Peg-in-Hole Manipulation of the Baxter Dual-Arm Robot

Mats Ingvar Niklasson

B. Eng. Mechatronic Engineering Research Project

Department of Mechanical Engineering

Curtin University

Semester 2, 2021

60 Barker St
Belmont
WA 6104

29 October 2021

The Head
Department of Mechatronic Engineering
Curtin University
Kent Street, Bentley,
WA 6102

Dear Sir,

I submit this thesis entitled “Peg-in-Hole Manipulation of the Baxter Dual-Arm Robot”, based on MCEN4000 Mechatronic Engineering Research Project I and MXEN4004 Mechatronic Engineering Research Project II, undertaken by me as part-requirement for the degree of B.Eng in Mechatronic Engineering.

Warm regards,

Mats Niklasson
19134299

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor, Dr. Lei Cui, for his endless enthusiasm and support. Your insight, expertise and suggestions have been invaluable throughout this work. The excitement you have for robotics and engineering has impacted me more than you know and I look forward to continue building upon the foundations of knowledge you have encouraged me to cultivate.

I would also like to extend my gratitude to Amir and Hamish, the lab technicians who assisted me in setting up a new computer when the only one I could use with the Baxter robot fatally malfunctioned in the weeks leading up to the deadline of this thesis submission.

Finally, I am grateful for my family, friends and partner who supported me throughout this work and put up with my incessant thesis discussions on a daily basis.

ABSTRACT

This work developed a hybrid velocity/force control solution to the fine manipulation Peg-in-Hole task, using the compliant dual-arm Baxter robot. Skilled manipulation gives robots the ability to interact with their environment, greatly extending their utility. However, these environments are typically unknown with uncertain goal poses (position and orientation) so pure joint position control is not suitable. Further, passively compliant robots are increasing in popularity due to their increased safety around humans. Unfortunately, they tend to suffer from poor positional accuracy. To overcome this drawback, a hybrid velocity/force control method was established. It divided the problem into three discrete phases: 1) Motion in free space; 2) Search, and; 3) Insert. A PID velocity controller was used to control and constrain motion, while force/torque feedback from a wrist-mounted sensor allowed the robot to adapt to its unknown environment. Noise filtering was implemented to reduce the effect of noise from the force/torque sensor. A state machine was designed to implement the control methodology and adapt the end effector movement for each stage of the task. The results demonstrated that the solution was able to insert the peg into a hole with a radial clearance of 1.5 mm in approximately 37 seconds, comparable to previous works.

Table of Contents

LIST OF FIGURES

LIST OF SYMBOLS

q	Joint angle (rad)
$\boldsymbol{q}$	Vector of joint angles
$\dot{q}$	Joint velocity (rad/s)
$\dot{\boldsymbol{q}}$	Vector of joint velocities
$\boldsymbol{X}$	End effector pose (position and quaternion orientation)
$\boldsymbol{x}_p$	Vector of end effector position in global cartesian coordinate frame
$\boldsymbol{x}_o$	Vector of end effector quaternion orientation in global frame
$\dot{\boldsymbol{X}}$	End effector twist (linear and angular velocities, respectively)
v_x, v_y, v_z	End effector linear velocities (m/s) in the x-, y- and z-axis respectively
$\boldsymbol{v}$	Vector of end effector linear velocities in global cartesian frame
$\omega_x, \omega_y, \omega_z$	End effector angular velocities (rad/s) in the x-, y- and z-axis respectively
$\boldsymbol{\omega}$	Vector of end effector angular velocities in global cartesian frame
$\boldsymbol{J}$	Jacobian matrix
$\boldsymbol{J}^{-1}$	Inverse Jacobian matrix
$\boldsymbol{J}^{+}$	Moore-Penrose pseudoinverse Jacobian matrix
K_p, K_i, K_d	Proportional, integral and derivative coefficients respectively
$\boldsymbol{\tau}$	End effector twist vector, made up of cartesian force and torque
f_x, f_y, f_z	Forces on the end effector in x-, y- and z-axis respectively
τ_x, τ_y, τ_z	Torques on the end effector about the x-, y- and z-axis respectively

1.0 INTRODUCTION

1.1 Motivation

Manipulation has been a key aspect of robotics since their inception. Skilled manipulation allows robots to interact within their environments and become useful tools for endless applications. However, despite advancements in the underlying technology powering robots, the same advancements in skilled manipulation are yet to be proven commercially or on a large scale. This results in humans still being required to undertake dangerous and dirty jobs that could otherwise be completed by robots.

Skilled manipulation is a broad term relating to the fine handling of a tool to accomplish a goal with high precision, repeatability and little margin for error. If the environment can be controlled it typically will be, as it is easier to operate in and proven by current commercial standards of robotic manipulation which use position control for tasks such as welding. While position control can work well in a known environment, it may not translate well to unknown environments with higher uncertainties. Additionally, position control in unknown environments typically requires extra sensing devices such as cameras, infrared sensors or lidars that add to the complexity of code and higher costs.

This work develops a hybrid force and velocity control model for the manipulation of the Baxter research robot, specifically for the Peg-in-Hole (PiH) task. The model does not require known goal coordinates and instead uses feedback from a force-torque sensor (FTS) to control the joint velocities and achieve fine manipulation. A PID velocity controller is developed to control motion as it responds to the force feedback from the FTS. The task is divided into three main phases and a state machine is used to integrate the control throughout the various states.

1.2 Objectives

The focus of this work is to develop a velocity control model using feedback from a force-torque sensor for skilled manipulation of Baxter's end effector. The model is applied to the manipulation task of round Peg-in-Hole and is validated on the physical dual-arm Baxter Robot.

The objectives of this work are to:

a) implement velocity control of Baxter's joints.

b) develop a hybrid model of velocity and force control for a round PiH task, using feedback from a force-torque sensor instead of known coordinates to achieve peg-in-hole.

1.3 Thesis Outline

This thesis is organised into 7 chapters, with the following structure:

Chapter 1 provides an introduction to this work. The motivation, background and objectives are stated and the content of the work is outlined.

Chapter 2 provides a background and a literature review of related research on fine manipulation and highlights the limitations of some current approaches. Firstly, it explores impedance control as a current standard for manipulation. The findings discuss why impedance control will not be suitable for a robot like Baxter with series elastic actuator (SEA) joints and for the PiH task. Secondly, it explains how hybrid control is used to mitigate position error inherent to joint position control and overcome the drawbacks of impedance control. Further, the benefits of hybrid control are explored, specifically in an unknown environment where goal coordinates are not defined. Thirdly, this chapter examines how hybrid joint control can be applied to the

PiH task. The chapter concludes with a discussion of non-technical aspects related to this work, primarily the social and ethical considerations related to robotic manipulation and human cooperation.

Chapter 3 develops and tests position control to examine its suitability for completing the PiH task on the Baxter robot. It also introduces the hardware used along with the experimental setup. Inverse position kinematics is used to calculate joint positions from the end effector goal pose in the cartesian space. This chapter concludes with a summary of the position control results, confirming that it will not be suitable for fine manipulation of the Baxter robot.

Chapter 4 explores joint velocity control of Baxter's limbs, implementing inverse velocity kinematics. Joint velocities are computed using the Moore-Penrose pseudo-inverse Jacobian to adjust the twist of the end effector to follow set paths. Further, a PID controller is developed to provide improved velocity control. Results in this chapter validate the notion that velocity control will be suitable for the final PiH task on the compliant Baxter robot.

Chapter 5 introduces the hybrid control method, adopting the velocity control framework from Chapter 4 in combination with force and torque feedback from a sensor attached to Baxter's right wrist. This chapter explores how the force feedback can be used to achieve movements without goal coordinates, such as lowering the end effector until a normal force threshold is exceeded. Additionally, a method of filtering noise from the FTS is developed to improve the overall reliability of the control method. An approach is developed to integrate the hybrid velocity/force control method with noise filtering. Results demonstrate its success and suitability for achieving the objectives of this work.

Chapter 6 combines the hybrid control methodology from Chapter 5 and applies it to the round PiH task. Three distinct phases are defined: 1) Motion in free space; 2) Search, and; 3) Insert. Control strategies for each phase are explained in detail. A grid

search algorithm is developed to find the hole within a small area, based off the normal force of the flat surface against the end effector. A peg insertion algorithm is developed to finely insert the peg and confirm that it is inside the target hole. These algorithms are combined in a state machine and validated on the Baxter robot. The final results are presented and discussed.

Chapter 7 presents the overall conclusions of this thesis. The main achievements and challenges of this work are highlighted and the chapter concludes with proposals for future work that arose as a result of this work.

2.0 BACKGROUND

2.1 Introduction

Robotic manipulation has evolved from operating in controlled, structured environments to uncontrolled environments with higher complexities, typically requiring motion planning and skilled behaviours for fine manipulation. Manipulation is widely used in the automotive and assembly industries, but these robots can only operate due to the known goal coordinates and the highly controlled environments which are unsafe for humans to work in without barriers. Significant research works exist which explore how robots can be programmed to complete manipulation and assembly tasks in less controlled environments where goal coordinates may not be defined. The Peg-in-Hole assembly task is a well-covered research topic which exists as a proof of concept for fine manipulation, and many methods have been developed to achieve this. This chapter summarises three main concepts that are widely used in the literature to achieve the PiH task.

Firstly, this chapter investigates impedance control and how it has been used to overcome unknown goal coordinates for robotic manipulation tasks in relevant research work. Additionally, it outlines why this work will not be implementing impedance control. Second, it summarises the various hybrid control models used in previous literature and how hybrid control will be useful for the objectives of this thesis. Third, this chapter investigates previous works that specifically address the PiH task on various single and dual-arm robots, and how both active and passive compliance can be used to achieve peg insertion on a positionally inaccurate robot, like the Baxter robot used in this work. This chapter concludes with a discussion of non-technical aspects of the work of this thesis, mainly the social and ethical considerations of robot and human collaboration.

2.2 Impedance Control

Impedance control develops a relationship between a robotic manipulator and the environment, with the aim of controlling the motion of the robot and the contact forces against the environment or surface [1]. Some of the earliest works on impedance control were developed by Hogan [2-4]. In [2] he describes impedance when seen from the environment as accepting motion inputs and yielding effort outputs *e.g.* force. Hogan uses this definition to describe a general manipulator control strategy as one to control its motion while also giving it a disturbance response for deviations from that desired motion. This disturbance is an impedance and changing it regulates the dynamic interaction between environment and manipulator. He adds that in most common cases when the environment is an admittance, the relation should be a function of impedance. This function produces a specified force in response to the imposed motion by the environment. In [3] Hogan specifies that the critical difference between impedance control and more conventional approaches to manipulation is that the controller creates a dynamic relation between position and force, rather than solely controlling these independently.

Hogan expands on impedance control in [4], when he outlines that an important physical constraint of a robotic manipulator is that it may impress a force on the environment or impose a displacement or velocity, but cannot do both simultaneously. Building on this, he explains that a manipulator must either be an impedance, displaying spring-like behaviour, or an admittance, displaying mass-like behaviour. Additionally, if the manipulator is an impedance then the environment must be an admittance and *vice versa*. Hogan summarises impedance control as a unity of robot transpose tasks with the control of interactive tasks like using tools, not dissimilar to the PiH task this work explores.

One of the challenges that impedance control overcomes is how a robot behaves if it collides with the environment unexpectedly. In pure joint position control a robot will aim to reach the goal pose regardless of whether an obstacle is present or not. This can lead to high forces and damage to the manipulator. Impedance control can be used to

regulate the mechanical impedance of a robot's end effector, rather than solely tracking a force [5]. In this manner, impedance control can be used to create active compliance with the environment which is particularly useful for pose uncertainties like those in the PiH task. The compliant behaviour allows the robot to move slightly to allow the peg to fall into the hole instead of becoming stuck a millimetre from the hole and generating high forces trying to reach the position exactly as it was commanded.

Impedance control is used specifically for the PiH task by Zou *et al*. in [6]. They successfully combine impedance control with a joint velocity control loop to achieve compliant behaviour of the manipulator and peg insertion. However, it is worth noting that this was achieved on the UR5 6-DoF robot arm, which is much more accurate than the Baxter robot used in this work. The UR5 robot has a repeatability of ± 0.1 mm [7] compared to Baxter's ± 3 mm, found in [8]. For a task like PiH where the clearance between peg and hole is generally less than 1 mm, this difference is crucial. The positional inaccuracy of the Baxter robot is one of main reasons impedance control will not be suitable for the work of this thesis. Yamada *et al*. [9] also found that impedance control was not well-suited to cooperative control of an arm and fingers because any error in the position was then used in calculating virtual dynamics. This error then propagated throughout the entire control loop. It is likely that the poor positioning accuracy of the Baxter robot would lead to similar error propagation in this work if impedance control was implemented.

Calanca *et al*. also confirm that one of the main drawbacks to impedance control is the need for accurate and precise positioning [10]. This is a significant issue for the Baxter robot used in this work. Baxter has poor positioning accuracy because its joints use series elastic actuators (SEAs) which provide mechanical compliance and make the joints 'soft'. Consequently, it loses repeatability and accuracy [8]. However, a benefit of the mechanical compliance is that impedance control is not necessarily required to achieve compliant behaviour since it already exists passively for the Baxter robot. This is discussed further in the next section. Another drawback for impedance control found in the literature is that if a force is being applied against a surface and the surface is suddenly removed the robot could theoretically attempt a dangerous motion to exert

the pre-set force [11]. This can result in harm to nearby humans, the robot itself and the environment it operates in.

As a final note on impedance control, the literature can often interchange the definitions of impedance and admittance control [4]. For the purpose of this work the difference has already been explicitly made previously. Admittance control is the inverse of impedance control and excels in low inertia environments so is generally not used for fine manipulation against a rigid surface, like PiH task [12].

The literature shows that impedance control can be a useful control method for fine manipulation and the PiH assembly task when used on stiff and accurate robots. It allows the manipulator to impose a defined force on its environment regardless of disturbances. However, it has been shown that impedance control is not suitable on a positionally inaccurate robot like Baxter, which has passive mechanical compliance through the SEAs present at each joint. While impedance control will not be used for this work, the notion it introduces of force feedback provides a solid foundation to build upon.

2.3 Hybrid Velocity/Force Control

Most applications of industrial robots are typically limited to tasks where the robot and its environment do not dynamically interact [13]. These tasks require accurate position control and there is almost no exchange of energy between the robot and the environment. Hybrid robotic manipulation control shares similarities with impedance control in that it uses force feedback to determine the motion of the end effector. However, hybrid control decouples force and position (or velocity) instead of establishing a single relationship between them, allowing for independent control and monitoring of the variables separately [14].

One of the earliest works on hybrid control was developed by Raibert and Craig in [15]. Here they develop hybrid position/force control of a manipulator, using a wrist-mounted force sensor to control the trajectory in a Cartesian coordinate system. They achieve positive results, however, using position control requires a positionally accurate robot, unlike the Baxter robot used for this thesis. To overcome the drawback of requiring an accurate robot, hybrid velocity/force control (also called motion/force control) was developed [16]. This approach, like position/force control, divides the problem into subspaces: velocity joint control and force feedback from the force/torque sensor (FTS). It guarantees force regulation while trading off positional accuracy. This overcomes one of the drawbacks of impedance control in that if the contact force is suddenly removed the robot does not risk attempting a dangerous movement to exert a force.

The difference between position and velocity joint control is critical to this work and is worth discussing in further detail. Position control is not optimal for fine manipulation assembly tasks or human cooperation. During such tasks the final goal pose (position and orientation) is unknown and typically has a high level of uncertainty [17]. Further, the slight position increments during these tasks generate small position errors which propagate and make position control PID tuning difficult. On a robot like Baxter these errors are significant and can amplify throughout the control loop. Consequently, by using velocity joint control instead of position control, these positional errors can be ignored as they do not contribute to the closed-loop control of the manipulator. In [18], Hou and Mason use hybrid velocity/force control to dictate the end effector velocity in some directions while controlling the force in other directions to maintain contact with the environment. They showed robust results and displayed how this control method can be used for fine manipulation regardless of position error.

The relevant literature on robotic manipulation highlight how hybrid control can be used to achieve fine end effector control. In particular, hybrid velocity/force control has been proven successful on positionally inaccurate robots with the presence of position error. In this way it overcomes the position error that makes impedance

control difficult on a robot like Baxter. Further, hybrid velocity/force control is well-suited for this thesis because monitoring force, in contrast to controlling it, is sufficient on a mechanically compliant robot like Baxter.

2.4 Peg-in-Hole

The Peg-in-Hole task is the focus of this work. It is an example of skilled manipulation that comes naturally to humans, but remains challenging for robots. The PiH task is particularly challenging for mechanically compliant robots like the dual-arm Baxter robot because of the inherent position error caused by SEA joints [8]. PiH is not defined by the shape of the object being inserted but instead refers to any generic pair of items. So long as one item is being manipulated into the other object it can be considered a PiH task, such as inserting a key into a lock or driving a screw into a hole. Further, the difficulty of the task can be varied by adjusting the clearance and geometry of the peg and hole pair. The versatility, real world application and challenges of the PiH task make it an excellent research topic and benchmark of robotic manipulation.

This problem is challenging for various reasons and in this section three main issues are discussed with context to previous works. Firstly, the PiH task can be separated into phases (*e.g.* search and insert) which require the robot to operate differently for each phase. The different manipulator behaviours required for each phase must be considered when developing a solution. Secondly, the goal pose of the hole is often unknown and typically contains uncertainties which must be overcome. Lastly, this section concludes by investigating the PiH problem specifically on dual-arm robots and the associated complexities that are introduced by such robots.

2.4.1 Phases of the Peg-in-Hole Task

The PiH task inherently has different phases which must be considered in trying to achieve peg insertion. These can generally by summarised as search and insert. Both phases require the robot manipulator to behave differently and the control of these

phases can be addressed in various ways. Zhang *et al.* propose a two-phase scheme based on an "outside adjustment" and "inside adjustment" [19]. The outside adjustment involves the approaching motion of the peg towards the hole, and then adjusting the pose when contact is made. The inside adjustment is the phase when the peg has been partially inserted, so the pose is adjusted until complete insertion is achieved. In their work they achieve peg alignment at each phase by calculating the centre of rotation based on single and double contact state models between the peg and hole.

In a similar manner, Li proposes different impedance controllers for the PiH task based on the discrete states of the problem: pick up/put down, approaching, and insertion [20]. Li then breaks insertion into subphases based on measured force/torque values from the FTS. Jasim *et al*. identify three phases: free space, sliding contact and peg directly on the hole [21]. Their work primarily addresses the second phase known as the contact-state (CS), because this phase is crucial in knowing if the robot has missed the hole.

The literature on PiH widely supports the notion that the problem can be divided into separate phases and will provide a foundation for the approach developed in this thesis.

2.4.2 Goal Pose Uncertainties

Overcoming the challenges of inserting a peg into an unknown and uncertain goal pose is significant and the focus of many research works. Several methods have been developed to address the problem. One approach is to use vision feedback in combination with visual-servoing [22-24]. This can work well on positionally accurate robots, but regardless of the robot this method can suffer from high contact forces when no force feedback is utilised. Additionally, the cameras are unable to see inside the hole so without a FTS it is hard to gain a full picture of the system and its interactions. Generally visual-servoing works best for tasks like pick-and-place which have more goal certainty that a camera can recognise.

The most widespread method to solving the PiH task implements impedance control which is a position/force controller discussed previously in this thesis. This approach works well by artificially reducing the stiffness to mitigate the undesirable effects of uncertainties during peg insertion. Impedance control introduces compliance to the system and is worth discussing further. Compliance can be achieved passively or actively. On a robot like the dual-arm Baxter robot which utilises SEAs at its joints, passive compliance is present because the robot will attempt to compensate for uncertainties regardless of the controller. Passive compliance can also be introduced through specially designed mechanical devices like a remote centre of compliance (RCC) introduced in [14]. Conversely, active compliance can be introduced through control strategies, such as impedance control, which work particularly well on traditional, stiff robots.

Zou *et al.* used a stiff robot with impedance control to achieve active compliance for solving the PiH task in [6]. Dong *et al.* built upon impedance control for the PiH task to also consider the elastic contact and friction between peg and hole surfaces in [25]. Park *et al.* developed an interesting solution to the PiH task and achieved compliant behaviour without the use of a FTS [26]. They designed a control scheme that estimated the assembly force depending on the contact state between peg and hole. Depending on the estimated force, the manipulator would attempt four different unit motions: pushing, rubbing, wiggling and screwing.

Further, Wang *et al.* approached the PiH task by developing a dual compliance strategy, utilising both passive and active compliance [27]. They built a mechanical displacement sensing device which was attached to the end effector of the robot. This device provided displacement feedback of a spring. The measured displacement was used in a position control loop for active compliance, in combination with the passive compliance provided by the spring. Tsumugiwa *et al.* address the PiH problem by developing a "switching control" method [28]. This method switched between impedance control for carrying an object in cooperation with a human operator, and torque control for fitting the peg into the hole.

A review of the literature highlights that there are several ways to address the goal pose uncertainties associated with the PiH task. While visual-servoing can be used, it leads to a risk of high contact forces and is better suited for manipulation tasks like pick-and-place. For the PiH task it is most common to address pose uncertainties through impedance control for active compliance or using mechanical devices for passive compliance. Active compliance requires accurate position control so for this work impedance control alone will not be suitable. The Baxter robot used in this work is built with SEA joints and is inherently compliant and positionally inaccurate. Therefore, to overcome the goal pose uncertainties of the PiH task this work will aim to develop a hybrid velocity/force control architecture in combination with the passive compliance of the Baxter robot.

2.4.3 Dual-Arm Robotic Manipulation

A subcategory of fine manipulation that is increasingly attracting the attention of researchers is dual-arm robots. Dual-arm robots offer significantly increased dexterity, manipulability and flexibility with respect to assembly tasks like PiH [29]. A discussion of previous works and the existing literature on dual-arm robotic manipulation will be presented in this section, given this work is undertaken on the dual-arm Baxter robot.

Dual-arm robots commonly suffer from poor positioning accuracy. Consequently they are usually constrained to tasks that do not require high repeatability or accuracy. These tasks include wiping tasks for daily chores, teaching a robot to draw shapes, and object sorting through pick-and-place [30-32]. Several approaches have been developed to overcome the poor positioning shared by many dual-arm robots. In [33] Shin and Kim propose human-like dual-arm manipulation using motion data captured from a sensor. They use this data to develop human-like trajectories with the rationale that robots operate in human environments so it seems logical that they operate in human-like ways. This natural movement is more understandable and predictable by humans which can increase safety and make it easier to recognise abnormal behaviour. For achieving fine manipulation they use virtual dynamic models (VDM), analogous

to impedance control. The human-like trajectories are fed into the VDM controller which controls the motion and force of the end effectors imposed on the manipulated object. Similarly, in [34] Krüger *et al.* use a dual-arm robot and impedance control for cooperative assembly and successfully complete the PiH task.

Another approach to fine manipulation tasks on dual-arm robots is proposed by Huang *et al.* in [35]. They develop a master-slave coordination between the two arms whereby the moving and fixed arm are identified as the master and slave arm respectively. The arms move alternatively and the scheme is modelled off human behaviours during the assembly process. Similar to this thesis, they also use the Baxter robot. Interestingly, they used position control for the non-contact phase and torque control for the contact phase. Position control is suitable for the non-contact phase because the peg is in a predefined pose that will eventually come into contact with the hole surface, despite the position error. Once contact is detected the search is performed in torque control mode, with a spring-damper model applied to all joints. Like impedance control, this uses measured joint positions so is subject to the relatively high positional inaccuracies of the Baxter robot [8].

Overall, the literature on dual-arm robot manipulation approaches the PiH task and other assembly tasks by implementing the well-covered impedance control strategy. This thesis will use hybrid velocity/force control to gain the benefits of tracking force while also avoiding the positional inaccuracies of the Baxter robot.

2.5 Social and Ethical Implications of Collaborative Robots

Human-robot interaction (HRI) is a growing area of interest as robots become more ubiquitous and smarter. This is especially true for humanoid and collaborative robots, or “cobots”, like the Baxter robot. This section of work will provide a brief non-technical discussion on the social and ethical implications of cobots in a workplace context. It is relevant to the objectives of this thesis because fine manipulation on a mechanically compliant robot is safer than traditional robots. The most widely-used

industrial robots are stiff and programmed to move from one position to the next, with no consideration of obstacles that may be in the way. Consequently, they are unsafe for humans to work with and are placed behind barricades.

Cobots are increasingly being adopted into the workplace, particularly by small and medium size companies. This introduces daily challenges for workers, such as technical challenges, learning new skills and the restructuring of working procedures and routines. In [36], Wallace highlights that historically robots have been segregated from human workers for safety reasons. This disconnection between human and robot has led to the commonly held view that robots are hazardous. He also presents four social challenges associated with introducing cobots into the workplace. These are summarised as:

1. Implementing cobots involves complex learning processes;
2. Values will typically emerge between groups and individuals, not the company;
3. Human skills are not fixed or explicit but instead will change as a result of how cobots are used;
4. The changes to human work are hard to predict before cobots are introduced and instead arise during the process [36].

Wallace's findings highlight the need for considering the social implications of introducing cobots into the workplace. If the human workers become too disrupted or disengaged it can lead to poor outcomes for the individuals and the companies.

It is also important to consider the ethical implications of introducing cobots into the workplace. Wallace identifies several ethical challenges. One challenge that arises is the need to establish ethical responsibility and codes of practice regarding how cobots are implemented. He also highlights that the design of robot platforms introduce ethical and moral consequences related to the tasks cobots are programmed to

complete, and any human injury that may arise. These challenges are similarly discussed by Saxena *et al.* in [37]. In this paper they discuss existing industry standards and call for legislation to prioritise the physical and mental health of human workers as cobots are introduced into the workplace.

While the objectives of thesis are not directly related to HRI, it can lead to developing solutions which make cobots safe to operate around humans. Consequently, it is important to consider the social and ethical effects this may have on a human workforce.

2.6 Summary

The literature shows that there are several ways to approach the PiH task. One widely used approach is to introduce compliant motion techniques to account for the stiffness of most industrial robots. Namely, these are impedance control and passive compliance through mechanical devices installed on the end effector of a robot. One downfall of impedance control is that it requires accurate position control which the Baxter robot used in this work lacks. Additionally, the Baxter robot is passively compliant so inherently possesses some of the benefits offered by impedance control. Another popular approach to the PiH task is hybrid position/force control. This has been shown to work well but still suffers from the need for accurate positioning. Therefore, the hybrid velocity/force control method is explored. Previous works highlight that this approach works particularly well for manipulation tasks like PiH where the goal pose is unknown and contains uncertainty. Further, it is important to consider the control strategies for the different phases of the PiH task and how these will interface on a dual-arm robot like Baxter. The following chapters will develop a hybrid velocity/force control scheme to achieve the PiH task on the Baxter robot.

3.0 POSITION CONTROL

3.1 Introduction

This chapter investigates joint position control on the Baxter robot. The literature widely supports impedance and hybrid position/force control as methods to completing the PiH task. However, these approaches are usually carried out on stiff robots with higher repeatability and accuracy than the Baxter robot used in this work. Firstly, this chapter introduces the hardware and software setup used throughout this work. Secondly, a background theory of inverse pose kinematics is introduced and the standard notions are defined. An approach is then developed to test position control. Lastly, it validates that position control will not be suitable for the PiH task, and presents results of using position control to follow a trajectory.

3.2 Baxter Robot Setup

This section outlines the experimental conditions under which this work was conducted. The same conditions were used for the remaining sections of this thesis and will only be outlined here. The robot used for this work was the Rethink Robotics Baxter robot. It is a dual-arm robot with 7 degrees of freedom (DoF) for each arm, shown below in Figure 1.

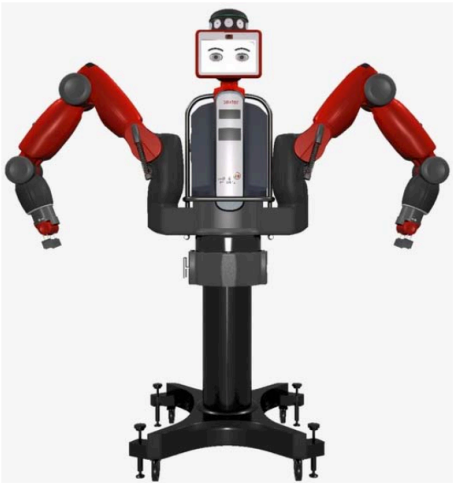

Figure 1. Baxter Robot [38]

Notably, the end-effectors depicted above were not used. Instead, a Barrett Hand BH8-282 and ATI Mini40 force/torque sensor (FTS) were mounted on the right wrist, shown in Figure 2 and Figure 3 below. These were used to grasp the peg, and monitor wrench (force and torque) respectively. Figure 4 is a photo of the final assembly of the right wrist, which will be used as the primary arm for moving and searching.

Figure 2. Barrett hand BH8-282

Figure 3. ATI Mini40 force torque sensor (FTS)

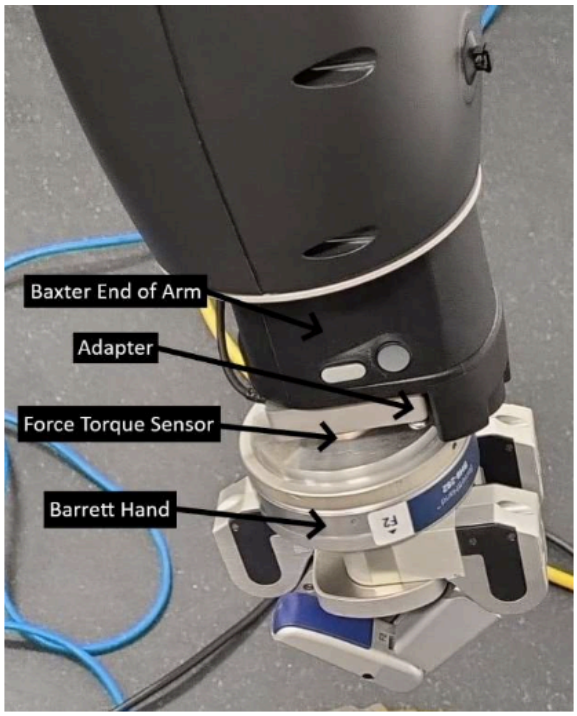

Figure 4. Barrett hand and FTS mounted on Baxter [39]

The Baxter robot, Barrett hand and FTS were all programmed in the Robot Operating System (ROS) framework. ROS provides a communication framework between independent tasks. These tasks, or processes, are called nodes and can communicate with each other through subscriber-publisher and client-server relationships. ROS is also powerful as it provides a collection of open-source software packages, like the Baxter robot software developer kit (SDK) and other packages required for the operation of the Barrett hand and FTS. The Baxter SDK is written in Python, and all code for this thesis was also written in Python. All units follow ROS SI conventions: metres, kilograms and seconds. All angles are described in radians.

3.3 Inverse Pose Kinematics

Position control was tested to validate whether it would be suitable or not for the final PiH task of this work. This relied on implementing inverse pose kinematics (IPK). Robots controlled by computers are actuated in the *joint space*, where each joint is commanded to a certain angle. However, objects to be manipulated by an end effector are expressed in a global cartesian coordinate frame, named the *task space*. Therefore, to solve the joint angles for a specified end effector frame, an IPK solution must be solved. Inverse kinematics is the calculation of joint angles, q, given the end effector pose (position and orientation) [40]. This is in contrast to forward kinematics, which solves the end effector pose given the joint angles. For this work, it translated to a vector $\boldsymbol{q}$ where each value is the angle of a joint, shown below in equation 3.1. *S*, *E* and *W* denote shoulder, elbow and wrist joints respectively. The kinematic chain begins from the first shoulder joint S_0 and ends at the third wrist joint W_2 in the order shown below.

$$\boldsymbol{q} = [\, S_0 \quad S_1 \quad E_0 \quad E_1 \quad W_0 \quad W_1 \quad W_2 \,]^T \tag{3.1}$$

The end effector task space is defined by two matrices, position and quaternion orientation. Respectively, these are shown below in equations 3.2 and 3.3:

$$\boldsymbol{x}_p = [\, x \quad y \quad z \,]^T \tag{3.2}$$

$$\boldsymbol{x}_o = [\, x \quad y \quad z \quad w \,]^T \tag{3.3}$$

The overall task space is a combination of these vectors and is shown below in equation 3.4. Further, the task space can be represented as a function of the joints, shown in equation 3.5, where $\boldsymbol{X}$ is the *end effector configuration vector* [40].

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_p \\ \boldsymbol{x}_o \end{bmatrix} \tag{3.4}$$

$$\boldsymbol{X} = f(\boldsymbol{q}) \tag{3.5}$$

There are several techniques to solve the IPK problem, such as decoupling, inverse transformations and iteratively [40]. Fortunately, the Baxter SDK from Rethink Robotics has its own IPK solver that is ready to use. An example of its usage is provided by Rethink Robotics [41]. It is simply used by specifying the goal pose into the code, as a "Pose" ROS message. It then solves the joint angles and moves the robot to the new configuration.

Another method of generating the joint angles for a given end effector pose is through the ROS motion planning library, MoveIt. This allows a user to interact the with a virtual Baxter robot through a GUI by moving the end effector. MoveIt will then plan and can execute the path on the physical Baxter robot, if a valid path is found.

3.4 Approach

This section describes the approach used to test position control on the Baxter robot. Firstly, the right end effector was manually moved to a starting position in front of Baxter. This was done using a convenient feature, called "Zero-G" mode. By pressing the buttons on the cuff of the wrist, a human is easily able to move the respective arm

without resistance from the motors of each joint. The Baxter interface package was then used to obtain the coordinates of this pose, which is provided in the Baxter SDK [42]. This package defines a "Limb" class with methods for obtaining the current end effector coordinates, moving the end effector and other methods. Alternatively, the end effector pose can be found through the command line interface by echoing the *endpoint_state* topic. This provides the current pose, twist and wrench at the end effector. The starting pose used in this case is shown below.

$$\boldsymbol{X} = [\ 0.550 \quad -0.131 \quad 0.191 \quad -0.014 \quad 0.708 \quad 0.703 \quad 0.051\ ]^T \qquad (3.6)$$

Next, a program was written in Python to move the end effector 10 cm in the positive y-axis, incrementing by 1 cm with each movement. This was achieved using the IK solver provided by Rethink Robotics and commanding the joint angles found for each incremental movement. All DoF were constrained, except the y-axis. The results are presented and discussed in the next section.

3.5 Results and Discussion

The results from position control of the Baxter robot confirmed that it would be unsuitable to use for the final solution to the PiH task. Figure 5, Figure 6 and Figure 7 below show the position of the end effector in the x-, y- and z-axis respectively while the aforementioned program was executed. The results were recorded using the *rosbag record* feature that ROS provides for recording topics to a file which can be played back.

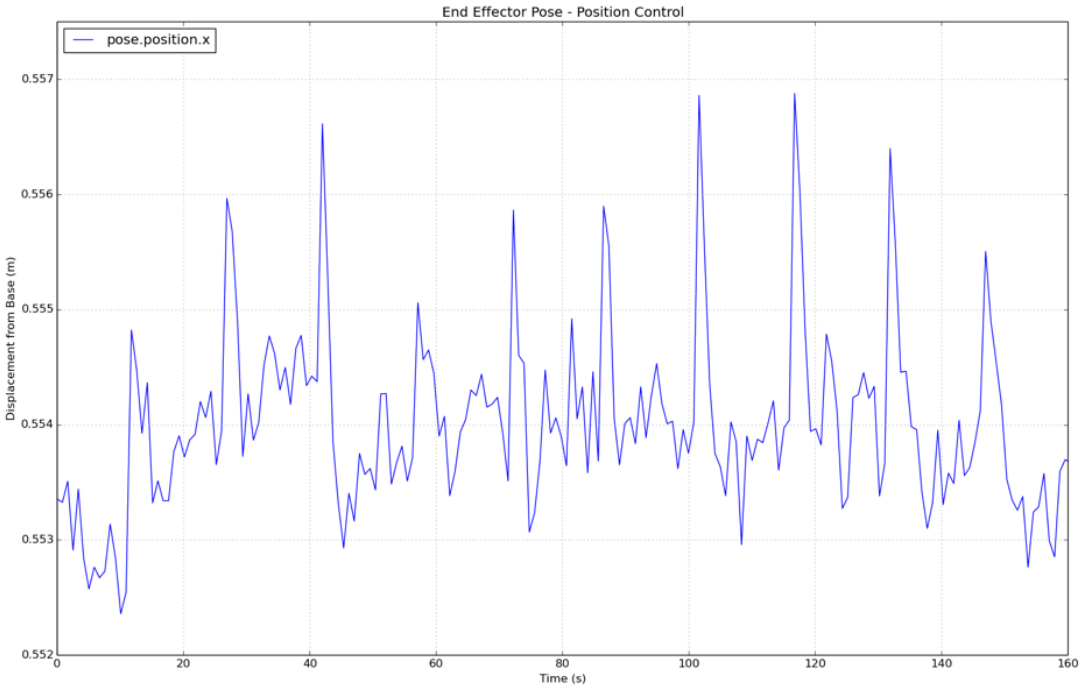

Figure 5. End effector pose in x-axis using position control

Figure 5 shows the movement of the end effector in the x-axis. The code was written to maintain the same x-position throughout the entire movement. However, the results presented indicate that this did not occur. Throughout the entire path the x-position fluctuated between approximately 0.5525 m and 0.557 m. This resulted in an overall fluctuation of ~ 4.5 mm and a maximum difference of ~ 7 mm from the setpoint of 0.550 m described in equation 3.6. The largest spikes in movement in the x-direction occur each time the end effector moves. For a task like PiH which has tolerances generally less than 1 – 2 mm, such large positional errors will make the task almost impossible.

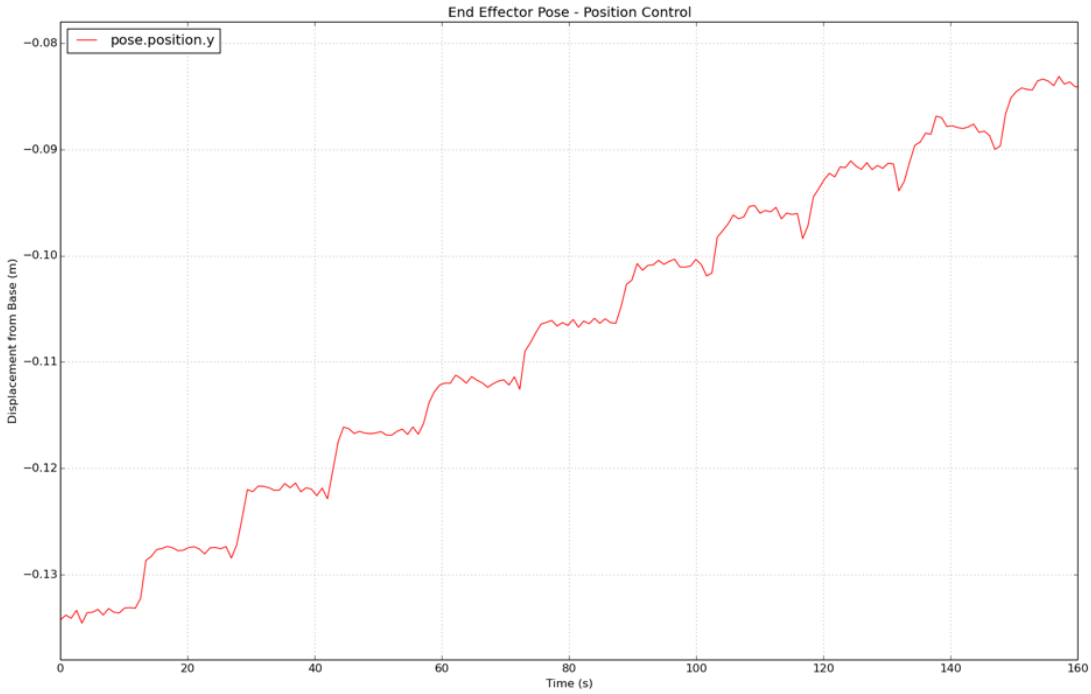

Figure 6. End effector pose in y-axis using position control

Figure 6 above displays the task space in the y-axis. This figure clearly shows the incremental movements in the only DoF that was not constrained for this test. As mentioned, the code was written to increment the end effector ten times by 1 cm in the positive y-direction. The starting y-position is shown in equation 3.6 as -0.131 m and immediately at $t = 0\ s$ (starting time) it can be seen the end effector is closer to -0.135 m. At this starting position, after ten increments of +1 cm, it should end at -0.035 m. However, it instead ended at approximately -0.085 m. Therefore, instead of moving 10 cm, it only moved 5 cm. The results show that each 1 cm increment that was commanded by the code appears to instead result in approximately 0.5 cm increments. Such large positional errors relative the clearances required for the PiH manipulation will render position control almost impossible to use for this work.

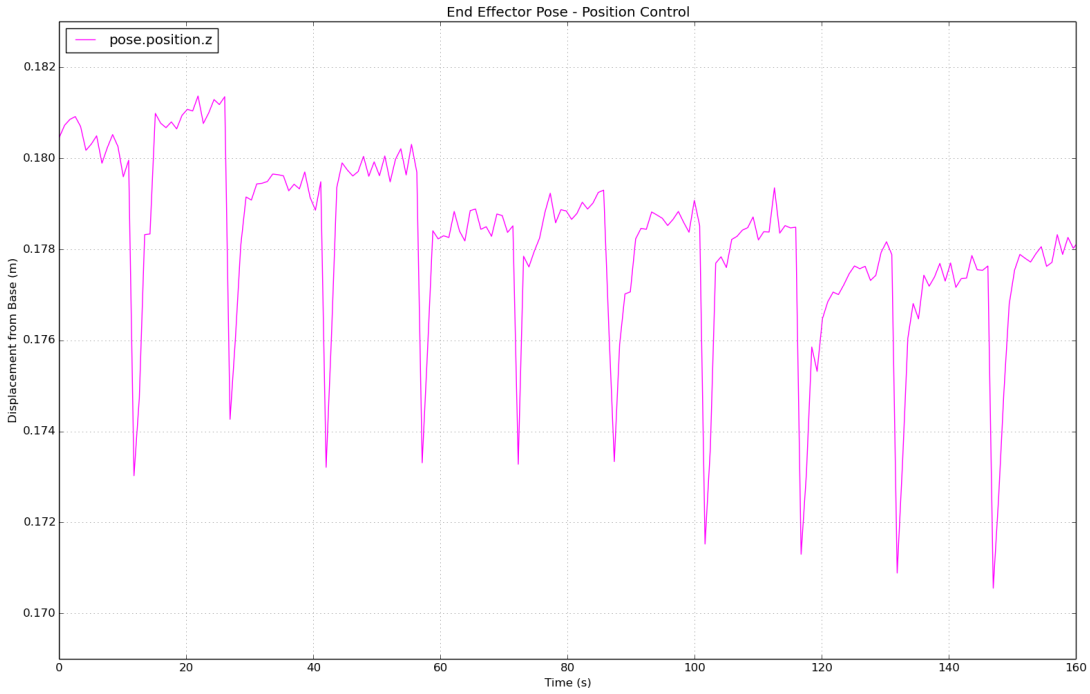

Figure 7. End effector pose in z-axis using position control

Figure 7 above shows the end effector position in the z-axis. Like the x-axis, movement in this direction was also constrained in the code. However, the inaccuracies of the Baxter robot resulted in positional fluctuations from approximately 0.181 m to 0.171 m. The largest spikes occur with each y-direction movement and the results indicate that at each step the end effector dropped towards the ground. The displacement of each drop is approximately 7 mm and is unusable for the PiH task. The above figure also shows that the start and end z-position changes from approximately 0.181 m to 0.178 m, for a drift of 3 mm throughout the entire movement.

3.6 Conclusion

The results presented in this chapter confirm that joint position control of the Baxter robot will not be suitable for the final objectives of this work. The path commanded to test position control was to move the end effector in a straight line, 10 cm in the positive y-direction while constraining all other degrees of freedom. The results showed that it only moved 5 cm in total, with large fluctuations throughout the path in

the x- and z-axis. The overall positional errors were significant when compared to typical PiH clearances of 1 – 2 mm. Such large errors render position control unsuitable for this work, and the rest of this thesis will focus on implementing velocity joint control.

4.0 VELOCITY CONTROL

4.1 Introduction

This chapter investigates joint velocity control of the Baxter robot and forms the foundation of the final hybrid velocity/force control developed for the PiH task. First, a background of inverse velocity kinematics is introduced along with standard notations. Secondly, a PID controller is developed to provide improved velocity control of the robot joints. An approach for testing velocity control and the PID controller is then introduced. This chapter concludes with a discussion of the results.

4.2 Inverse Velocity Kinematics

Unlike implementing inverse pose kinematics in Chapter 3, the Baxter robot SDK does not include a pre-made inverse velocity kinematic (IVK) solver. Consequently, this work required the solver to be programmed manually. This section introduces the mathematics of velocity kinematics. Greater detail is provided as a deeper understanding of the mathematics is required compared to using the IPK solver.

4.2.1 Forward Velocity Kinematics and the Jacobian

Forward kinematics relates joint properties to the end effector. More specifically, forward pose kinematics (FPK) solves the problem of finding the end effector pose, $\boldsymbol{X}$, for a given set of joint angles, $\boldsymbol{q}$. Likewise, forward velocity kinematics (FVK) relates the joint velocities, $\dot{\boldsymbol{q}}$, to the end effector *twist* (linear and angular velocities), $\dot{\boldsymbol{X}}$ [40]. Twist, or the end effector configuration velocity vector, is a combination of linear and angular velocities in the global cartesian coordinate frame. It is represented below in equation 4.1, where $\boldsymbol{v}$ and $\boldsymbol{\omega}$ represent the linear and angular velocities respectively, in the x-, y- and z-axis respectively.

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} = [\, v_x \quad v_y \quad v_z \quad \omega_x \quad \omega_y \quad \omega_z \,]^T \tag{4.1}$$

In terms of FVK, twist is related to the joint velocities by the following equation, where $\boldsymbol{J}$ is the *Jacobian* matrix:

$$\dot{\boldsymbol{X}} = \boldsymbol{J}\dot{\boldsymbol{q}} \tag{4.2}$$

The Jacobian is defined by [40] as: "a linear transformation, mapping joint velocities to Cartesian velocities." FVK is a result of the forward kinematics of the robot. The work of this thesis did not include deriving the Jacobian. This was achieved by using the *baxter_pydkl* package provided by Rethink Robotics [43]. This package builds upon the open-source ROS *Kinematics and Dynamics Library* (KDL) [44], and has been customised to easily integrate with the Baxter robot. The kinematic chain is generated by parsing the Unified Robot Description Format (URDF) file of the Baxter robot. The kinematic chain can then be used by the various KDL solvers, such as the FPK and FVK recursive solvers. Additionally, the Jacobian matrix can be attained by simply calling the *jacobian* method of the *baxter_kinematics* object.

4.2.2 Inverse Velocity Kinematics and the Inverse Jacobian

Inverse velocity kinematics (IVK), in contrast to FVK, is the process of calculating the joint velocities given the end effector twist. IVK is advantageous over IPK because it uses linear equations that can be solved in real-time. IPK for the kinematically redundant Baxter arm is a significant problem because there is no analytical solution. Instead, it requires a numerical procedure with an undefined number of attempts [38]. Like FVK, the end effector twist is related to the joint velocities by the Jacobian matrix, shown in equation 4.3 below. $\boldsymbol{J}^{-1}$ is the inverse Jacobian matrix.

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}\dot{\boldsymbol{X}} \tag{4.3}$$

It is important to note that the inverse Jacobian matrix can only be found when the Jacobian matrix is square and the determinant is not zero. However, the Baxter robot has 7 DoF and the inverse of a matrix is undefined for such a non-square Jacobian (6x7). To overcome this problem, the *Moore-Penrose pseudoinverse* of the Jacobian matrix, $\boldsymbol{J}^{+}$, is used. The calculation of the pseudoinverse Jacobian is shown below:

$$\boldsymbol{J}^{+} = [\boldsymbol{J}]^{T}\left[[\boldsymbol{J}][\boldsymbol{J}]^{T}\right]^{-1} \tag{4.4}$$

Similar to the Jacobian matrix, the pseudoinverse Jacobian matrix can be obtained by calling the *jacobian_pseudo_inverse* method of the Baxter KDL code. This method calculates the pseudoinverse using the *pinv* function from the open-source *NumPy* package [45]. With the pseudoinverse Jacobian matrix, the joint velocities can be calculated using equation 4.5 below.

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}\dot{\boldsymbol{X}} \tag{4.5}$$

The implementation of this equation on the Baxter robot is represented by Figure 8 below. First, a goal twist is specified. This is done in the code by initialising a starting twist with values equal to zero, then incrementing the desired variable, such as v_y. Next, the instantaneous pseudoinverse Jacobian matrix is obtained by calling the KDL method. Last, the pseudoinverse Jacobian is multiplied by the goal twist to compute the joint velocities which are commanded to the Baxter robot using the *set_joint_velocities* method. This is looped until a certain condition is met, such as displacement from the starting pose.

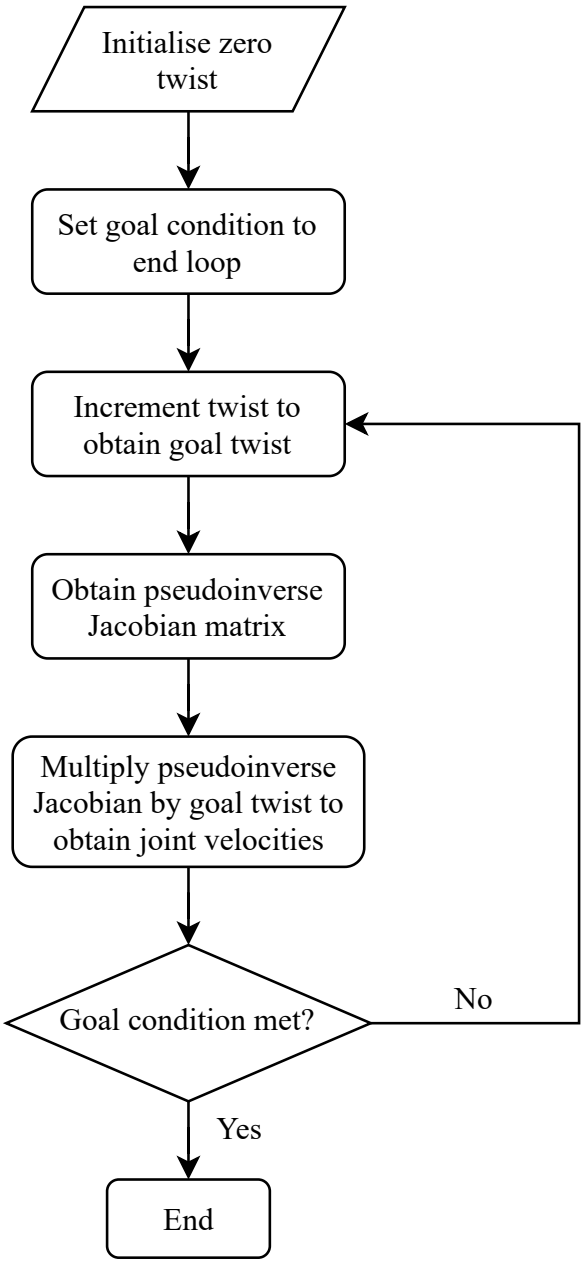

Figure 8. Flow chart of inverse velocity kinematics implementation

4.3 Velocity PID Controller

The PiH task requires fine manipulation so in an effort to improve the control, a PID velocity controller was developed. The controller was developed based off the work proposed by Lino in [46]. In this, he used pose and twist error with PID coefficients to calculate the goal twist which is then multiplied by the pseudoinverse Jacobian matrix. Notably, he used two different controllers. The first simply controls the linear

velocities while the second accounts for the end effector orientation. This work attempted to implement the PID control of the end effector orientation but was unsuccessful. Instead, the PID controller used for this thesis controls the linear velocities only. This is suitable for the objectives of this work as the end effector is commanded to a starting pose that is approximately normal to the hole surface. The PID controller used in this work is described by the following equations:

$$\dot{\boldsymbol{X}}_v = K_p\left(\boldsymbol{X}_{goal} - \boldsymbol{X}_{actual}\right) + K_d\left(\dot{\boldsymbol{X}}_{goal} - \dot{\boldsymbol{X}}_{actual}\right) + K_i \int \left(\boldsymbol{X}_{goal} - \boldsymbol{X}_{actual}\right) dt \tag{4.6}$$

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_v^{+}\dot{\boldsymbol{X}}_{\boldsymbol{v}} \tag{4.7}$$

The vectors $\boldsymbol{X}_v$, $\dot{\boldsymbol{X}}_v$ and $\boldsymbol{J}_v^{+}$ represent that only the linear velocity components were used. Therefore, $\boldsymbol{J}_v^{+}$ is a 7x3 matrix and $\dot{\boldsymbol{X}}_v$ is a 3x1 matrix. When multiplied they result in a 7x1 matrix representing the seven joint velocities commanded to the Baxter robot. K_p, K_i and K_d are the proportional, integral and derivative coefficients respectively. To start, the same values from [46] were used: $K_p = 5,\ K_i = 0.1,\ K_d = 1$. Figure 9 below is a block diagram of the PID velocity controller. The "IVK" block represents equation 4.7 which converts the end effector pose in the task space to joint velocities in the joint space. The actual end effector pose and twist are obtained by receiving a message from the *endpoint_state* topic which is published at all times. The block diagram shows that parallel control takes place because both pose and twist error are used by the PID controller. The next section of this chapter develops an approach to test the implementation of velocity control and the PID controller on the Baxter robot.

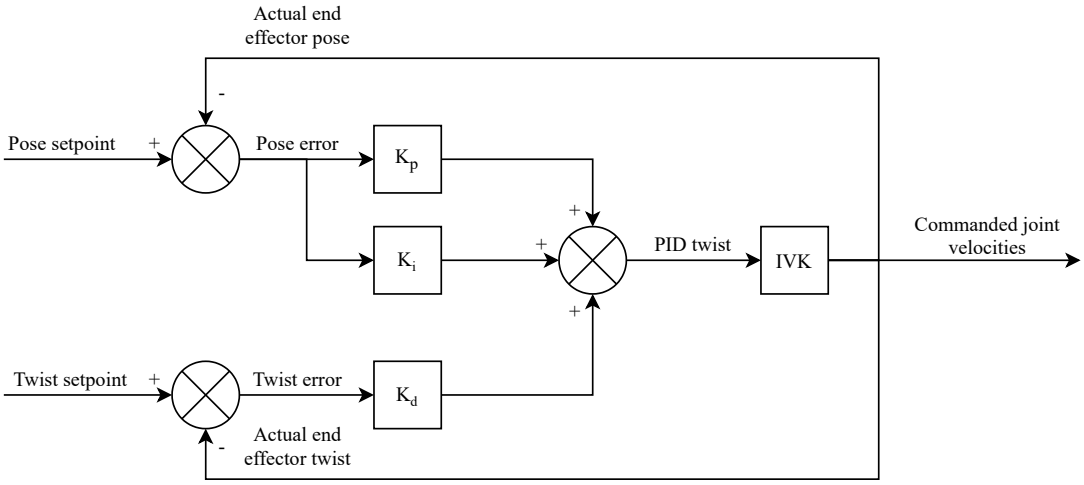

Figure 9. Control diagram of the PID velocity controller

4.4 Approach

The approach used to test velocity control of the Baxter robot was similar to the approach used to test position control in Chapter 3. This allowed for easy comparison between the two methods. All motion was constrained except for that in the positive y-direction. The end effector was commanded to follow a 10 cm straight line path. First, a test was conducted with no PID control. This was followed by a test with the PID controller. ROS bag files were recorded during both tests for playback of the results. The results are presented in the next section.

The PID controller is sensitive to the pose and twist increments that are used. For this experiment relatively small values were used as the PiH task requires small controlled movements over fast ones. The value of 0.005 was used for both the pose and twist increment. If faster movement was desired these could be increased but would trade off accuracy.

4.5 Results and Discussion

The results of this chapter confirmed that velocity control would be suitable for the final solution to the PiH task. As expected, the PID controller performed better than pure velocity control without feedback. Less noise was present throughout the path of the end effector. Additionally, the controller allows parameter tuning to alter response behaviours of the end effector which pure velocity control does not offer.

Figure 10 and Figure 11 display the end effector pose in the x-axis throughout the movement, without and with PID respectively. While there is still some fluctuation in pose in the x-direction with the PID controller, the fluctuation is much less noisy. Additionally, without the PID controller the pose in the x-direction drifts in the negative direction. However, Figure 11 shows that the feedback is being used to counteract this drift. The x-position with PID starts at approximately 0.551 mm, drifts down to 0.5465 and then moves back towards 0.552 mm. After this it moves in the negative direction until the movement is complete at approximately $t = 12\ s$.

Figure 12 and Figure 13 show the end effector pose in the y-axis throughout the movement, without and with PID respectively. Similar to the results in the x-axis, the signal with the PID controller is smoother. Further, the movement happens much faster with the PID controller than without it. The movement takes approximately 10 seconds without the PID controller, compared to approximately 2.5 seconds with the controller. As a result of the faster movement, the final y-position of the end effector is approximately 2 cm more than the commanded path of 10 cm. This is not a significant concern for the work of this thesis as the movement will be slow and not towards a known goal pose. The smoother movement with less noise is a priority of the PID controller over the final offset from the commanded pose.

Lastly, Figure 14 and Figure 15 show the end effector pose in the z-axis. Figure 15 shows the most improvement out of all the axes as a result of the PID controller. Similar to the pose in the x-axis, the results of using the PID controller are much

smoother than without using it. Figure 15 shows that during the movement ($t > 9.5\ s$) the z-position is much more constant than that shown in Figure 14 without the PID controller. The initial large drop seen in Figure 15 is a result of the Baxter joints switching from position control to joint control at the start of the movement. Testing showed that this initial drop is unavoidable and must be accounted for in the final control model.

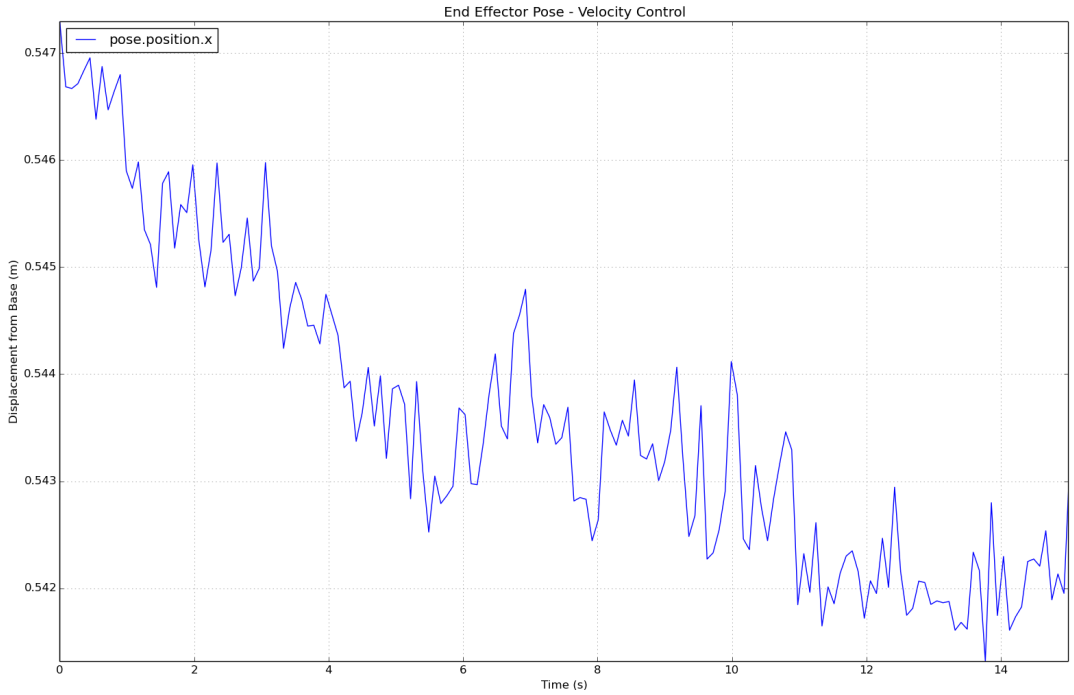

Figure 10. End effector pose in x-axis using velocity control without PID

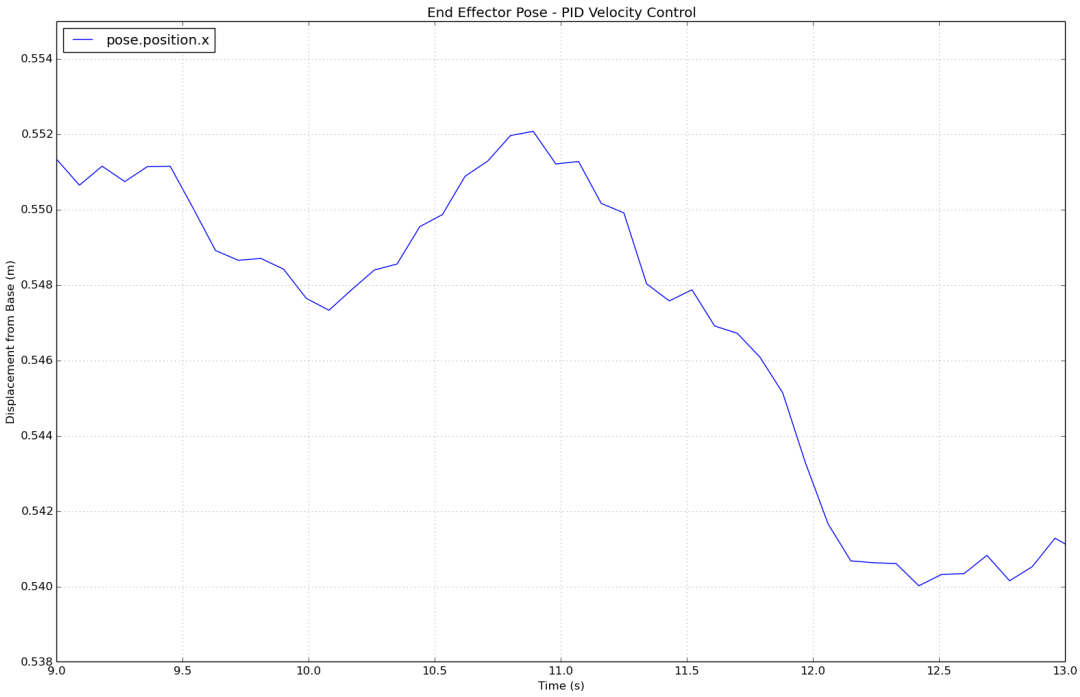

Figure 11. End effector pose in x-axis using velocity control with PID

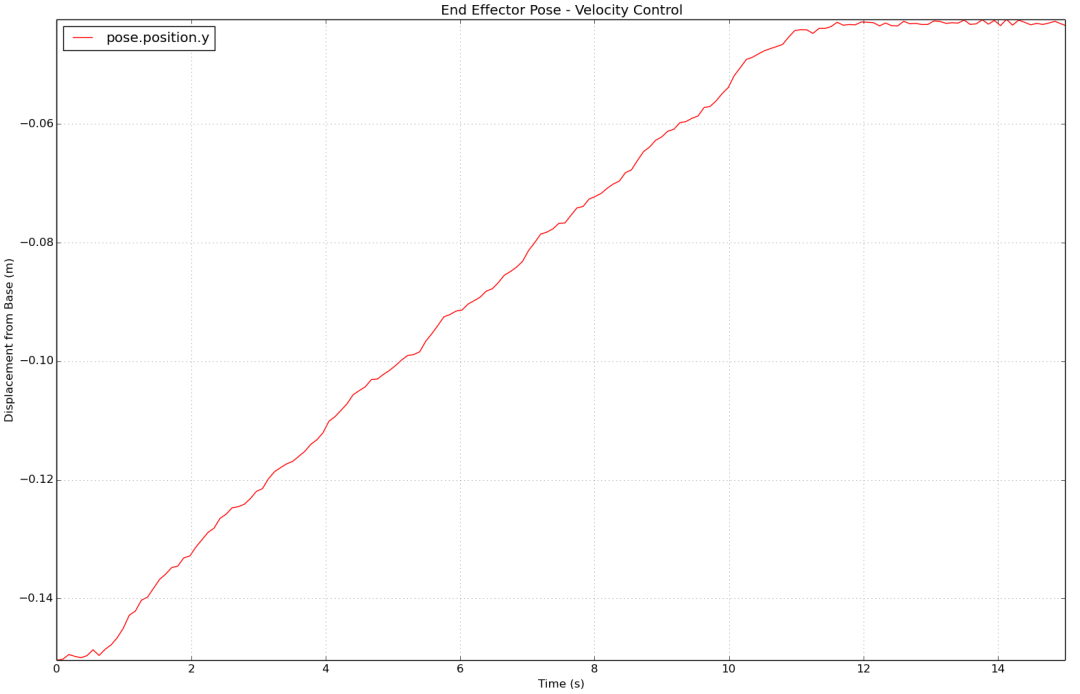

Figure 12. End effector pose in y-axis using velocity control without PID

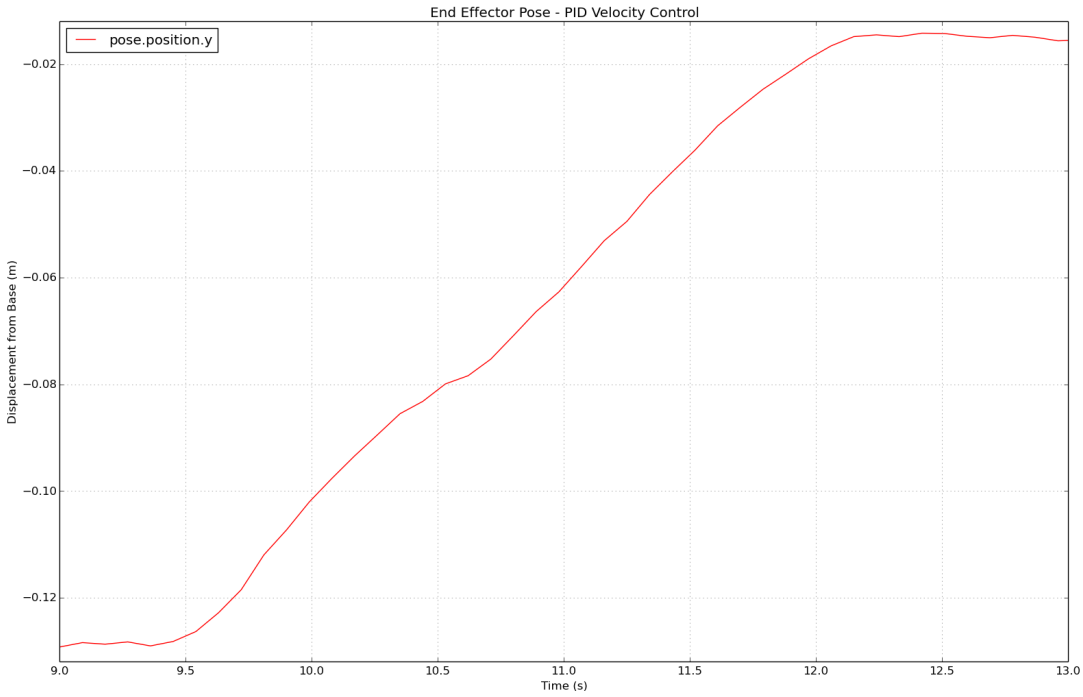

Figure 13. End effector pose in y-axis using velocity control with PID

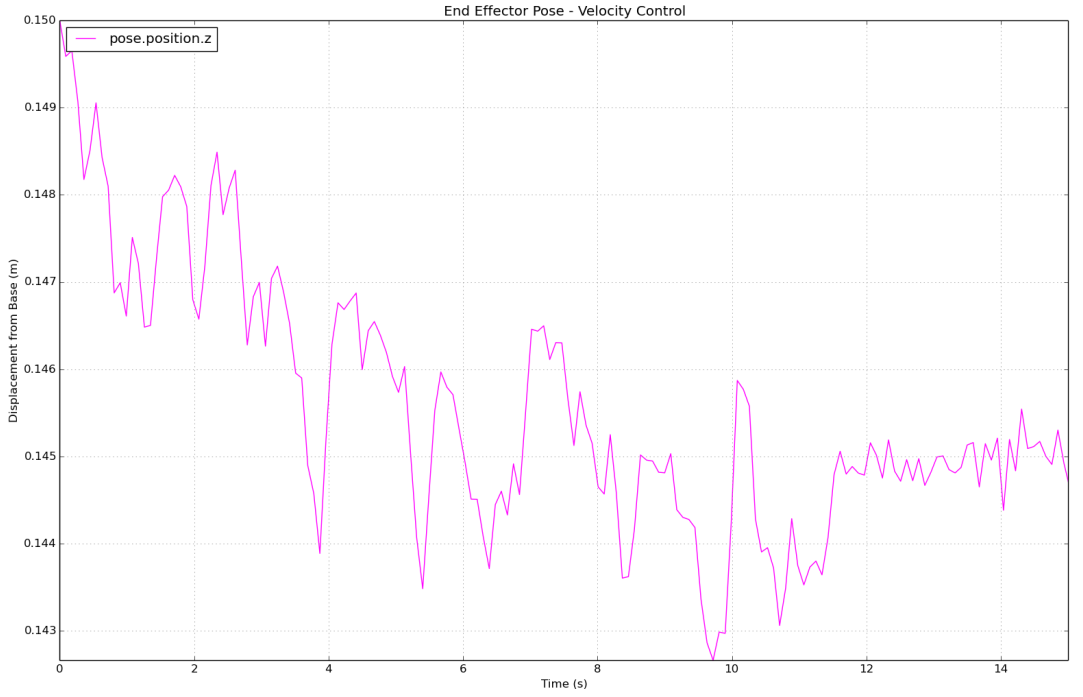

Figure 14. End effector pose in z-axis using velocity control without PID

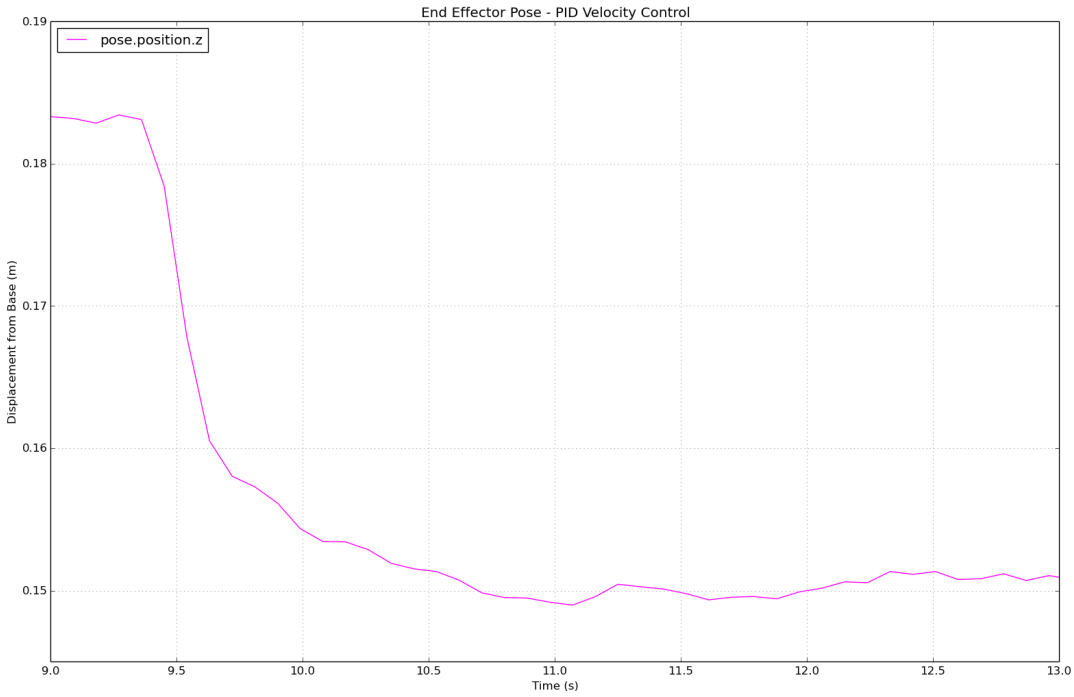

Figure 15. End effector pose in z-axis using velocity control with PID

4.6 Conclusion

This chapter introduced velocity kinematics and the relevant mathematics used to implement velocity joint control of the Baxter robot. Once velocity control was established a PID velocity controller was developed. The results of commanding the end effector to move in a straight line were presented, and it was validated that using feedback and the PID controller improved the stability and performance of the Baxter robot. The following chapter builds upon this velocity controller and combines it with force feedback.

5.0 HYBRID VELOCITY/FORCE CONTROL

5.1 Introduction

This chapter builds upon the PID velocity controller developed in Chapter 4 by combining it with force feedback from a force/torque sensor (FTS) mounted on the right wrist of the Baxter robot. The chapter develops an approach to monitor force in a given direction while constraining twist in most directions. In this manner, velocity and force will be decoupled. An approach for testing the hybrid control and noise filtering from the FTS is developed. The chapter concludes with a presentation of these results.

5.2 Force/Torque Sensor Data and Wrench

This section provides a brief background of the data received by the ATI Mini40 FTS. As a ROS standard, the force/torque data of an end effector is described as *wrench*, $\boldsymbol{\tau}$. Wrench is a vector that describes the force and torque in the cartesian task space, denoted by equation 5.1 below. The values f_x, f_y and f_z denote the force in the x-, y- and z-axis respectively while τ_x, τ_y and τ_z represent torque about the same respective axes. For the work of this thesis wrench is decoupled from the PID velocity control and only monitored; it is not used in any mathematical relations.

$$\boldsymbol{\tau} = [\, f_x \quad f_y \quad f_z \quad \tau_x \quad \tau_y \quad \tau_z \,]^T \tag{5.1}$$

The ATI FTS used in this work is launched through a C++ node which publishes the wrench data to the */transformed_world* topic, after it is biased to zero [47]. Biasing the sensor resets it to zero so the starting forces and torques present, like those from gravity or the weight of the Barrett hand, are ignored. The FTS was mounted on the right wrist of the Baxter robot as this was the "searching" arm, and the axes were aligned with the same global axis of the Baxter robot.

5.3 Noise Filtering

Initial data taken from the FTS were very noisy which made the control less reliable. One popular method of reducing this type of signal noise is through the use of Kalman filters [48, 49]. Due to time constraints this method was not used. Instead, a moving average of the ten most recent values from the FTS was used. Initially it used twenty values however this made the code take too long to execute. As a result, the joint velocity commands would sometimes exceed the timeout period and the joints would revert to position control. This led to undesirable motions and is a feature that cannot be overwritten; it is programmed into the low level joint controllers.

The code was written to perform as fast as possible, utilising the Python queue object. Only one wrench variable can be read at a time. This is in contrast to reading all wrench values, storing the ten most recent readings of all these values in memory and then passing the moving average of the six separate values. Reading just one wrench value worked well and aligns with the control because generally only one or two force/torque values need to be monitored. Motion in all other directions is constrained by the velocity controller. The approach for testing the hybrid velocity/force control in conjunction with filtered FTS data is described in the following section.

5.4 Approach

A simple experiment was conducted to test the hybrid velocity/force control and the noise filtering. The end effector was programmed to start above a table while grasping a peg. The peg face is approximately normal to the table surface but the distance between the two surfaces is unknown. After running the program, the end effector slowly moves down in the negative z-direction until a force threshold of 3 N is met. It then uses the same code written for testing the PID velocity controller in Chapter 4 to move 10 cm in the positive y-direction. The flow chart of this experiment is depicted by Figure 16 below:

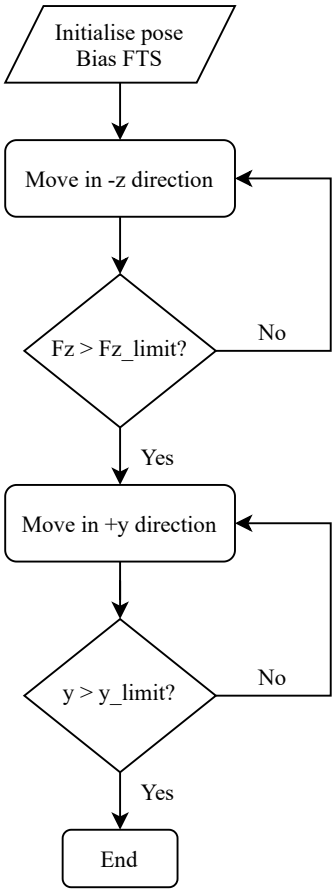

Figure 16. Flow chart of hybrid velocity/force control testing

The filtered and unfiltered force in the z-axis is monitored throughout the test to compare the results, which are presented in the next section. Figure 17 below shows the setup of the Baxter robot at the start of the experiment.

Figure 17. The Baxter robot grasping a peg to test hybrid velocity/force control

5.5 Results and Discussion

Combining force feedback with the PID velocity controller developed in Chapter 4 yielded positive results. Combining the hybrid control with noise filtering of the values from the FTS improved the control and allowed for setting a lower normal force threshold. This is beneficial for the Baxter robot as it reduces the force experienced by the joints.

Figure 18 and Figure 19 show the end effector pose in the z-direction and y-direction respectively. Figure 18 makes it clear that the end effector begins lowering at approximate t = 13 s and contacts the table at approximately t = 21 s after lowering approximately 11 cm. Figure 19 shows that there is a 1 cm change in the y-axis when the movement begins. Like previous results, this is due to the movements induced when the joints change from position control to velocity control. At approximately t =

22 s the end effector begins moving left (positive y direction) until the limit is met. At this point the movement ends.

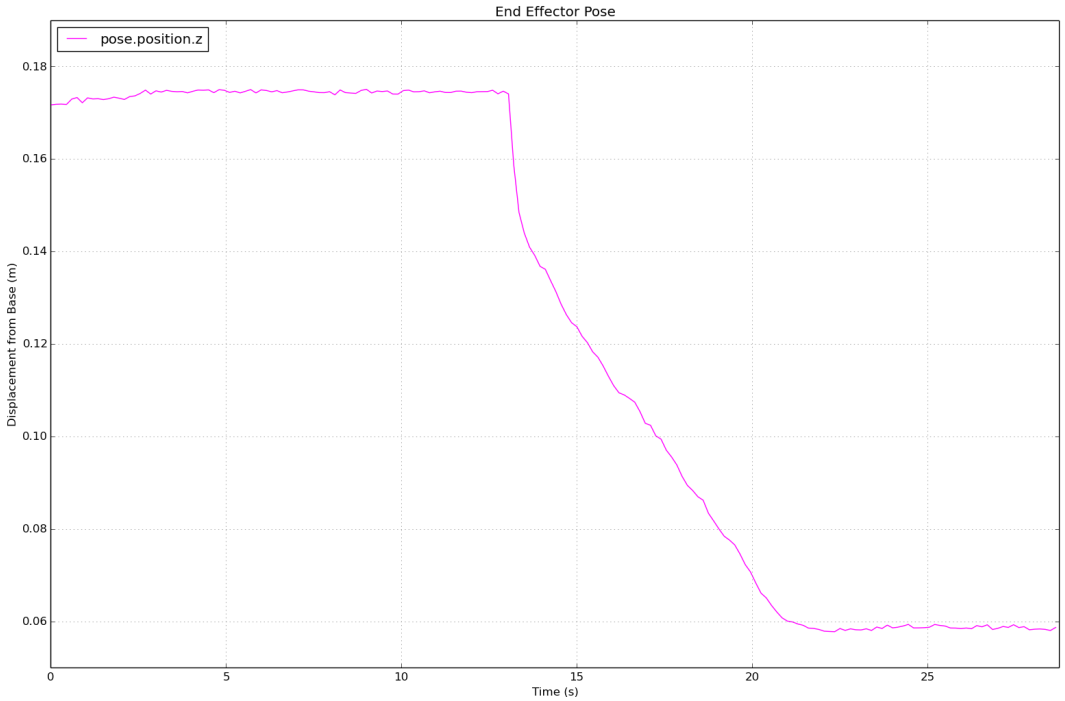

Figure 18. End effector pose in the z-axis during hybrid control testing

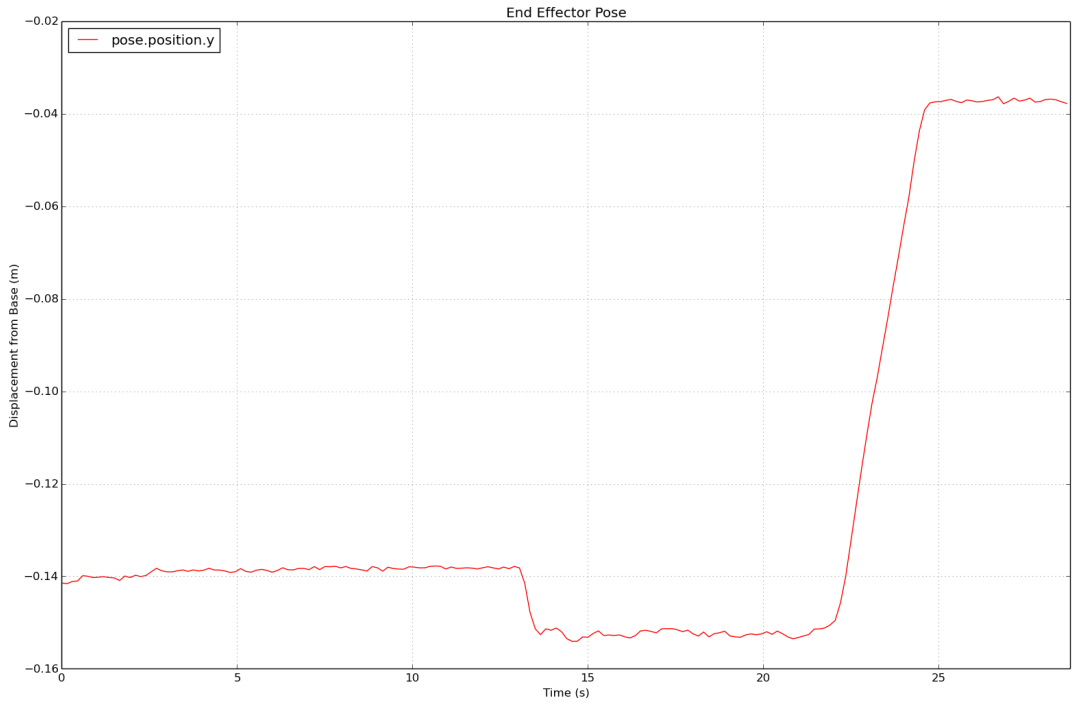

Figure 19. End effector pose in the y-axis during hybrid control testing

Figure 20 and Figure 21 below show the unfiltered and filtered f_z data, the force along the z-axis, respectively. Figure 21 validates the need for filtering the sensor data. The noise is significantly reduced throughout the entirety of the test, both during motion, and before and after. The filtered data shows the normal force is closer to 0 N before the movement begins when the end effector is in free space. Further, the maximums and minimums throughout the movement are reduced. This is advantageous because it is more likely that if a wrench limit is exceeded it is valid. Once the movement ends at approximately t = 25 s, the force in Figure 20 fluctuates approximately 1 N, from 3.9 N to 4.9 N. Meanwhile the filtered force in Figure 21 remains closer to a resting value of approximately 4.1 N.

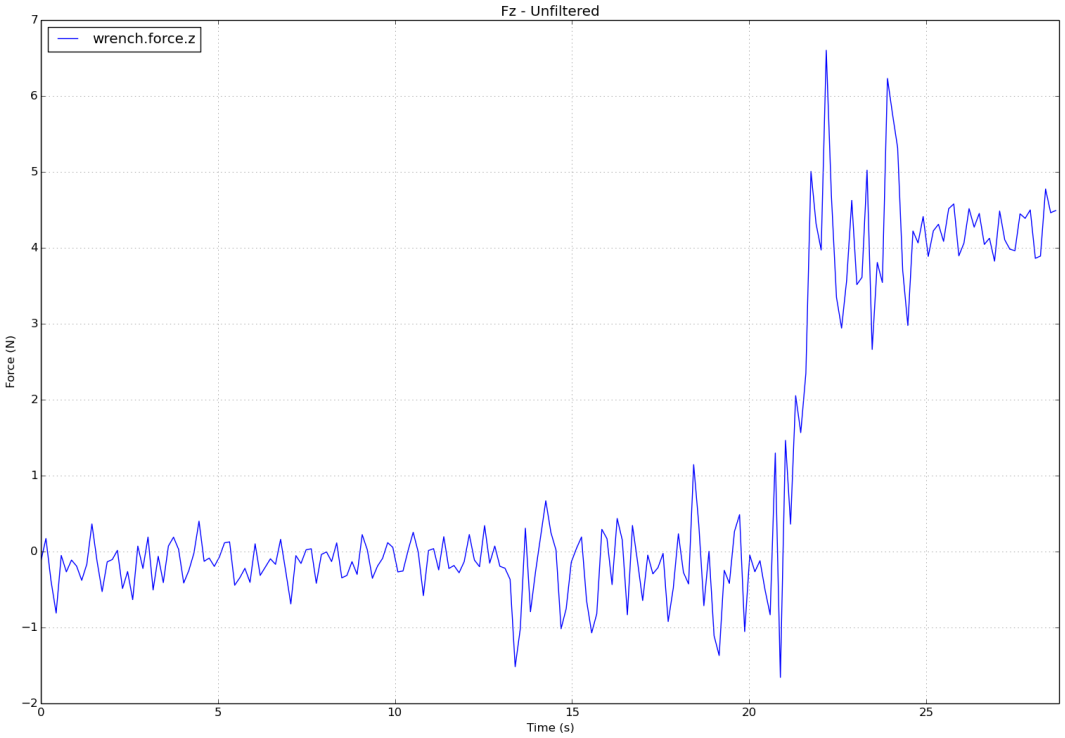

Figure 20. Unfiltered force in the z-axis during hybrid control test

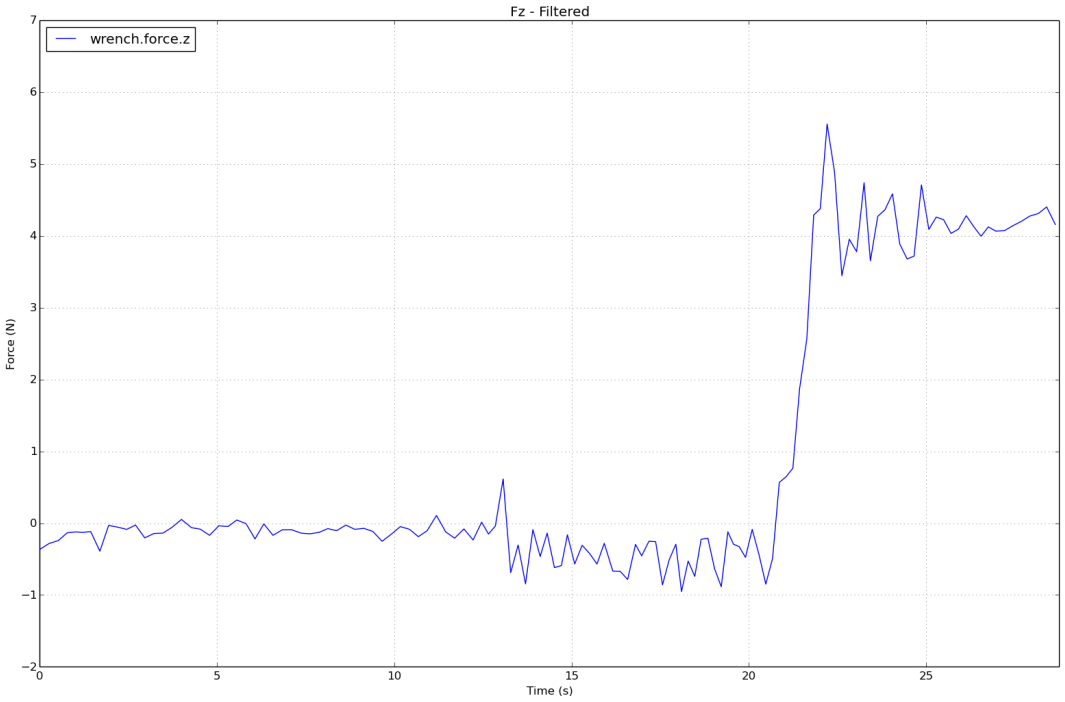

Figure 21. Filtered force in the z-axis during hybrid control test

5.6 Conclusion

This chapter developed hybrid velocity/force control by combining the PID velocity controller developed in Chapter 4 with data from a wrist-mounted FTS. The FTS hardware was introduced with an explanation of the wrench data used by ROS. Filtering noisy data from the FTS is crucial in developing a repeatable and reliable algorithm for the final PiH task. Consequently, a method for filtering the force/torque data was implemented and results validated its improved ability for wrench monitoring. This will be used in the final PiH algorithm, developed in the next chapter.

6.0 PEG-IN-HOLE

6.1 Introduction

This chapter builds upon the hybrid force/velocity control model developed in Chapter 5 and applies it to the PiH task. The overall task is split into three main phases: motion in free space, searching for the hole, and inserting the peg. The chapter explains the control of each phase independently. Further, it will develop an approach to combine them in a state machine to achieve peg insertion. It concludes with a presentation and discussion of the results.

6.2 Phase 1: Motion in Free Space

This section describes the first phase of the PiH task: motion in free space. This is when the end effector is commanded to a starting position above the hole. For this work the pose is hardcoded but theoretically it can be placed in the starting position using Zero-G mode previously mentioned. The robot does not know the coordinates of the hole but it starts off with the peg face approximately normal to the hole. It is placed so that it is approximately above the hole, but not directly over it. In this phase the end effector and peg are not in contact with any surfaces.

The movement for this phase is similar to the approach used to test hybrid velocity/force control in the previous chapter. Movement in all directions is constrained except in the z-axis. It moves in the negative z- direction until either a force threshold is met or the pose limit is reached. If the force threshold is met it enters the second phase, discussed in the next section. If the pose limit is reached then it did not make contact with a surface and the process ends. The flow diagram is shown in Figure 22 below. The variable *z_limit* represents the pose limit. Since the end effector is moving down it must fall below this limit to end the search.

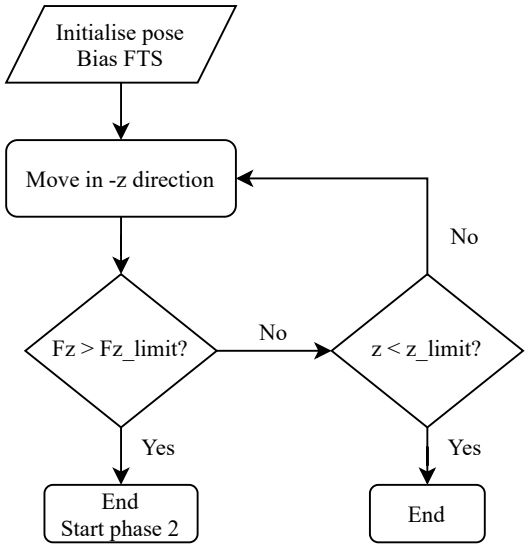

Figure 22. Flow chart of PiH phase one

6.3 Phase 2: Search

The second phase of the PiH solution developed in this work is the search phase. This phase begins once the free motion phase ends, assuming the end effector makes contact against a surface. This phase defines a small square, with the centre being the pose at which phase one ends and phase two begins. The end effector will move in a grid pattern within this small square until either: the x-position limit is reached or the normal force drops below a threshold limit. If the pose limit is met then the hole has not been found and the process ends. If the normal force falls below the threshold this phase will end and the final insert phase will begin.

A flowchart of this phase is shown in Figure 23 below. To avoid overcrowding, this flow chart only shows how the pose affects the movement during the search phase. It is important to note that at all times normal force is monitored and if it falls below a threshold the search phase ends and the insert phase begins. The flow chart shows that the end effector will move in three directions: positive and negative in the y-axis and positive in the x-axis. This was to simplify the code and state machine. If the hole is not detected by the time the end effector reaches the x-limit the process ends.

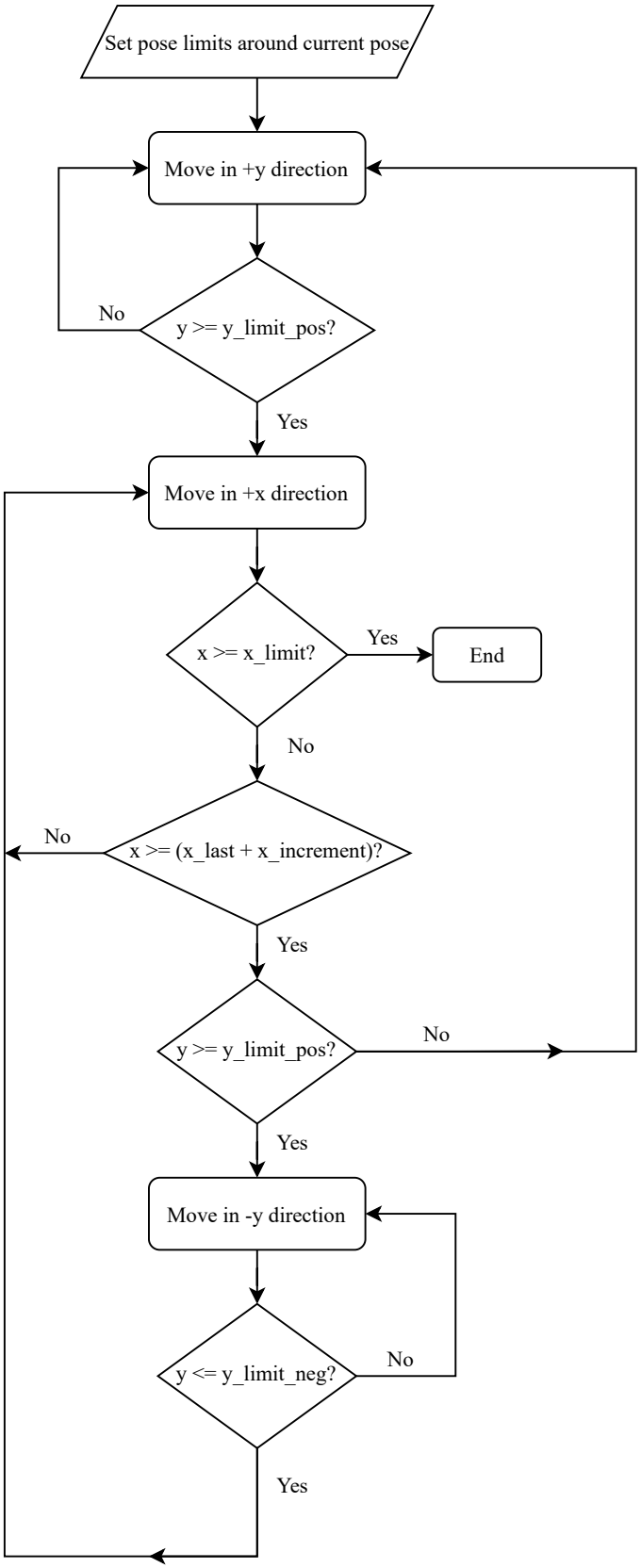

Figure 23. Flow chart of PiH phase two movement

The variables *y_limit_pos* and *y_limit_neg* denote the edges of the grid, dictated by the pose at the start of this second phase. Respectively, they are the limits in the positive and negative y-direction relative to the starting pose. For example, if the y-position when phase two begins is 0.000 m, they could be set at ±1cm. Therefore, *y_limit_pos = 0.010 m* and *y_limit_neg = -0.010 m*. Using this method, a target grid can be created relative to the end effector's starting pose instead of a known goal pose. In a similar manner, *x_limit* would +1 cm in the positive x-direction. The variable *x_last* represents the last x-position prior to starting the movement in the x-direction This is updated when each y-direction movement ends. The *x_increment* is the distance the peg will move in the positive x-direction until movement in the y-direction is triggered. For example, if the peg has finished moving in the positive y-direction it updates *x_last* to the current x-position, say 0.020 m. The increment is hardcoded and for this example is set to 0.5 cm. Therefore, the end effector will move in the positive x-direction until it reaches *x_last* + *x_increment*, in this case 0.025 m. Once it has reached this pose it will resume movement in the y-axis and 0.025 m would be the new *x_last*. Figure 24 below illustrates the pattern of movement and the limits. The black circle is the starting pose.

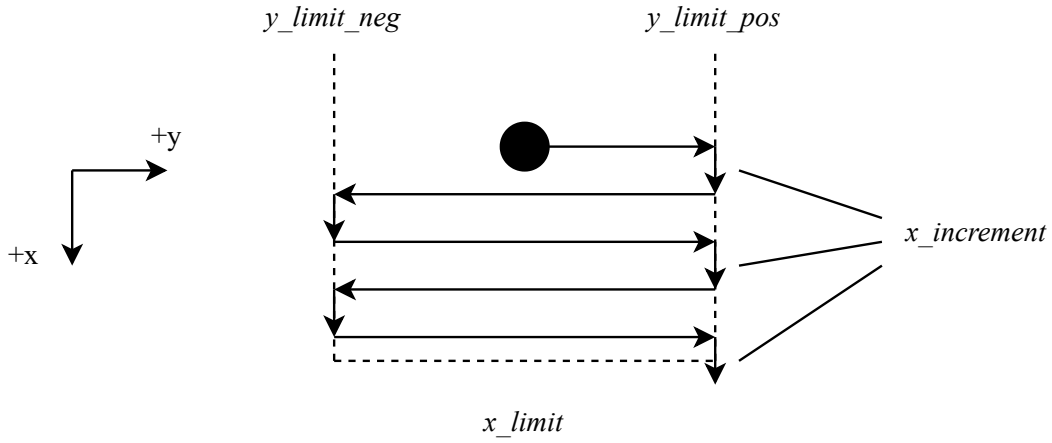

Figure 24. PiH phase two search pattern

6.4 Phase 3: Insert

The insert phase is the final phase of the PiH task developed in this work. It begins if the normal force during the search stage drops below a threshold force. Once in this phase the robot must:

1. Insert the peg;
2. Confirm the peg is inserted, and;
3. Remove the peg.

These steps are described in detail below and are treated as separate states within this phase of the PiH task.

6.4.1 Step 1: Insert Peg

The peg is inserted by moving the end effector in the negative z-direction until either a normal force threshold is exceeded, or a pose limit is met. The pose limit is defined by the hole depth and is added to the pose at the time this phase begins. If the normal force threshold is exceeded then the peg has made contact with either the bottom of the hole or some other surface. If the pose limit has been met then the peg has moved off the hole face surface but is not in the hole; the process ends. An example of this is if the peg moves off the hole face during the search causing the normal force to drop and phase three to begin. The peg will be pushed in the negative z-direction but reach a predefined limit in this direction, causing the process to end. However, this alone could result in false positives if, for example, the peg comes into contact with a surface before the pose limit is reached. To avoid false results the robot must confirm that the peg is inserted.

Figure 25 below is the flow chart of the first stage of this phase: inserting the peg. The variable *z_last* is the z-position when this phase begins and *hole_depth* is predefined depending on the hole being used. It is subtracted from *z_last* to define a z-limit which relies on the current end effector pose instead of a known pose. The limit *fz_limit* is a normal force limit defined by the user which will trigger exiting this stage and entering the next if it is met.

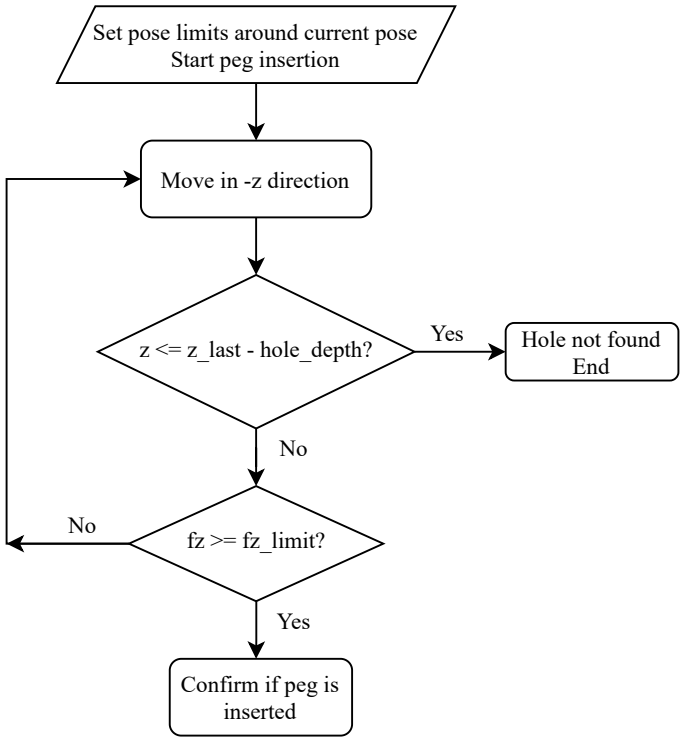

Figure 25. Flow chart of PiH phase three, step one: peg insertion

6.4.2 Step 2: Confirm if Peg is Inserted

Once the peg has been "inserted" and a normal force threshold exceeded, the robot must confirm that the peg is actually inserted to avoid false positive results. This is achieved by applying velocity in the positive and negative x- and y-axis. The applied velocities generate respective torques which the FTS can read and compare to predefined limits. If these limits are met in every direction the peg has successfully been inserted. If, instead, the end effector moves beyond a certain pose limit in any of these directions without meeting the torque limits the peg was not successfully inserted

and the process ends. The pose limits are predefined and added to the current pose at which this process starts, similar to the way the grid search limits were defined in the search phase.

Figure 26 is the flow chart of the second step within the insert phase. Here the robot is testing to confirm that the peg is in fact inside the hole, after it has progressed from the first step of inserting the peg. Each loop of applying velocity and testing the torque in a certain direction is treated as a sub-state within this step. The *y_last* and *x_last* variables denote the y- and x-positions when this step begins. The *y_limt* and *x_limit* variables are defined by the user and should be similar but larger than the peg/hole clearance. They regulate how far the peg will need to move before triggering the end of the process. Generally they would equal each other for a round peg/hole pair. The *ty_limit…* variables are torque limits in each direction of motion. In this work they are obtained experimentally by observing the values through the terminal output.

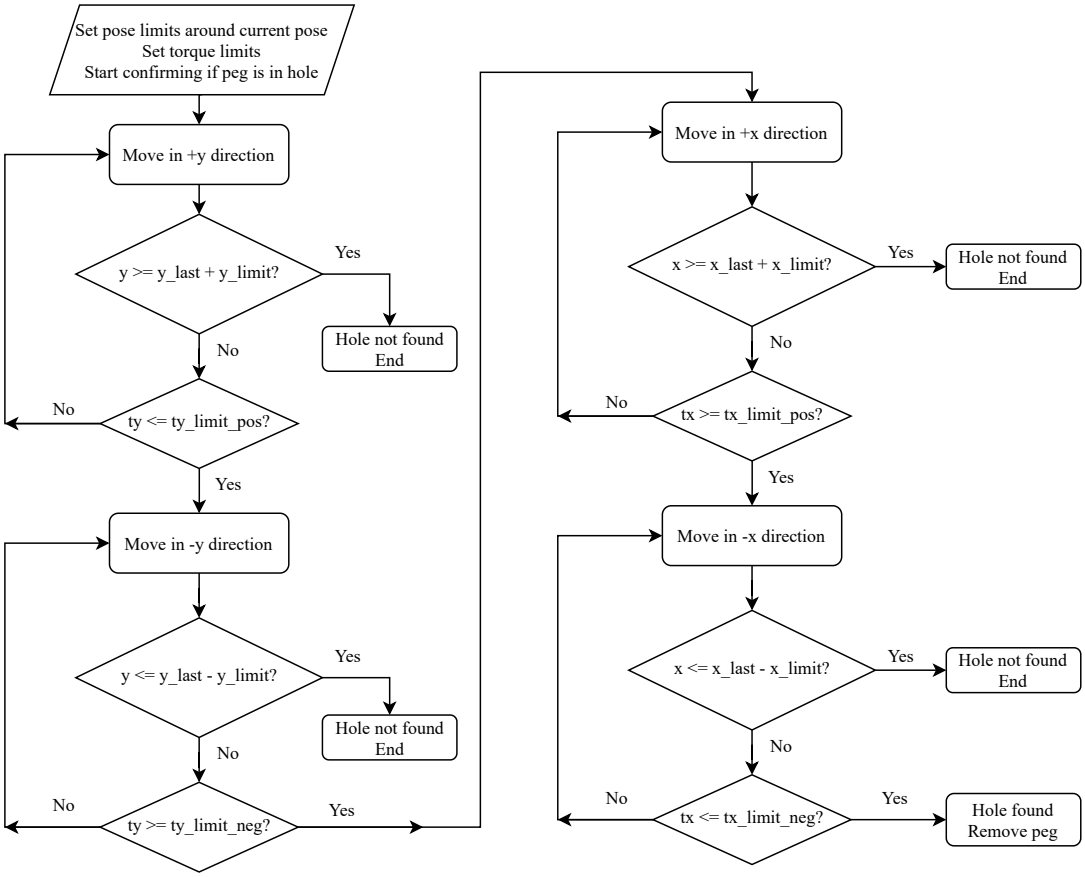

Figure 26. Flow chart of PiH phase three, step two: confirming if peg is in the hole

6.4.3 Step 3: Remove Peg

If the peg has been successfully inserted into the hole the final step of the process is to remove the peg. This is simply achieved by moving the end effector in the positive z-axis. Since the end pose of the hole is unknown before the task begins, the distance it must be moved is simply added to its current z-position at the end of the second step in this phase.

6.5 Approach

This section describes the approach taken to combine the phases of the PiH task described earlier in this chapter. The PiH algorithm was designed as a state machine with several high level states and lower level sub-states for the various phases of the task. The three phases outlined previously serve as the three primary states. The third "insert" phase is further decomposed into substates based on the three insert stages already outlined: insert, insert confirmation, and peg removal. The second substate of this phase, insert confirmation, is divided into four further substates. These four states represent testing the torque experienced by the peg in the positive and negative x- and y-directions of movement. Figure 27 below represents the overall state machine used for this work. The previous sections of this chapter have provided the low level details of each state shown in this figure. This figure aims to show the high level implementation and integration with all the other states of the system. All limits were derived experimentally and specified in the code implementation.

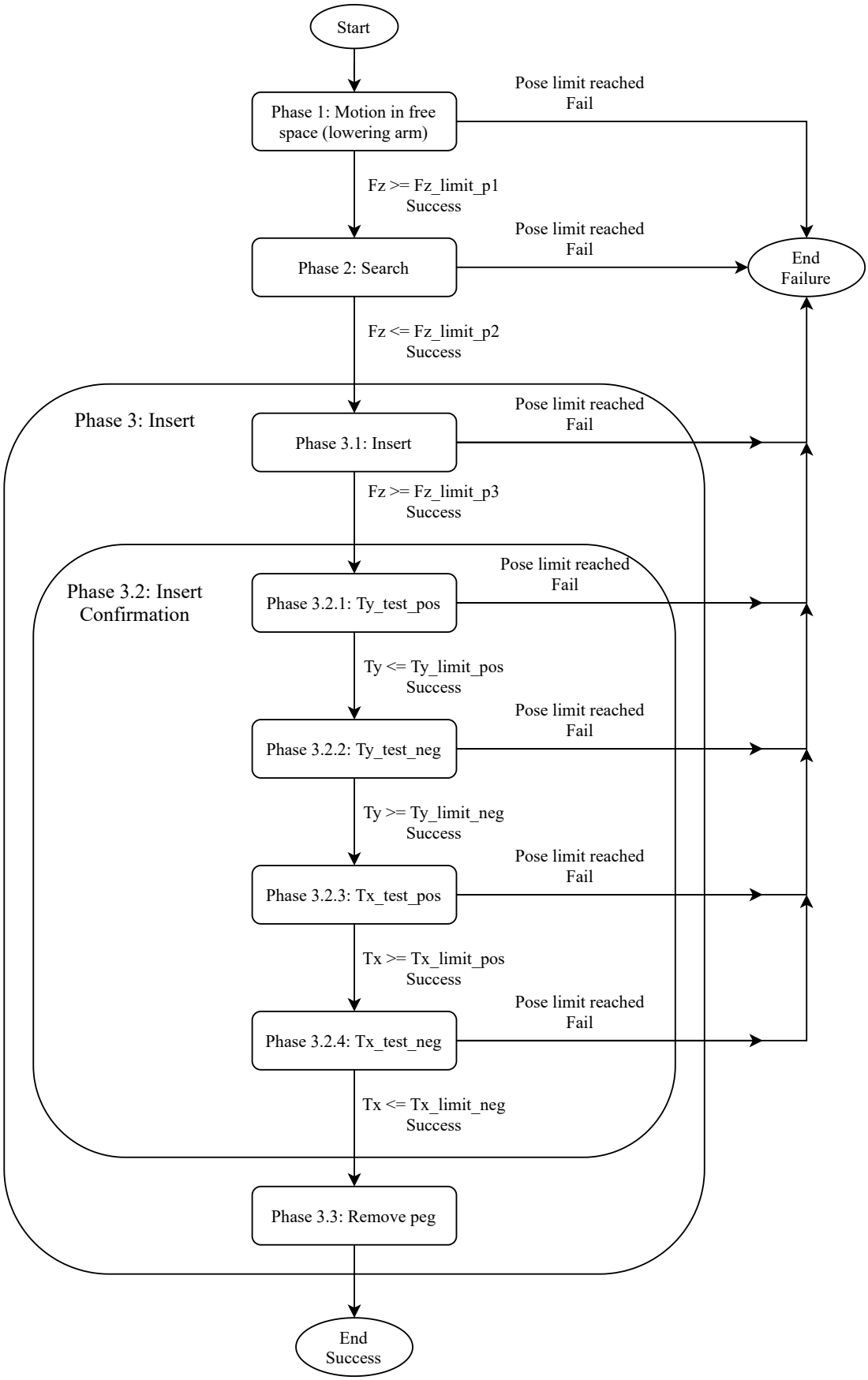

Figure 27. State machine diagram of the overall PiH solution

The dimensions of the peg/hole pair used for this experiment were:

- Peg outside diameter of 20 mm and
- Hole inside diameter of 23 mm.

This gave a radial clearance of 1.5 mm either side of the peg when centred in the hole. The peg and hole used were both plastic parts that could be found in a hardware store. They are shown below in Figure 28.

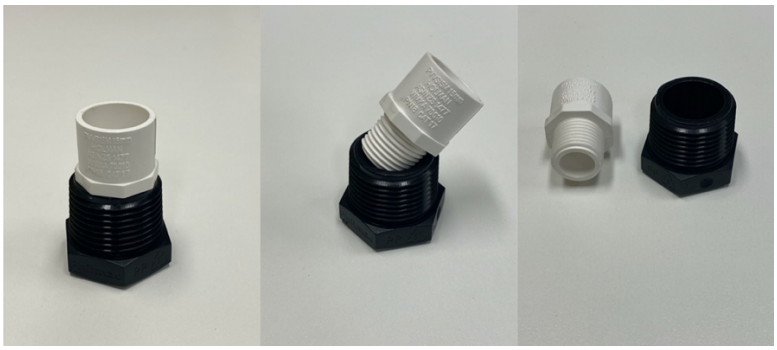

Figure 28. The peg and hole used during final testing

Figure 29 is an image of the Baxter robot at its starting pose for the final test. As shown, the peg is grasped by the right arm while the hole is secured to the table with clamps. The contact surface of the hole is relatively small in contrast to a solid surface with a hole in it. This is more representative of real life conditions when a robot might be tasked to assemble to pipe ends together, for example. Due to the small contact area the search grid defined was relatively small: ± 5 mm from the starting pose of the search phase.

Figure 29. Baxter setup during final testing

6.6 Results and Discussion

This section presents and discusses the results of the final PiH task performed on the Baxter robot during this work. Images of the Baxter robot throughout the PiH task will be shown to provide a visual of the state machine in action. Additionally, graphs of the pose and wrench values will be examined.

Figure 29, above, shows Baxter at the start of the program. Figure 30, Figure 31 and Figure 32 below shows the peg during the first phase while the end effector is being lowered, and then when it makes contact with the peg and enters the second state: search.

Figure 30. Phase 1 of PiH task while the end effector is lowering the peg

Figure 31. Phase 1 of the PiH task as the peg approaches the hole surface

Figure 32. Phase 1 of the PiH task as the peg makes contact with the hole surface

Figure 33 depicts the second phase of the PiH solution developed: search. Once this phase begins the peg can be seen moving in the grid pattern illustrated by Figure 24 previously.

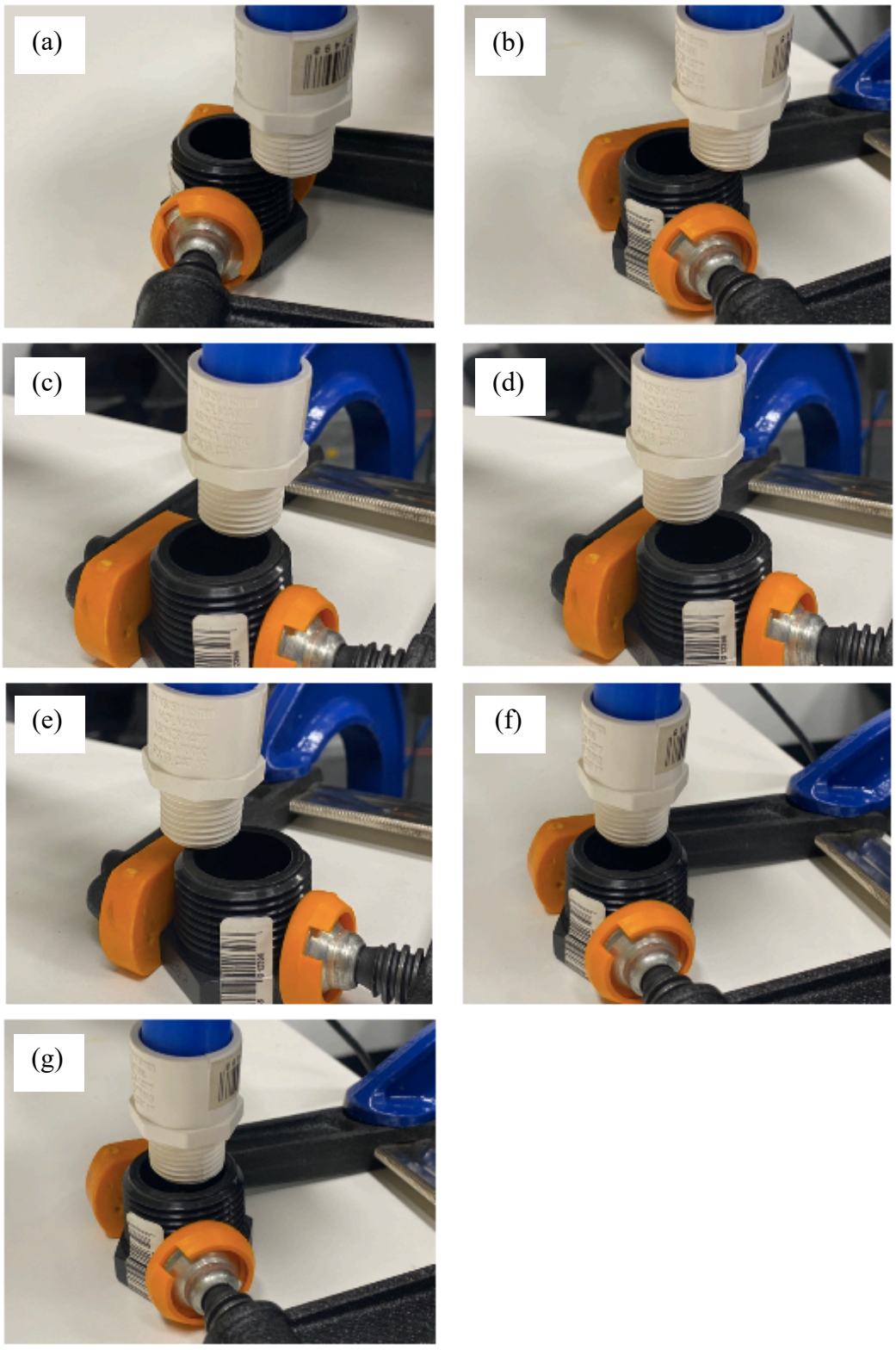

Figure 33. Phase 2 of the PiH task: search. (a-b) show the initial movement in the +y direction. (c-d) show the movement in the -y direction. (e-f) show the movement in the +x direction. (g) shows the peg going over the hole and finishing this phase after the normal force drops below the threshold.

Figure 34 below represents the final phase of the PiH task: insertion. The peg can be seen moving into the hole. The torque tests are difficult to observe through images but the graphs will be presented later in this section. The figure also shows the peg removal when the process ends.

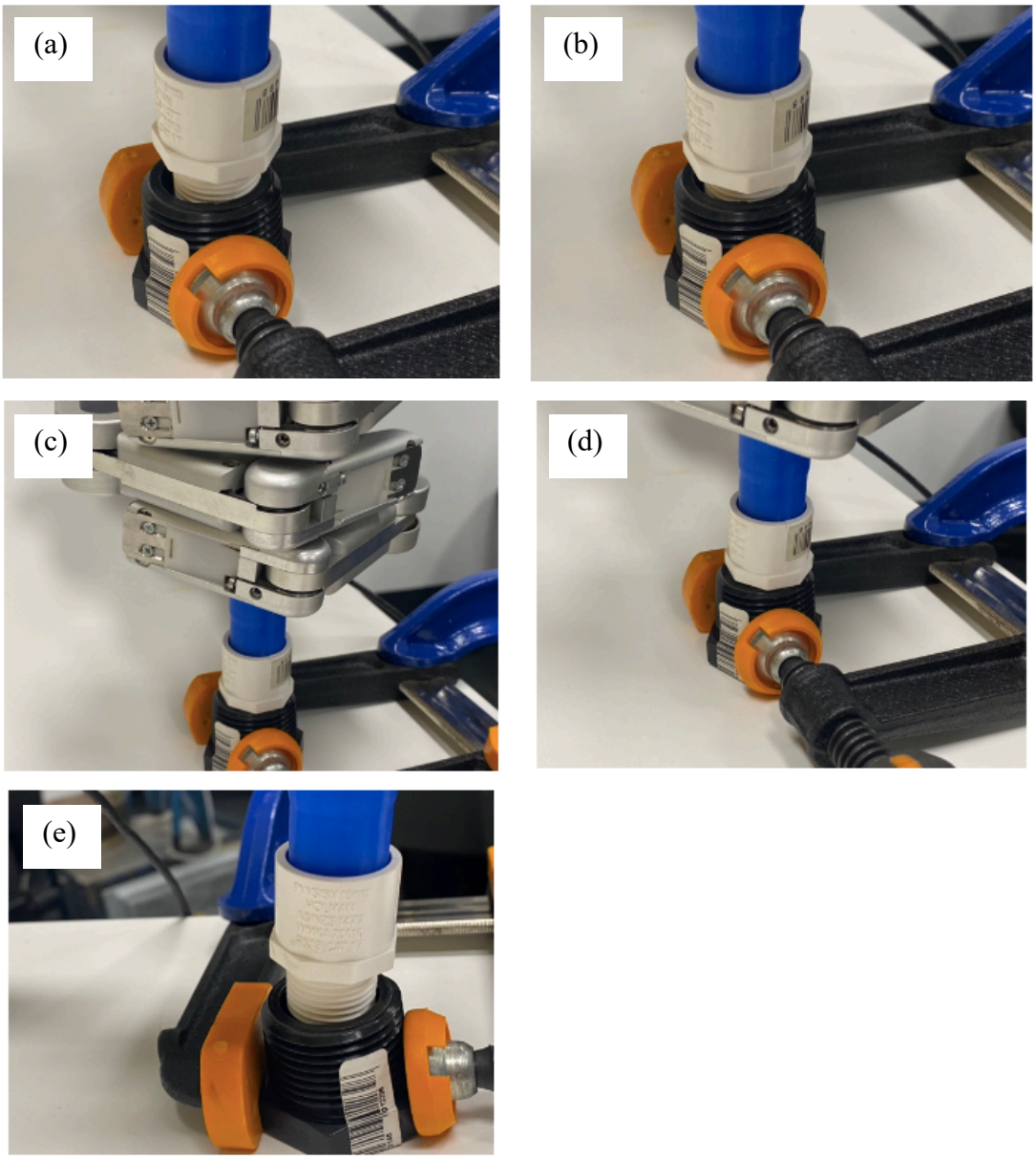

Figure 34. Phase 3 of the PiH task: insertion. (a-b) show the peg being inserted. (c-d) show the torque tests being implemented. The entire process ends in (e) when the peg is removed from the hole.

The next set of figures are the graphs of the end effector pose and wrench throughout the PiH task. Figure 35 below shows the end effector pose in the z-axis. This graph shows that phase one begins at approximately $t = 7.5\ s$. The peg is lowered until $t = 15\ s$ when the normal force threshold is met. Between $t = 15\ s$ and $t = 36\ s$ the z-position remains relatively constant while the robot is searching for the hole in phase two. At $t = 36\ s$ the z-position drops as a result of f_z dropping below the set limit. At this stage the robot enters its final phase: insertion. From $t = 36\ s$ to $t = 44\ s$ the robot is performing the torque tests to confirm the peg is inserted in the hole. At $t = 44\ s$ the robot removes the peg and the process ends.

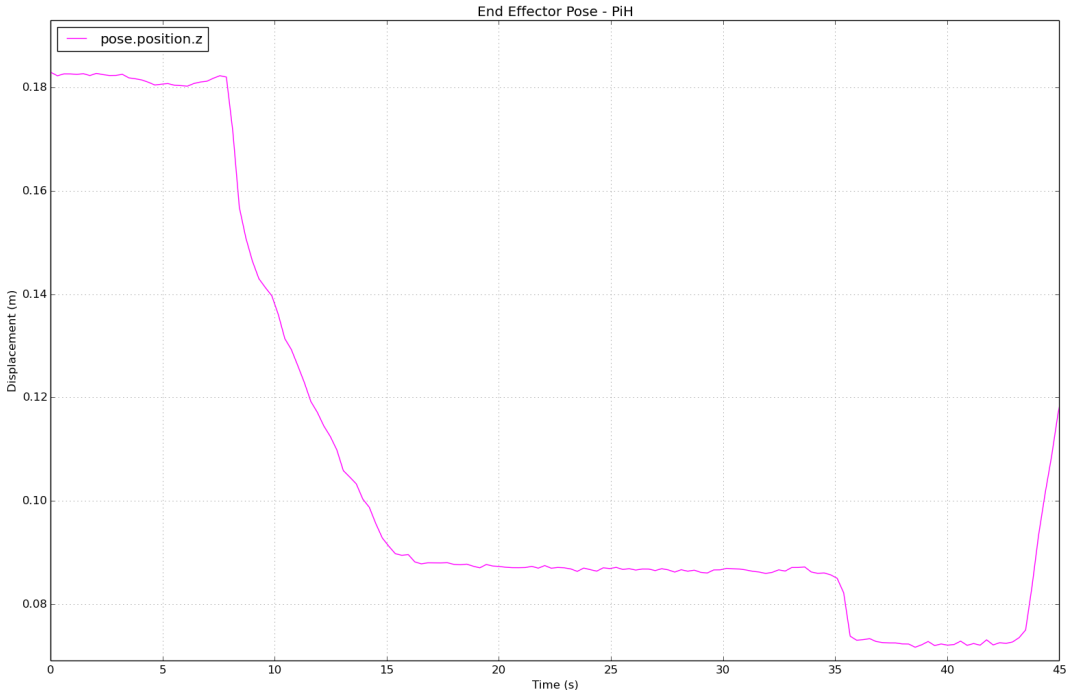

Figure 35. End effector pose in z-axis during PiH task

Figure 36 shows the end effector pose in the x-axis. There is a sharp change when the process starts due to the joints entering velocity control mode but it can be seen that the velocity controller corrects this. It overshoots its starting pose by approximately 2 mm. The large changes in the x-position at $t = 20\ s$ and $t = 27\ s$ represent the x-increments from the grid search pattern. It increments first when the positive y-limit is met and then again when the negative y-limit is reached. The x-position stays relatively

constant from $t = 35\ s$ to $t = 40\ s$, then there are large changes. These are a result of the torque tests in the positive and negative x-directions. It first moves in the positive direction at $t = 41\ s$ and then at $t = 42\ s$ moves in the negative direction once the first torque limit is reached in the x-axis. The large changes at the end of the process occur while the peg is being removed.

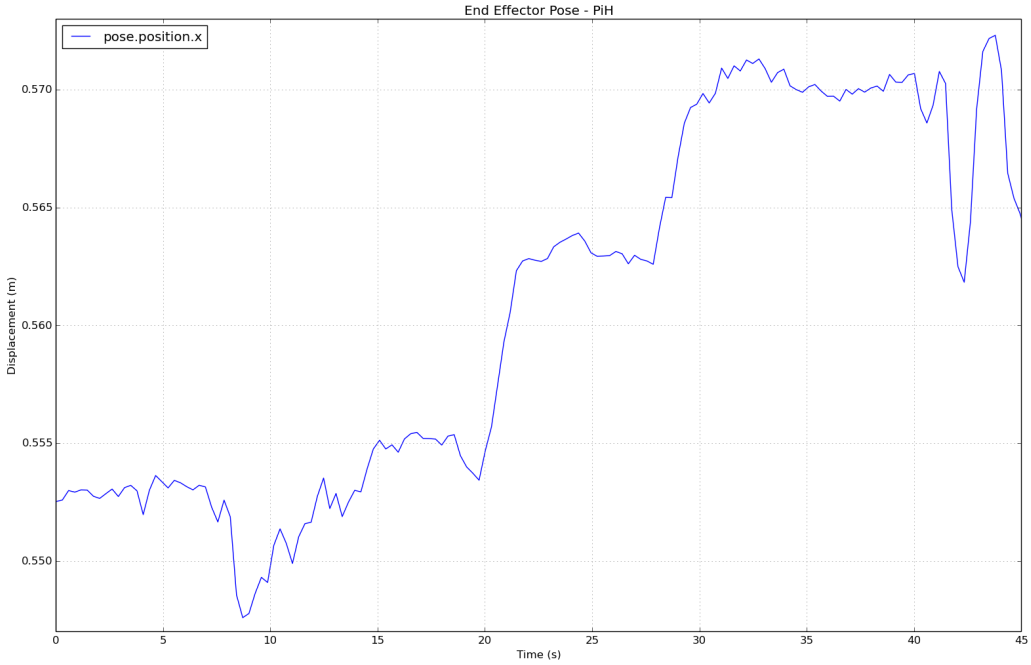

Figure 36. End effector pose in x-axis during PiH task

Figure 37 below shows the pose of the end effector in the y-axis. Similar to the position in the x-axis, there is a sharp change at $t = 7.5\ s$ when the joints change to velocity control mode. The velocity controller corrects for this until $t = 15\ s$ when the peg comes into contact with the hole surface and transitions to the search phase. From $t = 15\ s$ to $t = 20\ s$ the peg moves in the positive y-direction while it searches for the hole. At $t = 20\ s$ it reaches the positive y-limit. There is a small change in the y-position while it increments in the positive x-direction, and then the end effector moves in the negative y-direction relatively linearly until $t = 28\ s$ when it reaches the negative y-limit. There is again another small change in the y-direction while the peg is moved in the positive x-direction. From $t = 30\ s$ to $t = 36\ s$ the end effector moves in the positive y-direction at a relatively constant rate until the f_z limit is met and the robot transitions

to the insertion phase of the program. There are sharp changes in the y-position between $t = 38\ s$ and $t = 42\ s$. This is when the torque tests are made to confirm the peg is correctly inserted.

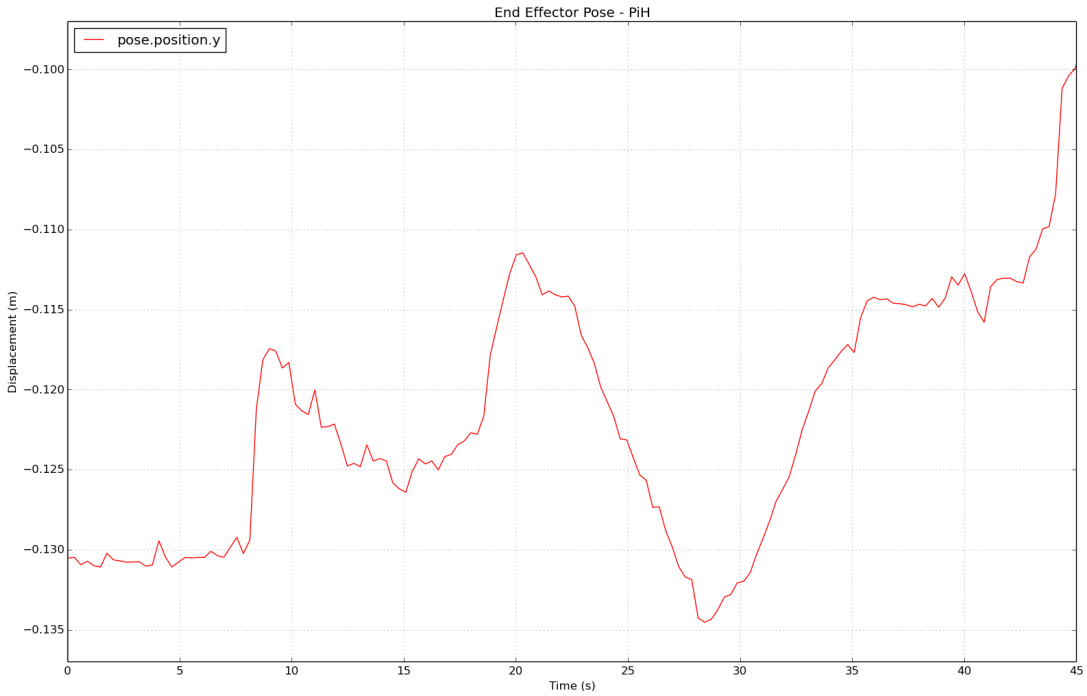

Figure 37. End effector pose in y-axis during PiH task

The last figures to be shown will be the end effector wrench. It is important to note that these will show the raw values measured from the FTS, not the filtered values used by the control algorithm. The attempts to publish the filtered sensor values were unsuccessful. The execution time to filter and publish the sensor values while also running the program was not fast enough and affected the robot's movement. Consequently, the following graphs are very noisy.

Figure 38 below shows the end effector force in the z-direction (f_z) throughout the PiH task. There is a drop in the force when the phase one movement begins and it stays lower until approximately $t = 16\ s$ when the peg contacts the hole surface. From the noise present in this data it is difficult to distinguish the different states. However, from Figure 35 which shows the z-position, it is known that the hole is found at

approximately $t = 35\ s$. Additionally, the normal force limit was set to 1.2 N. This is first met at $t = 31\ s$ but was likely filtered out as noise. The two drops in normal force at $t = 34\ s$ are likely what were captured as the normal force after filtering the noise and triggered the second phase of insertion. At $t = 35\ s$ the normal force sharply rises as a result of the peg being lowered and contacting the bottom of the hole. The large drop in normal force at $t = 43\ s$ to below 0 N is when the peg was being removed as the final step of the process.

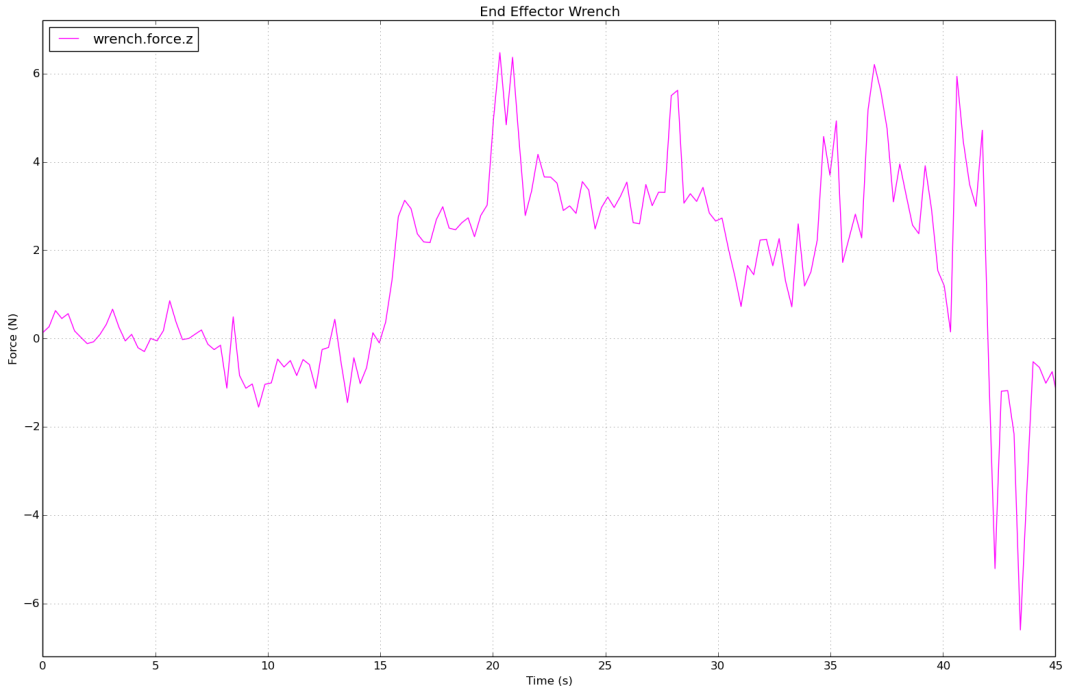

Figure 38. End effector force in z-axis (f_z) during PiH task

Figure 39 shows the torque of the end effector about the y-axis. The signal is noisy after an initial spike to 0.18 Nm at $t = 16\ s$ when the peg contacts the hole and begins the search phase. The large spikes between this point and $t = 35\ s$ are a result of the peg performing the grid search against the hole surface. The friction of these movements create variable torque as the end effector motion changes throughout the search phase. The notable points in this figure are at $t = 39\ s$ and $t = 41\ s$. This is when the peg has been inserted and begins the torque tests to confirm that it is inside the target hole. The positive y-motion causes t_y to fall below its limit of 0.1 Nm at

approximately $t = 39\ s$. It then moves in the negative y-direction until τ_y meets the second limit of 0.25 Nm. It then begins testing torque in the x-axis.

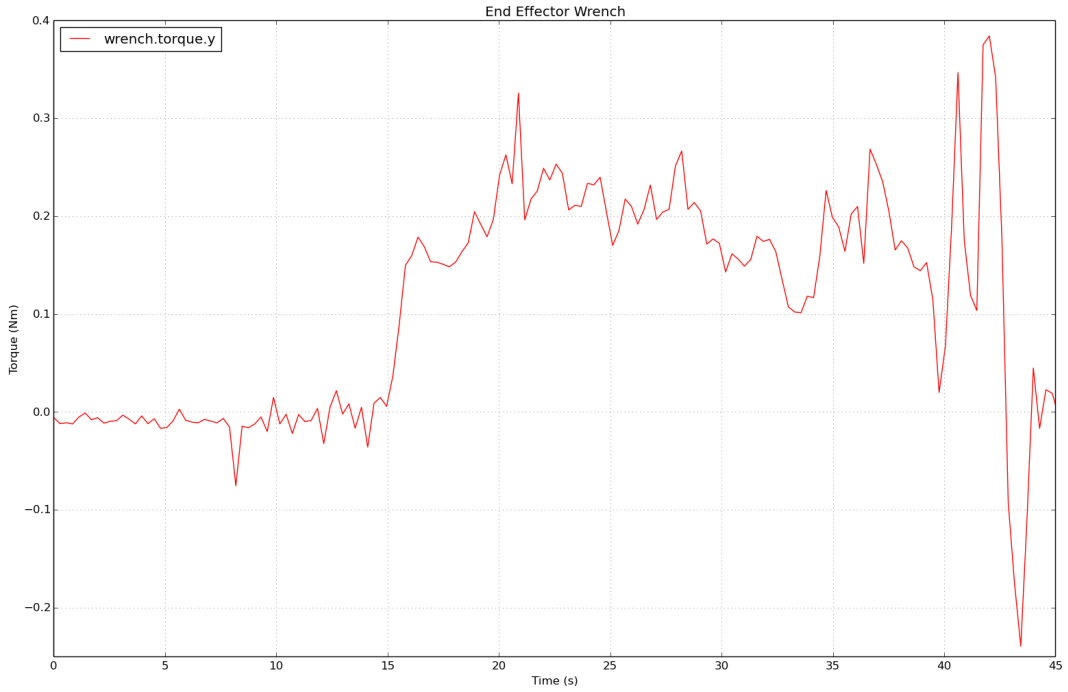

Figure 39. End effector torque about y-axis (τ_y) during PiH task

Figure 40 below is the end effector torque about the x-axis, τ_x. Similar to the previous figure, the large fluctuations begin at approximately $t = 7.5\ s$ when the end effector begins the PiH task. τ_x drops until $t = 16\ s$ when the peg contacts the hole. The signal is noisy while the robot searches for the hole as a result of the contact friction between the peg and hole surfaces. The largest τ_x values occur while the robot is testing that the peg is inserted in the hole as part of the final phase. At $t = 41\ s$ the motion in the positive x-direction causes the torque to rise above the set limit of – 0.05 Nm. Then, as part of the final substate in this phase, the motion in the negative x-direction causes the torque to fall below the limit of -0.30 Nm. With the final torque test complete the peg is confirmed to have been successfully inserted. This phase ends with the removal of the peg and the end to the overall process.

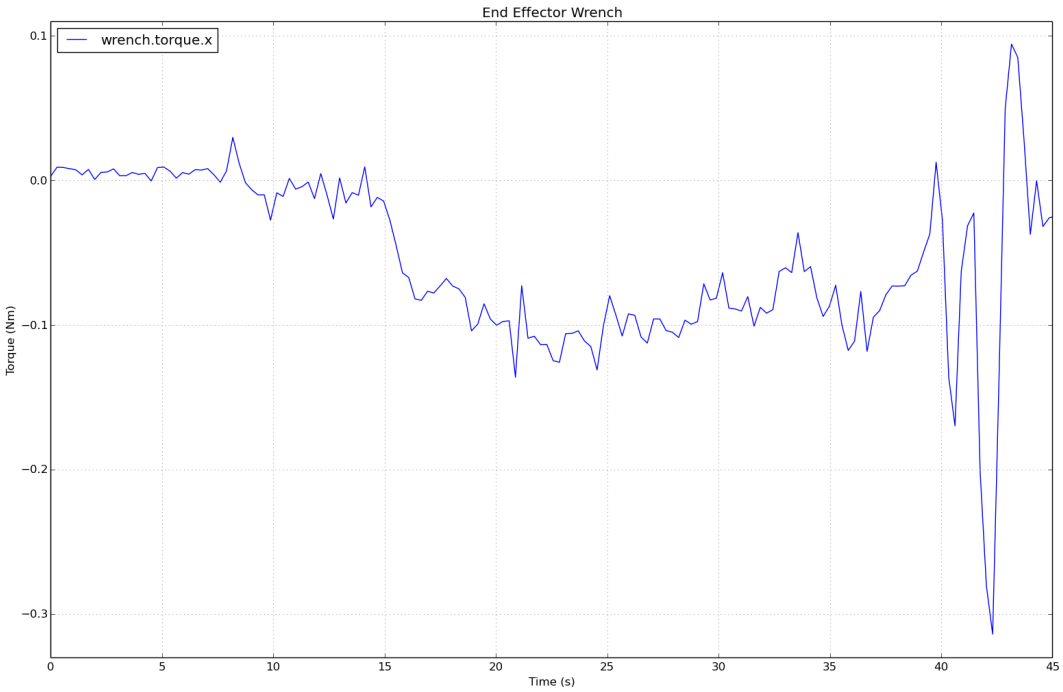

Figure 40. End effector torque about x-axis (τ_x) during PiH task

6.7 Conclusion

This chapter brought together the work of the previous chapters to apply the hybrid velocity/force control method to the PiH task on the compliant Baxter robot. A solution was developed by dividing the problem into discrete states and implementing the control as a state machine. These three states were: motion in free space, searching for the hole, and inserting the peg. Substates were defined when necessary and the transitional conditions of each state were defined. All physical limits were obtained experimentally.

The results validated the solution developed in this work and the final peg insertion was completed in approximately 37 seconds with a round peg/hole radial clearance of 1.5 mm. This is comparable to results observed in the literature. The images provided a visualisation of the state machine in action on the Baxter robot and how the control affected the end effector characteristics like pose and wrench. The graphs reflect the different states defined by the code and control. The wrench figures shown have high

noise but the hybrid control model sufficiently filters the noise out to successfully achieve peg insertion.

7.0 CONCLUSION AND FUTURE WORK

7.1 General Conclusions

This work was dedicated to developing a hybrid velocity/force control method to apply to the PiH task on the Baxter robot. Hybrid velocity/force control was chosen as the most suitable solution due to the poor positional accuracy of the SEA joints in the Baxter robot arms. Additionally, the goal pose of the hole was unknown to the robot. Implementing velocity control with force feedback allowed the robot to adapt to the uncertain environment much more realistically than if it used position control without force feedback. The results validated the hybrid control method and the state machine used to implement the control. The solution is scalable and can be transferred to other PiH tasks with minor modifications to the limit and PID values.

The general conclusions of this work are:

Chapter 1 introduced the motivations, objectives and structure of the thesis.

Chapter 2 introduced the background of fine robotic manipulation in the context of the PiH task. It explored the literature of impedance control and analysed why it would not be a suitable solution for this work. Further, it provided context on hybrid position/force and velocity/force control, and how these have been implemented in previous works. The chapter then presented the literature on the PiH task specifically and how others have designed solutions to address the problem such as defining distinct states, handling pose uncertainty and combining dual-arm manipulation. This chapter concluded with a discussion of the social and ethical implications of introducing collaborative robots like Baxter to the workplace.

Chapter 3 is an exploration of position control on the Baxter robot. Pose kinematics are introduced, along with the setup of the Baxter robot for this work. This chapter explained how inverse pose kinematics can be implemented on the Baxter robot using the SDK provided by Rethink Robotics in the ROS framework. A test is conducted to validate joint position control on the Baxter robot. The results confirmed that it would not be suitable for the objectives of this thesis.

Chapter 4 implemented joint velocity control of the Baxter robot. It introduced velocity kinematics: both forward and inverse, and how velocity kinematics relate to the Jacobian matrix. Further, this chapter explained how inverse velocity kinematics and the pseudoinverse Jacobian matrix are implemented in Python and ROS for this work. To improve the velocity control and account for pose uncertainty a PID controller was developed. This chapter concluded by developing an approach to test velocity control on the robot, both with and without the PID controller. The results demonstrated that the PID increased the control by allowing parameter tuning and reducing pose error.

Chapter 5 built upon the PID velocity controller to develop a hybrid velocity/force control method. It introduced the wrist-mounted force/torque sensor along with the ROS notation and implementation. Further, it developed a method for filtering the sensor noise to improve the reliability of the control algorithm and reduce force thresholds which is beneficial for the longevity of the mechanical robot components. The chapter developed an approach to test the hybrid control and validated its suitability for the objectives of this work.

Chapter 6 validated the hybrid velocity/force control on the Peg-in-Hole task using a round peg/hole pair. It explained the three main phases that the task was divided into, as well as the subphases for the peg insertion phase. Namely these were: 1) Motion in free space; 2) Search, and; 3) Insert. Within the insert stage the robot needed to insert the peg; confirm it was inserted; and remove the peg after insertion. An approach was introduced which combined the three phases and subphases into a single cohesive state machine. The state machine successfully implemented hybrid control on the Baxter

robot, achieving peg insertion using a peg/hole pair with a radial clearance of 1.5 mm in approximately 37 seconds, comparable to results examined in the literature review.

Chapter 7 is a general conclusion of this work, providing a reflection and guidance of future work related to this thesis.

7.2 Reflection

At the onset of this work I naively underestimated its complexity; programming a robot to insert a peg into a hole similar to the way a child does sounded deceivingly simple. The learning curve to achieve the final objectives was steep and I have learnt a lot throughout this project. While not discussed in this work, the challenge of learning ROS and Python were significant undertakings because I had very limited experience with these tools prior to this work. Upon reflection, I should have achieved more in the first semester of this work and laid out a clearer plan. Reading through my notes at the time I can see that my learning was relatively unstructured. However, after completing a Python programming course during the semester break I made up for this in the second semester through making mistakes quickly and learning from them. I created a plan for developing a solution, which has been laid out in this work in the form of the different chapters.

Overall, I am very pleased with the progress made and thankful for the opportunity to learn so much. The process of teaching myself new concepts on programming, robotics, controls and kinematics has been extremely rewarding. I am confident that the solution I developed in this thesis can be scaled for broader application both on the Baxter robot for other PiH tasks (*e.g.* square peg/hole pair) and other compliant robots.

There are limitations to the solution developed and it is not robust enough to handle all edge cases such as a moving hole target or a peg/hole pair with smaller clearances. The largest area for improvement of this work specifically is that it did not implement

dual-arm control, mainly due to time constraints. Expanding the solution proposed to include dual-arm control could lead to significantly improved results in terms of speed, repeatability and robustness. Such cases can be researched further and future works are presented in the next section to address some of the limitations of this model.

7.3 Future Work

This work focused on developing a hybrid velocity/force control method for fine manipulation on the Baxter robot. It then validated this control method experimentally and demonstrated successful results. The applications of this control method are numerous and the possible avenues of exploration are endless.

With regard to the Baxter robot specifically, this solution can be built upon to address more edge cases such as a moving hole target if, for example, the hole was on a slow moving assembly line. Further, this solution relies on a human to place the end effector approximately above and normal to the hole surface. This could be improved upon by developing a solution which adjusts the end effector so that it is always normal to the hole surface by optimising the normal force. Computer vision through a Kinect sensor or camera can be integrated with the hybrid controller to identify the hole and eliminate the need for a human to move the end effector above the hole. Better methods of reducing the FTS noise would also improve the reliability and control. Additionally, expanding this solution to handle different geometric peg/hole pairs with smaller clearances would require further research and development. Finally, combining hybrid control with dual-arm function of the Baxter robot is an excellent area of research that can increase the scope of manipulation tasks the robot is capable of.

In terms of the hybrid control method, it can be applied more broadly in many industries to improve automation capabilities. For example, autonomous refuelling stations can use hybrid control for inserting the end of a fuel nozzle into the car's fuel tank, or a car charger into the charging port. Another industrial use case is for mounting external panels on assembly lines, such as car panels.

8.0 REFERENCES

1. *Impedance Control*. 2 October 2021]; Available from: https://robotics-explained.com/impedancecontrol.
2. Hogan, N., *Impedance Control: An Approach to Manipulation: Part I—Theory.* Journal of dynamic systems, measurement, and control, 1985. **107**(1): p. 1-7.
3. Hogan, N., *Impedance Control: An Approach to Manipulation: Part III—Applications.* Journal of dynamic systems, measurement, and control, 1985. **107**(1): p. 17-24.
4. Hogan, N., *Impedance control of industrial robots.* Robotics and Computer-Integrated Manufacturing, 1984. **1**(1): p. 97-113.
5. Kelly, R., et al. *On adaptive impedance control of robot manipulators*. in *Proceedings, 1989 International Conference on Robotics and Automation*. 1989.
6. Zou, P., et al. *An Approach for Peg-in-Hole Assembling Based on Force Feedback Control*. in *2019 Chinese Automation Congress (CAC)*. 2019.
7. *UR5 Technical specifications*. Universal Robots; Available from: https://www.universal-robots.com/media/50588/ur5_en.pdf.
8. Chen, K.S., *Application of the ISO 9283 standard to test repeatability of the Baxter robot*, in *Electrical & Computer Engineering*. 2015, University of Illinois at Urbana-Champaign.
9. Daisuke Yamada, J.H., Tetsuro Yabuta, *Comparison Between Admittance and Impedance Control of a Multi-Finger-Arm Robot using the Guaranteed Manipulability Method.* Precision Instrument and Mechanology, 2013. **2**(1): p. 85-93.
10. Calanca, A., R. Muradore, and P. Fiorini, *A Review of Algorithms for Compliant Control of Stiff and Fixed-Compliance Robots.* IEEE/ASME Transactions on Mechatronics, 2016. **21**(2): p. 613-624.
11. Schindlbeck, C. and S. Haddadin. *Unified passivity-based Cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks*. in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015.

12. Keemink, A.Q.L., H. van der Kooij, and A.H.A. Stienen, *Admittance control for physical human–robot interaction.* The International Journal of Robotics Research, 2018. **37**(11): p. 1421-1444.
13. Ott, C., R. Mukherjee, and Y. Nakamura, *A Hybrid System Framework for Unified Impedance and Admittance Control.* Journal of intelligent & robotic systems, 2015. **78**(3): p. 359-375.
14. Ortenzi, V., et al., *Hybrid motion/force control: a review.* Advanced Robotics, 2017. **31**(19-20): p. 1102-1113.
15. Raibert, M.H. and J.J. Craig, *Hybrid Position/Force Control of Manipulators.* Journal of dynamic systems, measurement, and control, 1981. **103**(2): p. 126-133.
16. Villani, L. and J. De Schutter, *Force Control*, in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Editors. 2016, Springer International Publishing: Cham. p. 195-220.
17. Duchaine, V. and C.M. Gosselin. *General Model of Human-Robot Cooperation Using a Novel Velocity Based Variable Impedance Control*. in *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*. 2007.
18. Hou, Y. and M.T. Mason, *Robust Execution of Contact-Rich Motion Plans by Hybrid Force-Velocity Control*. 2019.
19. Zhang, X., et al., *Peg-in-Hole Assembly Based on Two-phase Scheme and F/T Sensor for Dual-arm Robot.* Sensors (Basel, Switzerland), 2017. **17**(9).
20. Li, Y., *Hybrid control approach to the peg-in hole problem.* IEEE Robotics & Automation Magazine, 1997. **4**(2): p. 52-60.
21. Jasim, I.F., P.W. Plapper, and H. Voos, *Position Identification in Force-Guided Robotic Peg-in-Hole Assembly Tasks.* Procedia CIRP, 2014. **23**: p. 217-222.
22. Huang, S., et al. *Realizing peg-and-hole alignment with one eye-in-hand high-speed camera*. in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. 2013.
23. Chang, R.J., C.Y. Lin, and P.S. Lin, *Visual-Based Automation of Peg-in-Hole Microassembly Process.* Journal of Manufacturing Science and Engineering, 2011. **133**(4).

24. Huang, Y., et al., *Vision-guided peg-in-hole assembly by Baxter robot.* Advances in Mechanical Engineering, 2017. **9**(12).
25. Dong, Q., R. Hu, and L. Zhang. *Robot Compliance Control for Peg-in-Hole Assembling Based on Contact Mechanics*. in *Signal and Information Processing, Networking and Computers*. 2020. Singapore: Springer Singapore.
26. Park, H., et al., *Compliance-Based Robotic Peg-in-Hole Assembly Strategy Without Force Feedback.* IEEE Transactions on Industrial Electronics, 2017. **64**(8): p. 6299-6309.
27. Wang, S., et al., *A Robotic Peg-in-Hole Assembly Strategy Based on Variable Compliance Center.* IEEE Access, 2019. **7**: p. 167534-167546.
28. Tsumugiwa, T., et al. *Switching control of position/torque control for human-robot cooperative task - human-robot cooperative carrying and peg-in-hole task*. in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*. 2003.
29. Makris, S., et al., *Dual arm robot in cooperation with humans for flexible assembly.* CIRP Annals, 2017. **66**(1): p. 13-16.
30. Leidner, D., et al., *Robotic agents representing, reasoning, and executing wiping tasks for daily household chores.* AUTONOMOUS AGENTS AND MULTI-AGENT SYSTEMS, 2016.
31. Mohan, V., et al., *Teaching a humanoid robot to draw 'Shapes'.* Autonomous Robots, 2011. **31**(1): p. 21-53.
32. Huang, Y., X. Chen, and X. Zhang. *Kinematic Calibration and Vision-Based Object Grasping for Baxter Robot*. in *Intelligent Robotics and Applications*. 2016. Cham: Springer International Publishing.
33. Shin, S.Y. and C. Kim, *Human-Like Motion Generation and Control for Humanoid's Dual Arm Object Manipulation.* IEEE Transactions on Industrial Electronics, 2015. **62**(4): p. 2265-2276.
34. Krüger, J., G. Schreck, and D. Surdilovic, *Dual arm robot for flexible and cooperative assembly.* CIRP Annals, 2011. **60**(1): p. 5-8.
35. Huang, Y., et al., *Peg-in-hole assembly based on master-slave coordination for a compliant dual-arm robot.* Assembly Automation, 2020. **40**(2): p. 189-198.
36. Wallace, J., *Getting collaborative robots to work: A study of ethics emerging during the implementation of cobots.* Paladyn, Journal of Behavioral Robotics, 2021. **12**(1): p. 299-309.

37. Anshu Saxena, A. and A. Amit, *The Race Between Cognitive and Artificial Intelligence: Examining Socio-Ethical Collaborative Robots Through Anthropomorphism and Xenocentrism in Human-Robot Interaction.* International Journal of Intelligent Information Technologies (IJIIT), 2020. **16**(1): p. 1-16.
38. II, R.L.W. *Baxter Humanoid Robot Kinematics*. 2017.
39. Smith, C.D.J., *Peg in hole robot manipulation using Robot Operating System*, in *Department of Mechanical Engineering*. 2021, Curtin University.
40. Jazar, R.N., *Theory of Applied Robotics : Kinematics, Dynamics, and Control / by Reza N. Jazar*, ed. SpringerLink. 2007, Boston, MA: Boston, MA : Springer US.
41. *IK Service Example*. 2015; Available from: https://sdk.rethinkrobotics.com/wiki/IK_Service_Example.
42. *Baxter Interface*. 2015; Available from: https://sdk.rethinkrobotics.com/wiki/Baxter_Interface.
43. *baxter_pykdl*. 2015; Available from: https://github.com/RethinkRobotics/baxter_pykdl.
44. *KDL wiki*. 2014; Available from: https://www.orocos.org/kdl.html.
45. *numpy.linalg.pinv*. 2021; Available from: https://numpy.org/doc/stable/reference/generated/numpy.linalg.pinv.html.
46. Lino, P.J.d.C., *The Control of Baxter Robot and its Interaction with Objects Using Force Sensitive AR10 Hands, Guided by Kinect*, in *Department of Electrictronic and Computer Engineering*. 2017, University of Coimbra: University of Coimbra.
47. *netft_utils*. 2017; Available from: http://wiki.ros.org/netft_utils.
48. Cao, F., et al., *Contact force and torque sensing for serial manipulator based on an adaptive Kalman filter with variable time period.* Robotics and Computer-Integrated Manufacturing, 2021. **72**: p. 102210.
49. Sun, J., Y. Shen, and J. Rosen, *Sensor Reduction, Estimation, and Control of an Upper-Limb Exoskeleton.* IEEE Robotics and Automation Letters, 2021. **6**(2): p. 1012-1019.

9.0 APPENDICES

9.1 Appendix A: Extended Abstract Permission Form

School of Civil and Mechanical Engineering

MXEN4002 Mechatronic Engineering Research Project 2
MXEN4004 Mechatronic Engineering Research Project 2A

Permission Sheet to make the Extended Abstract publically available

By signing below I acknowledge that my extended abstract: (i) is two (2) pages in length, (ii) has been checked and deemed satisfactory by my academic supervisor and (iii) may be made publically available on the School of Civil and Mechanical Engineering or other Curtin University websites or else made publically available in any other form.

Please note that if you submit your extended abstract without this permission sheet (either through Blackboard or hard copy) then your submission will be deemed to be incomplete and may result in you being awarded a Fail or Fail Incomplete grade. The only exception to this is if you have previously obtained **permission in writing from the unit coordinator** to have this requirement waived.

Student Name: Mats Niklasson

Student ID: 1913 4299

Project Title: Peg-in-Hole Manipulation of the Baxter Dual-Arm Robot.

Academic Supervisor: Dr. Lei Cui

Signature (Student) _____ Date 27/10/21

Signature (Supervisor) _____ Date 28/10/2021

9.2 Appendix B: Project Completion Form

Completion of a 4th Year Engineering Project (MERP II)

Student must attach this form with the final thesis/report

Student Name: MATS NIKLASSON

Student Number: 19134299

Project title: Peg-in-Hole Manipulation of the Baxter Dual-Arm Robot

4th Year Research Project has now been completed.

1. **All Chemicals have been returned.**
   Yes / No / Not applicable

2. **All Equipment has been cleaned and returned.**
   Yes / No/ Not applicable

3. **All Samples and products from the project have been disposed of and/ or dealt with in an appropriate manner.**

   Yes / No / Not applicable

Mechatronics Engineering Technical Staff Member

Name: Hamish

Signature:

Date: 21/10/21

9.3 Appendix C: List of Attached Files – USB

1. Thesis
2. Extended Abstract
3. MERP I Progress Report
4. ROS workspace, titled “mn” with all source code. The main PiH program is in the file titled “final_peg_insert.py” found at the file address within the workspace: “/mn/src/peg_insert/scripts/final_peg_insert.py”. The FTS is launched by first launching “ft.launch” and then “bias_ft.launch”.
5. Video of the Baxter robot completing the PiH task