-axis robots for optimum fivedimensional tasks," *Mech. Mach. Theory*, vol. 100, pp. 155–169, 2016, doi: 10.1016/j.mechmachtheory.2016.01.015.
- [25] N. Rokbani and A. M. Alimi, "Inverse kinematics using particle swarm optimization, a statistical analysis," *Procedia Eng.*, vol. 64, pp. 1602–1611, 2013, doi: 10.1016/j.proeng.2013.09.242.
- [26] N. C. N. Doan, P. Y. Tao, and W. Lin, "Optimal redundancy resolution for robotic arc

welding using modified particle swarm optimization," *IEEE/ASME Int. Conf. Adv. Intell. Mechatronics, AIM*, vol. 2016-Septe, pp. 554–559, 2016, doi: 10.1109/AIM.2016.7576826.

- [27] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Part. Swarm Optim.*, pp. 1942–1948, 1995, doi: 10.1002/9780470612163.
- [28] M. R. Bonyadi and Z. Michalewicz, "SPSO2011 Analysis of stability, local convergence, and rotation sensitivity," *GECCO 2014 - Proc. 2014 Genet. Evol. Comput. Conf.*, pp. 9–15, 2014, doi: 10.1145/2576768.2598263.
- [29] H. W. Stone, *Kinematic Modeling, Identification, and Control of Robotic Manipulators*. Springer, 2011.
- [30] P. Corke, "Robotics Toolbox." https://petercorke.com/toolboxes/robotics-toolbox/ (accessed Jul. 05, 2021).
- [31] S. Mesloub and A. Mansour, "Hybrid PSO and GA for global maximization.," *Int. J. Open Probl. Comput. Sci. Math.*, vol. 2, no. 4, pp. 597–608, 2009.
- [32] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, "Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements," *2013 IEEE Congr. Evol. Comput. CEC 2013*, pp. 2337–2344, 2013, doi: 10.1109/CEC.2013.6557848.
- [33] S. Chen and T. Zhang, "Force control approaches research for robotic machining based on particle swarm optimization and adaptive iteration algorithms," *Ind. Rob.*, vol. 45, no. 1, pp. 141–151, 2018, doi: 10.1108/IR-03-2017-0045.
- [34] K. Sörensen, "Metaheuristics-the metaphor exposed," *Int. Trans. Oper. Res.*, vol. 22, no. 1, pp. 3–18, 2015, doi: 10.1111/itor.12001.
- [35] Bansal et al., "Inertia Weight Strategies in Particle Swarm Inertia Weight Strategies in Particle Swarm," *Pap. Conf. Technol. Inf. Kharagpur, Technol.*, no. May 2014, p. 7, 2011, doi: 10.1109/NaBIC.2011.6089659.

[36] T. Krantz, F. Oswald, and R. Handschuh, "Wear of spur gears having a dithering motion and lubricated with a perfluorinated polyether grease," 2007 Proc. ASME Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf. DETC2007, vol. 7, pp. 689–695, 2008, doi: 10.1115/DETC2007-34089.