Development of an assisted–teleoperation system for a dual–manipulator nuclear decommissioning robot

Craig West

MEng Mechatronic Engineering

A thesis submitted in partial fulfilment for the degree of Doctor of Philosophy at Lancaster University.

June 2019

Abstract

This thesis concerns a robotic platform that is being used for research into assisted tele–operation for common nuclear decommissioning tasks, such as remote handling and pipe cutting. The machine consists of dual, seven–function, hydraulically actuated HYDROLEK manipulators mounted (in prior research) on a mobile BROKK base unit. Whilst the original system was operated by remote control, the present thesis focusses on the development of a visual servoing system, in which the user selects the object of interest from an on–screen image, whilst the computer control system determines and implements via feedback control the required position and orientation of the manipulators.

Novel research contributions are made in three main areas: (i) the development of a detailed mechanistic model of the system; (ii) the development and preliminary testing in the laboratory of the new assisted–teleoperation user interface; and (iii) the development of improved control systems for joint angle set point tracking, and their systematic, quantitative comparison via simulation and experiment. The mechanistic model builds on previous work, while the main novelty in this thesis relates to the hydraulic component of the model, and the development and evaluation of a multi–

objective genetic algorithm framework to identify the unknown parameter values.

To improve on the joystick direct teleoperation currently used as standard in the nuclear industry, which is slow and requires extensive operator training, the proposed assisted–teleoperation makes use of a camera mounted on the robot. Focussing on pipe cutting as an example, the new system ensures that one manipulator automatically grasps the user–selected pipe, and appropriately positions the second for a cutting operation. Initial laboratory testing (using a plastic pipe) shows the efficacy of the approach for positioning the manipulators, and suggests that for both experienced and inexperienced users, the task is completed significantly faster than via tele-operation. Finally, classical industrial, fuzzy logic, and novel state dependent parameter approaches to control are developed and compared, with the aim being to determine a relatively simple controller that yields good performance for the hydraulic manipulators. An improved, more structured method of dealing with the dead–zone characteristics is developed and implemented, replacing the rather *ad hoc* approach that had been utilised in previous research for the same machine.

Acknowledgements

I would like to express my gratitude to my supervisors, Professor C J Taylor, Dr S Monk and Dr A Montazeri, for their continued invaluable support and encouragement throughout this project. I would also like to thank the technicians within the department for their help in maintaining and repairing the hardware necessary for this research. Finally I would like to thank my friends, particularly the ones made throughout the process of the PhD, for helping me to maintain a level of sanity.

Declaration

I declare that the work in this thesis has been done by myself and has not been submitted elsewhere for the award of any other degree.

Craig West

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The research behind this thesis is motivated by the demand for robotic solutions to the nuclear decommissioning challenge in the UK and beyond. The thesis makes contributions in three main areas: physically–based and data–driven modelling for hydraulically–actuated manipulators (as commonly utilised on nuclear sites); an investigation into 'low–level' feedback control of the joint angle for such manipulators; and, in regard to the 'higher–level' control problem, assisted teleoperation for common decommissioning tasks, such as remote handling and pipe cutting.

To bring focus, the algorithms in this study are designed and implemented in the laboratory for a dual–arm mobile robotic platform previously developed at Lancaster University [1–4]. This takes the form of a Brokk–40 demolition robot with caterpillar tracks, to which two seven–function HydroLek–7W robotic manipulators have been attached, as shown in Figure 1.1. The unit is electrically powered, with an on-board hydraulic pump to power the caterpillar tracks and, by means of hydraulic pistons, the manipulators.

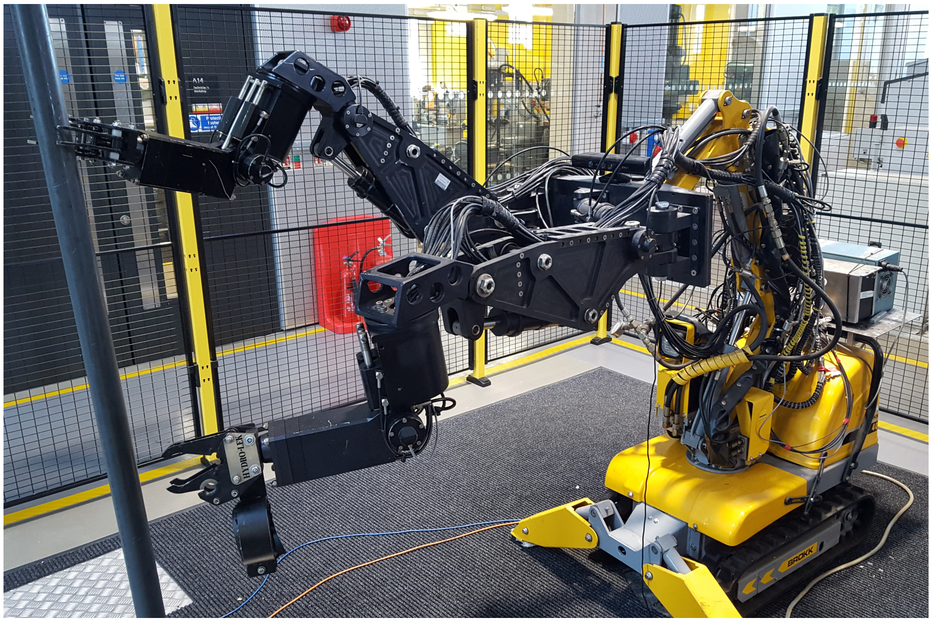


Figure 1.1: *Brokk–HydroLek platform used for experimental work.*

The platform has some similarity to the Hitachi system [5,6] which consists of two hydraulic manipulators on a tracked vehicle. However, the latter is teleoperated with a rather complex user interface that arises from the many cameras and sensors on the Hitachi, as well as from the teleoperated control system. By contrast, the assisted teleoperation system developed in this thesis is designed to keep the user engaged and in control at all times, whilst being as straightforward as possible to implement, and with minimal sensor requirements.

It should be noted that the research project behind this Engineering Department PhD studentship[1] was designed with a practical focus throughout, and was explicitly based on the Brokk–Hydrolek robotic platform, with all the experimental results

[1] Lancaster University funded PhD studentship, awarded in the context of external industry funded bursaries. Hence, the author is also grateful to the Nuclear Decommissioning Authority and the National Nuclear Laboratory, for their support of research into this robotic platform.

conducted in a (non–radioactive) laboratory setting. In this regard, the present author was the lead researcher for operation of the system during the project, responsible for its operation in a wide range of experiments, including e.g. calibration, sourcing replacement parts and National Instruments (NI) Labview software coding.

1.1 Research Context

Robotics for the nuclear industry is a large and very active area of research, recently attracting a lot of attention internationally due to the increasing demands for nuclear decommissioning. Many nuclear power stations around the world are approaching the end of their working life, or being shut down early due to policy change or safety concerns, such as in Japan following the Fukushima disaster in 2011.

The UK has a long track record of research in robotics for the nuclear industry, with recent projects such as the RoMaNs (Robotic Manipulator for Nuclear Sort and Segregation) funded as part of an EU Horizon 2020 project, and led by the University of Birmingham. Similarly, at the time of writing, two new research Hubs have been announced for nuclear robotics, funded through the Engineering and Physical Sciences Research Council (EPSRC) and industrial partners. The first, a National Centre for Nuclear Robotics, led by the university of Birmingham and consisting of eight UK Universities, aims to bring together expertise from across the UK and globally for research into robots for extreme environments. The second is the Robotics and Artificial Intelligence for Nuclear, RAIN, led by Manchester University and consisting of seven UK universities and the UKAEA's (UK Atomic Energy Authority) RACE

(Remote Applications in Challenging Environments) centre, as well as international partners from Japan, Italy and the US. Lancaster University is involved in both of these Hubs. There are also numerous other projects being carried out at different Universities both independently and working with industry partners, such as the National Nuclear Laboratory (NNL).

It is clear that robotics for the nuclear industry is a key research area that will be growing over the next few years. Therefore motivation for the present thesis is to contribute research towards the development of modular and general purpose robotic systems that can be used in different situations for different tasks. In this regard, the Brokk–Hydrolek platform is an ideal test bed for demonstrating new solutions, since it is a mobile platform capable of using two hydraulic arms cooperatively or independently to perform a range of tasks.

Chapter 2 of the thesis discusses the nuclear decommissioning challenge in more detail, and addresses the use of robots in the nuclear industry. This is followed in Chapter 3 with information about the Brokk–Hydrolek system, and past research into the control of this machine. Hence, for brevity, these discussions and citations to the literature are not repeated here. However, it is clear that a major aim in the sector is to make nuclear robotic systems more flexible, such that they can carry out a variety of tasks in different locations and situations. This would make decommissioning cheaper, easier and safer, as it would reduce the number of people having to enter high risk areas, and reduce the time taken to complete decommissioning tasks.

1.2 Research Objectives

The overall objectives of the research are summarised as follows:

1. To develop a comprehensive mechanistic model of the robotic platform. In contrast to earlier control model orientated research using the same machine [7,8], this new nonlinear, simulation model aims to capture the complex mechanistic interactions between various mechanical, hydraulic and electrical components of the Brokk-Hydrolek. Some preliminary results in relation to modelling the mechanical components are discussed in reference [9]. However, important changes and additions to the model are reported in this thesis, including an entirely new hydraulic sub system.

2. To estimate the unknown parameters of the mechanistic model, through the use of evolutionary algorithms, ensuring that realistic parameter values are utilised and that the model output matches the experimental data. A multi-objective genetic algorithm approach is selected for this research because of the nonlinear and non-convex nature of the problem. Note that the prior work alluded to above [9, 10] did not consider the parameter estimation problem.

3. To develop an assisted teleoperation interface to replace the current industry standard joystick operation, with the aim of reducing operator workload and improving task performance. The case study selected for this research is remote pipe cutting, a very common task in the nuclear sector. Since research into e.g. laser cutting is beyond the scope of the present project, the focus here is

the control system and user–interface for *positioning* the manipulators, and the simulation of cutting using a plastic pipe in a laboratory setting.

4. To develop, implement and systematically evaluate various feedback control algorithms. Here, the aim is to optimise set point tracking of the nonlinear hydraulically actuated manipulator joints. As discussed in more detail in section 1.3 below, research into Objective 4 was in part motivated by preliminary testing of the assisted teleoperation system, which highlighted the need for improved control. This research aims to build on references [7, 8], in which the new contributions here include the design of fuzzy logic control systems, the use of an Inverse Dead–Zone (IDZ) control element, and the implementation of the control systems to both manipulators (references [7, 8] considered resolved motion for a maximum of three joints of one manipulator).

1.3 Thesis Structure

This introductory Chapter 1 provides a brief overview of the project. A detailed literature review covering nuclear decommissioning and robotics is provided in Chapter 2, whilst Chapter 3 gives details on the robotic platform and discusses previous work using the system. On the basis of such previous work, it is clear that simple linear PID algorithms are not sufficient to accurately control the non-linear hydraulic manipulators, so more advanced control systems are needed. Furthermore, within the nuclear industry, solutions have to be proven to work via simulation and subsequently by laboratory demonstration before even being considered for implementation. These two factors provide strong motivation for development of a detailed model that accurately represents all aspects of the system, which can be used for simulation work and advanced controller development.

Hence, Chapter 4 of the thesis focusses on the development of a comprehensive mechanistic model that includes all the hydraulic, electric and mechanical aspects of the system. The model aims to capture all the physical aspects of the manipulators and not just the kinematics as used previously. In general terms, the model can be used for advanced controller design, as the nonlinearities of the hydraulic system and mechanical interactions are captured, and also allows for simulation studies of how the manipulators move for specific inputs. Finally, the model will provide a graphical representation of the manipulators based on CAD models.

In parallel to this, a novel user interface is developed, utilising assisted teleoperation techniques. This is a design level, falling between full autonomy and teleoperation, in which a human operator gives high level commands such as selecting target objects, but the low-level control aspects are handled automatically. For the demonstrator system proposed in this thesis, which initially focuses on pipe cutting as an example of the generic concept, the operator selects a target pipe and cut location from an on–screen image, and the system calculates the joint angles necessary to position the manipulators at these targets, and subsequently moves them to grasp and cut the pipe. The development and testing of the assisted teleoperation interface is discussed in Chapter 5.

Initial testing of the new user interface identified a need for improved control to allow for successful completion of the low-level tasks. The interface creates target set points that an existing industrial controller couldn't reliably or consistently reach in a usable manner; for example, there may be considerable oscillation around the set point. Such experiments also identified that a big challenge is the dead-zone that exists in the hydraulic actuators, meaning that small inputs did not cause any movement. Considering these two issues, and because the mechanistic model was still under development, other controller options were investigated.

In this regard, Chapter 6 focusses on identifying the dead-zone region through a series of open-loop experiments. From these open–loop data, a State-Dependent Parameter (SDP) model was estimated and used to model each joint separately. One aspect of the new SDP controller, in contrast to earlier work [7, 8], is the use of an IDZ control element, implemented to deal with the dead-zone in a systematic way and to allow the use of underlying linear non-minimal state space control design methods [11]. This process provides a systematic procedure for tuning the parameters of the controller in the event of any changes to the system, for example oil pressure changes or future changes to the hardware. Fuzzy logic controllers are also considered, with these novel controller developments, discussed in Chapters 6 and 7.

Whilst Chapter 6 considers the simulation model response of individual joints, Chapter 7 evaluates the control systems on the hardware platform, for individual joints and for resolved motion of the manipulators in combination with the new assisted teleoperation interface from Chapter 5. This research aims to identify the best controller for the pipe cutting case study application example. Finally, Chapter 8 presents the overall conclusions of the thesis and makes recommendations for future work.

1.4 Publications Arising

The following peer reviewed international conference and journal articles have arisen from the research presented in this thesis:

- **A genetic algorithm approach for parameter optimization of a 7–DOF robotic manipulator**, West, C., Montazeri, A., Monk, S. D. & Taylor, C. J., *8th IFAC Conference on Manufacturing Modelling, Management and Control*, Troyes, France, June 2016. Appears in: IFAC–PapersOnLine. 49, 12, pp. 1261–1266 DOI: 10.1016/j.ifacol.2016.07.688. Conference article discussing preliminary parameter estimation results for the mechanistic model from Chapter 4, using a standard (not multi-objective) genetic algorithm.

- **Dynamic modeling and parameter estimation of a hydraulic robot manipulator using a multi–objective genetic algorithm**, Montazeri, A., West, C., Monk, S. D. & Taylor, C. J., *International Journal of Control*, **90**, 4, pp. 661–683, 2017. DOI: 10.1080/00207179.2016.1230231. Journal article summarising key results from Chapter 4, including development of the mechanistic model and the multi-objective genetic algorithm for parameter estimation.

- **A new approach to improve the parameter estimation accuracy in robotic manipulators using a multi–objective output error identification technique**, West, C., Montazeri, A., Monk, S. D., Duda, D. & Taylor, C. J., *26th IEEE International Symposium on Robot and Human Interactive Communication*, Lisbon, Portugal, August 2017. Appears in: IEEE Xplore, DOI: 10.1109/ROMAN.2017.8172488. Conference article, focusing on the novel multi–objective genetic algorithm from Chapter 4, but also including preliminary discussion of the assisted teleoperation system from Chapter 5.

- **State–dependent parameter model identification for inverse dead–zone control of a hydraulic manipulator**, West, C., Wilson, E.D., Clairon, Q. Montazeri, A., Monk, S. D. & Taylor, C. J., *18th IFAC Symposium on System Identification* (SYSID–2018), Stockholm, July 2018. Appears in: IFAC–PapersOnLine, Vol. 51, No. 15,2018, p. 126-131. DOI: 10.1016/j.ifacol.2018.09.102. Conference article focusing on system identification for SDP models in relation to Chapter 6.

- **A vision–based positioning system with the inverse dead–zone control for dual–hydraulic manipulators**, West, C., Montazeri, A., Monk, S. D. & Taylor, C. J. *2018 UKACC 12th International Conference on Control*, Sheffield, September 2018. Appears in: IEEE Xplore. DOI: 10.1109/CONTROL.2018.8516734. Conference article with a brief summary of the new assisted teleoperation system from Chapter 5 and SDP control of joint angle using an IDZ control element, based on selected results from Chapters 6 and 7.

- **Towards a Cooperative Robotic System for Autonomous Pipe Cutting in Nuclear Decommissioning**, Burrell, T., West, C., Monk, S. D., Montazeri, A. & Taylor, C. J. *2019 UKACC 12th International Conference on Control*, Sheffield, September 2018. Appears in: IEEE Xplore. DOI: 10.1109/CONTROL.2018.8516841. Conference article, first author is T Burrell, another PhD student at Lancaster, looking at building on the assisted teleoperation system in Chapter 5 and adding a mobile camera system to improve the vision input.

As well as the above full articles, there were poster presentations at 'Intelligent Robotic and Autonomous Systems' (IRAS; Lancaster, 2016) and at the N8 Robotics Group Meeting (Sheffield, 2017). Finally, the present author gave talks at the '24th International Conference on Systems Engineering' (ICSE; Coventry, 2015) and the 'Sellafield Graduate Nuclear Conference' (2016), with the former involving publication of an extended abstract focusing on the hardware platform (Research and teaching using a hydraulically–actuated nuclear decommissioning robot, West, C., Burrell, T., Montazeri, A., Monk, S. & Taylor, C.J.).

Chapter 2

Literature Review

The present chapter provides background information and motivation for the research, and discusses some of the technical challenges arising, via a necessarily selective literature review. The nuclear decommissioning context and the current state–of–the art for robotics in nuclear is discussed in sections 2.1 and 2.2. This is followed by consideration of some common robotics research themes and examples of existing platforms in sections 2.3 and 2.4 respectively.

Firstly it is important to consider what a robot is. The term robot is widely used in the media and this often blurs what a robot truly is. Although no universal definition of a robot exists it is generally agreed that a robot is a physical system that follows the sense, think, act paradigm [12,13]. The thinking, or computation, element is key to being a robot as this allows autonomous behaviour based on sensor inputs. The level of autonomy can vary however, from being fully autonomous and utilising artificial intelligence to react to changing scenarios, to something simpler like collision avoidance to prevent an otherwise teleoperated system colliding with the world.

2.1 Nuclear decommissioning

Within the next 10 years nearly all the existing nuclear power stations in the UK will reach the end of their life and will enter a shut down and decommissioning phase [14]. The decommissioning process involves characterisation, dismantling of all buildings and equipment, and removal of all radioactive material, leaving the site in a state where it can be used for other purposes and waste is safely and securely stored.

The characterisation stage involves measuring, analysing and characterising the type and level of radiological contamination, both within the buildings and for the waste. This is a critical stage as it allows for the most appropriate decommissioning and dismantling strategy for buildings and equipment to be determined, and allows waste to be sorted by radioactivity levels, having a large impact on both safety and cost [15].

There are two widely used options when decommissioning a nuclear power station [16]. The first is Safe Enclosure (SAFSTOR, or safe storage), during which the plant is kept intact and placed in protective storage for a number of years to allow radiation levels to naturally decrease. The second is DECON (Decontamination) where all contaminated materials are removed, and the decommissioning begins as soon as practically possible. A third option, entombment, has been proposed, in which the site is enclosed in a large concrete structure to contain radiation and subsequently left in that state. To date, however, entombment has not been implemented as a decommissioning strategy.

SAFSTOR has the advantage that radiation levels are lower when the dismantling of the plant occurs, therefore the job is safer, easier and cheaper. However, it also means future generations are given the job of cleaning up the mess created by current generations, as the storage period is typically 50 years [16, 17]. DECON results in the site being made safe as soon as possible and the land can then be used for another purpose; however, it also means that highly contaminated waste needs sorting and disposing of. DECON is also a very expensive process. Due to the high radiation levels, workers can only spend very short periods of time carrying out the work, even when wearing protective suits [18, 19]. Furthermore, the protective suits become contaminated creating further waste that needs to be sorted, and a large number of workers are required to share the work. As a result, robots are now being used wherever possible, since this minimises the exposure of people to dangerous environments.

Using a combination of the two techniques is common [20]. Here, all nuclear fuel is removed from the site and cooling ponds are emptied and cleaned. Buildings that do not pose a radiation risk are then demolished, for example turbine housing buildings. The site is subsequently put into safestore, allowing radiation levels in areas such as the reactor buildings to naturally decay. This makes decommissioning and final site clearance much simpler. This approach has been used by Magnox, for example at the Berkel site in the UK, which entered Safestore in 2010 [21]. EDF used the same approach for three sites in France i.e. Chinon, Bugey and St Laurent [20].

As well as power stations other nuclear facilities such as fuel processing plants, mines and submarines also require decommissioning, and present similar problems to power stations. The UK's current policy is that nuclear facilities should be decommissioned as soon as reasonably practical, which is determined separately for each case [17], whilst the radioactive waste should be placed in long term storage in safe and secure facilities [22].

2.2 Robotics in the nuclear industry

As discussed above, high radiation levels limit the amount of time workers can spend in contaminated areas, and prevent access to some areas completely. As a result, robotic systems are used for tasks across the nuclear plant life cycle, from inspection of active facilities to characterisation of legacy facilities, to dismantling and sorting of waste during decommissioning. A recent article by Tsistsimpelis et al. [23] gives a detailed overview of ground based robots used in the nuclear industry over the last few decades. The paper highlights how over the years there has been considerable research into robotics for different aspects of the nuclear industry, but that most systems are not universal and are developed for unique requirements. There are now attempts to change this and make robotic systems more flexible, such that they can carry out a variety of tasks in different locations and situations. This would make decommissioning cheaper, easier and safer, as it would reduce the number of people having to enter high risk areas, and reduce the time taken to complete tasks.

Decommissioning tasks range from cutting up pipes and equipment, to moving and sorting waste, to demolishing walls and so on. Dismantling large pieces of equipment is a large part of decommissioning, and items are usually cut up into manageable sizes. A variety of cutting techniques exist, and are used within decommissioning, ranging from mechanical saws to lasers and plasma cutters, with the technique used dependent on several factors, including: material type and thickness, accessibility to the area, radiation levels of the area/item, and hazards such as dust [24]. With traditional bespoke systems there would, for example, be a separate system for plasma cutting, as distinct from a system based on a mechanical saw. However, the focus is now on developing more flexible systems, such as devices with interchangeable end effector tools, or manipulators that can grasp and use standard tools and equipment.

When previous generation nuclear facilities were being designed, their future decommissioning was rarely considered, and as a result accessibility can often be an issue. This can make tasks that should be simple very time consuming, since just getting equipment in place becomes difficult, and external support structures may be required to take the weight of necessary equipment. This fact, coupled with the limited time people can spend in radiated areas, can make even simple jobs very slow and expensive. This is one reason why teleoperated robots are commonly used. They can move equipment and perform tasks whilst a human operator is at a safe distance, reducing the radioactive dose received by workers. This is important due to the danger of exposure to high levels of radiation, and international regulations limiting exposure dose levels of workers [18,25]. Because of these regulations, decommissioning jobs require a lot more workers than would be needed if the radiation was not present, and this contributes to the high costs.

The use of robotic systems reduces the number of people who need to enter high risk areas, which improves safety on site, and it can also reduce the cost and the time scale of the project. It also provides opportunities that may not exist if using human workers. For example, a hydraulically actuated manipulator can carry heavier loads than a human, which may allow items to be left as larger pieces, hence reducing the cutting and processing time, and potentially reducing secondary waste from dust which would need to be dealt with. Decommissioning involves a lot of boring, repetitive, simple jobs, for which robots are ideally suited, whereas human workers may get fatigued and loose concentration.

The choice of robotic platform to use is a big decision driven by many factors. Some of these are obvious, such as what tasks need performing and accessibility to the area. Other factors are less obvious, such as factoring in the lifetime cost of the system. Any robot used in decommissioning will become contaminated and so become waste itself, hence the cost of storing the contaminated device needs factoring in to the decision. The cost is mainly driven by the size of the waste and level of contamination.

One of the biggest decisions to make when selecting a robotic platform to use, is whether to use a commercial off–the–shelf system or to design and build a bespoke system. These two approaches are discussed below and examples of both approaches given, further examples can be found in [23].

2.2.1 Commercial off–the–shelf systems

Off–the–shelf systems have often been the preferred solution as they are known to ‘work’, and in the nuclear industry in particular this is very important. Safety is an overriding concern with work in the nuclear industry, so knowing a system will work as expected and has many hours of successful operation ahead, with a long mean time between failure (MTF) is important when choosing a system. Off–the–shelf solutions will usually have readily available replacement parts available too, whereas this may not be the case with bespoke solutions. However, off–the–shelf solutions may not be suitable, or may not be optimal, for all tasks during decommissioning, hence bespoke systems are often used to meet specific operational requirements.

An example of a commonly used off–the–shelf system is the Brokk series of remote control demolition machines [26]. They are designed to be rugged enough for demolition work, and are small enough to work inside buildings. They are often electrically powered which helps with working inside buildings, as there are no exhaust fumes. They also have a wide range of different end–effector tools available off–the–shelf, ranging from shears, to claws to excavator buckets. The major disadvantage of the Brokk series is that the teleoperation is very basic, so the user needs to be able to see the machine to operate it, and it requires extensive training and experience to use effectively. This means that if the machine is used in a high radiation area, the operator usually also needs to be in that area, putting themselves at risk, or they need to be able to see the machine through special lead glass windows, limiting the area the system can operate in. Also, being an off–the–shelf system, the machines are not specifically designed for a nuclear decommissioning environment and they are not an optimum solution for some tasks.

2.2.2 Bespoke nuclear robotic systems

When a commercial off the shelf solution is not viable, a bespoke system is needed. When deciding on a bespoke solution, an important consideration is if the system will be single use or reusable. Reusable solutions tend to be highly engineered, radiation hardened and need to be designed for remote maintenance, for example by having accessible plug and play sensors or equipment, as human workers cannot necessarily enter the area to carry out maintenance. This makes them expensive, and potentially increases the time the design and testing stages take, as there needs to be confidence that the system will work. Alternatively, single use sacrificial solutions can be used; these are generally cheaper but only suitable for certain tasks, and typically have a short lifetime before they become damaged by radiation.

When making the choice between a single use or reusable system, several factors have to be taken into account, for example obvious matters such as what the task is and how long it is likely to take, but also less obvious aspects such as lifetime costs. As the system will be contaminated and become waste, it may actually end up cheaper to use one expensive reusable system than multiple cheap single use systems, which could cost more to sort and store in the future.

Previously, bespoke systems have been made for a very specific tasks in the nuclear sector. The downside is that the system may only be used for one task at one site because, for example, of differing plant architectures. An example of these highly task specific solutions is ‘Reactorsaurus’ [27], a large Reactor Dismantler Manipulator (RDM), designed and built to dismantle the reactor at the Dounreay site in Scotland. It was designed and built specifically for that site. The 75 ton machine will be lowered into the reactor, and using two large manipulator arms, cut up and remove all the equipment, such as pipework and carbon shield rods. The whole device is remotely operated from a central control room.

More recently, the approach has shifted towards making more flexible systems that are capable of adapting to carry out a particular task in different scenarios, or being able to carry out a larger range of tasks. Examples of this approach are described next. The Avexis project, involving Sellafield and the University of Manchester [28], concerns a range of small remotely operated underwater systems that are being developed for in situ inspection of fuel storage ponds [29], as well as for fuel characterisation at the Fukushima Daiichi plant [30]. Both systems are being designed for particular situations and locations, i.e. Sellafield storage ponds and Fukushima, but the technology would be applicable to other nuclear or hazardous environments.

OC Robotics, a Bristol based company, specialise in snake arm robots. These machines, with high degrees of flexibility and small size, have already been used for inspection and repair tasks within nuclear power plants on pipework in hard to access areas [27,31]. They are now working in collaboration with Sellafield and The Welding Institute, TWI, on adding a laser cutting head to allow a snake arm machine to be used for decommissioning tasks [32–34], such as cutting up pipework, with successful results so far. However this approach introduces new challenges such as controlling the laser beam, and controlling potential waste products from the reaction of the laser with different materials. OC Robotics have worked with nuclear sites such as Sellafield to develop snake arm devices that meet the needs of the industry and can perform tasks that other machines are not capable of due to accessibility issues.

The French energy company Areva have developed a small robot named CHARLI, for laser cutting pipework in the reactor vessel of the Superphénix fast neutron reactor in France [35, 36]. It is designed to remotely move around pipework and using the laser cutting head, mounted on the end of a robotic arm, cut sections of pipework to allow for the removal of sodium from the reactor vessel. Although designed for one particular plant, this system would be applicable to anywhere with pipe work of a similar size.

Another French company, Cybernetix, has developed a range of remotely operated robots for nuclear decommissioning and inspection tasks [37]. Some are based on a Brokk base unit with the added functionality of a manipulator arm. The manipulator arms can carry a variety of tools for performing different tasks. They also have manipulator arms mounted to devices that are lowered into work areas on a hoist, similar in principle to the large device used at the Dounreay site in Scotland.

The Sarcos Gaurdian GT [38, 39] is a system with dual hydraulic manipulators, that is directly teleoperated through a custom interface allowing highly dexterous control. It is designed so that the manipulators have kinematic equivalency with a human arm, so user movements are accurately captured and recreated. This makes control intuitive and easy to learn. The platform is in development, and is custom made to order, so as to meet differing customers specifications. It has been demonstrated to have the dexterity to turn valves and press buttons in a disaster response scenario, so will likely be used in nuclear environments in the future where a direct teleoperation solution is desired.

Many of the mobile robots currently being used in decommissioning have only a single manipulator, such as those by Cybernetix above, and the Kuka manipulators now commonly used at the Sellafield site. Utilising two arms working together would potentially allow more flexibility and improved functionality, as for the Hitachi system mentioned in Chapter 1 [5], but does add extra challenges such as cooperative control and collision avoidance to prevent the arms hitting each other.

2.3 Common themes in robotics

Robotics is a huge area of research, in both industry and academia, and has the potential to enter almost every industrial sector and aspect of life, from the military to healthcare to manufacturing. Each application and industry presents its own unique challenges, but there are several major themes that affect all sectors. For example, automation, vision and environmental sensing, and human robot interactions, which are all discussed below. These three areas are chosen for this highly selective review, since they impact on most industries and are particularly relevant to the research in the thesis.

2.3.1 Automation

Automation is important in the development of all fields of robotics. Increasing the number of tasks that can be reliably automated reduces the required input from a user, which has different benefits for different scenarios. In a military or security context, for example, it would allow robots to be used alongside people, carrying equipment say, without requiring someone to be constantly controlling the robot, potentially improving safety whilst reducing costs.

Production line robotic manipulators have been automated for years, but traditionally operate in cages or away from human workers for safety and perform the same simple repetitive task over and over, all strictly controlled (e.g. dealing with identical parts in an identical manner). There is no ‘intelligence’ in these robots. The focus is now shifting to robots that work alongside human workers on a production line such as robots by Epson [40], ABB [41] and the Baxter by Rethink Robotics [42], that can adapt to the work environment and do not need parts to be in the exact same position every time i.e. they can sense and adapt. A big challenge with this approach, is that the system needs to be able to sense what is going on around it and react appropriately e.g. to be able to operate safely alongside humans, such robots need to stop if a potential collision is detected.

Full automation may not always be realistic or even desirable, however certain aspects often can be automated. Semi-automation reduces work load on the operator and can improve safety as well as efficiency. Montague [43], for example, reports on a semi-autonomous pipe repairing robot, which traverses the inside of the pipe and repairs it from the inside, improving efficiency over the traditional method of having to dig up the road to access the pipe. Another example is described by Shaban et al. [44], in which a vibrolance system, used to compact surrounding soil as it penetrates into the ground on a construction site, is semi-automated so that the lance is automatically kept vertical. The system, which built on earlier research by Dixon et al. [45], was shown to increase operation speed by a factor of three.

A situation where full automation is not desirable, is for search and rescue operations. For example, when looking for trapped people in a collapsed building, relying purely on a robot could lead to survivors being missed, hence keeping a human in the loop seems desirable. Marques et al. [46] developed a search and rescue robot that can semi-autonomously traverse uneven terrain, whilst a human operator monitors the system, controls direction and watches the various live video feeds. This keeps the human as the one spotting people or objects of interest, rather than relying entirely on object recognition algorithms or similar technology, which may not be reliable in the unstructured environment in which a search and rescue robot operates. Mughal et al. [47] detail another example of a robot with semi-autonomous navigation ability. This robot can be directly remote controlled or set to autonomously follow a light source, is designed to be used in hazardous environments, and is equipped with sensors to detect different gasses.

2.3.2 Vision and environment sensing

Vision and environment sensing is vital in robotic systems, whether teleoperated or fully autonomous. With teleoperated systems the vision could be as simple as a single camera to give the operator a view of the work space, or consist of multiple cameras giving different views, as used in Marturi et al. [48]. Also the equipment used could range from simple cameras giving just a video feed, to 3D systems such as LiDAR (Light detection and ranging) which give a 3D representation of the environment and can be integrated with CAD models for high quality user interfaces. An example is described by Leeper et al. [49], where 3D point cloud data are combined with a CAD model of the robotic platform.

Automated systems typically make use of multiple different sensors, for example combining a simple or stereo camera system with LiDAR provides extensive information about the environment so that autonomous algorithms have enough data to work with. Wei et al. [50] utilise multiple sensors including cameras, LiDAR and radars in combination, to develop an autonomous driving research vehicle.

The choice of which and how many sensors is dependent on the task and work environment; a factory production line manipulator would need different information to a mobile platform that navigates unknown or changing environments. However, having too many or redundant sensors could slow down computational time and actually lower performance, so getting the balance right is important.

To illustrate, Shauri et al. [51] consider a dual arm robot that autonomously screws a nut and bolt together, with multiple cameras and force sensors. Their algorithm switches between different sensors depending on which stage of the task the robot is at. Starting with a camera that gives an overview of the workspace to move the arms to the general area, then switching to cameras mounted on the end effectors to give fine position control near the target object. This approach limits the information being used to just that which is relevant, improving performance.

2.3.3 Human robot interaction

Human robot interaction is important, both in terms of controlling the robots and working alongside them. Traditional teleoperation is performed using a bespoke input device, mouse and keyboard, or more commonly with a joystick. This often requires extensive training and practice before a robotic manipulator could be controlled in a useful and efficient manner. A big development was using master slave systems, where the operator moved a joint on a model in front of them and that translated into moving the corresponding joint on the actual robot. The problem with these methods is they are relatively slow, hard to learn and sometimes require unnatural movements from the user [52].

In fact, master slave systems at Sellafield have changed little since the 1960s. The operator moves the master, which is mechanically linked to the slave through thick concrete walls, to control the slave manipulator in the hazardous area. This limits the operators view of the work environment, and relies on experience of the operator for accurate useful motion. These systems require highly skilled and experienced operators, and these operators are now ageing (mean age over 50 in the UK) [48]. Newly trained operators are required, or the dated system needs replacing with a more efficient solution.

Recent developments in technology have allowed new ways of interacting with industrial robots in a more natural and intuitive way. For example, using accelerometer based input devices to track user movement and control end effector position [53]; gesture inputs, including accelerometer based [54,55], camera motion tracking [52,56,57], and direct motion tracking; using either markers attached to the human arm, or equivalent markerless systems [52, 57]. Results of using gesture input or motion tracking, in whichever form, have shown to be positive, with a smaller learning curve than traditional systems. Neto et al. [53] use a Nintendo Wii games console remote to provide gesture input using the accelerometers within the remote. The results show that after a training period, during which the system learns the particular users movements, even non-expert operators can control the manipulator successfully for a variety of tasks.

Making robots safe to work alongside human workers is essential. Work such as that by Loughlin et al. [58] has looked at ways of making robot manipulators safe to work around humans, by detecting possible collisions and acting to prevent them. The manufacturing industry has traditionally used industrial robots on production lines, and human workers had to be kept separate for safety reasons. New robots can work side by side with humans safely and cooperatively. Examples of this include the robots made by Epson [40], ABB [41], and Baxter by Rethink robotics [42], which can safely work alongside humans on a production line in an automated manner.

2.4 Common types of robotic platform

Many different robotic platforms exist and are used for different tasks, from industrial manipulators to autonomous vehicles. In the following subsections, some of the most well-known and widely used robotic platforms are introduced.

2.4.1 Biped/Humanoid

Biped robots have two legs and are usually designed to look as human as possible. The challenge is making them walk! Walking on two legs is incredibly difficult from a robotics perspective. As you, a human being, take a step, there are constant adjustments to your balance. Walking on uneven terrain is even more challenging, since you have to react to movement and slippage when placing your foot, adjust the force as you put your foot down, and adjust your balance to remain stable. Humans do all this almost subconsciously, programming a robot for this is a major challenge.

Humanoid robots have potential advantages in that they could enter areas humans can enter and work in the same way a human would. In a nuclear decommissioning context, this could mean using standard tools, opening doors, climbing stairs and opening and closing valves. This would also make them valuable in disaster response scenarios.

Probably the most well-known biped robot is ASIMO (Advanced Study In Motion) made by Honda [59], which is able to walk and even climb stairs. Although currently there are no practical uses for ASIMO, it is a good demonstration of what is possible. Honda have used the technology developed with ASIMO to build a humanoid disaster relief robot, the E2-DR [60]. Currently at prototype stage, the E2-DR is capable of walking on 2 legs, crawling on 4 and climbing ladders.

Another well-known bipedal robot is ATLAS [61–63], based on the Petman platform [64], with both developed by Boston Dynamics. At the time of writing, the device has been making news headlines for its ability to jump and perform backflips. ATLAS has been designed to be as human like as possible, and to be able to navigate on uneven terrain. It is a lot more agile than other humanoid robots, and is able to quickly adjust its balance, hence it moves in a more human like manner than ASIMO. Robots such as ATLAS could in the future be used in a variety of roles, from working in people's houses, to entering dangerous areas where people cannot enter to perform tasks. By being as human like as possible, they can work in a similar way to people and hence use the same tools as a human would, making them very flexible.

The development of biped robots has been accelerated by the DARPA (Defense Advanced Research Projects Agency) robotics challenge. The final of the challenge had teams from around the world competing to finish a series of tasks that ranged from opening doors, using hand tools to cut a hole, and turning a valve.

2.4.2 Quadruped and multi-legged robots

Another large category of robots is quadrupeds that walk on four legs. These are usually based on the way animals move, for example the MIT Cheetah robot [65, 66] and Boston Dynamics Big Dog [67–69]. stems.

The Boston Dynamics Big Dog was funded by DARPA, a part of the US government, and is intended for military applications. Unlike the Cheetah , it is not designed for speed, rather it is designed to carry heavy equipment anywhere a soldier might go, so has to be able to walk over rough terrain. It has been shown that it can walk over ice, correcting itself when it slips, and can regain balance after being pushed. Using legs instead of wheels or tracks allows it to go to places that it would not otherwise be able to go, but introduces a lot of difficult challenges in controlling the legs, as discussed by Li et al. [70].

Research is being carried out by the University of Manchester on hexapod robots for nuclear decommissioning tasks, supported by Forth Engineering Ltd and Sellafield. Two systems are under development, Latro and Corin [71]. These platforms are intended to navigate over unknown and rough terrain, to carry out decommissioning or inspection tasks. However, detailed information about these systems is currently very limited.

Another example of a legged robot being developed for nuclear environments, although aimed primarily at inspection rather than decommissioning, is Robug-II proposed by Luk et al. [72, 73]. Here, a quadruped system is developed that has the ability to walk on vertical surfaces using suction pads on the legs and body of the system. Legs are used to allow traversal over uneven and unknown surfaces, where wheeled systems would not be able to travel.

2.4.3 Snake like robots

Snake like robots are becoming increasingly popular. Carnegie Mellon University has developed a series of modular snake robots that are very versatile and could be used for a variety of tasks [74, 75]. OC robotics [76, 77] also develop snake like robots, which have been used in a variety of applications from assembly of aircraft, to inspection of nuclear power plants. Snake arm robots can access hard to reach areas and work in confined spaces where more traditional robots cannot go. OC robotics recently provided snake arm robots with high pressure water jets mounted on the end for inspection and cleaning tasks of a tunnel boring machine in Miami [78]. The hazardous environments that tunnel boring machines operate in, makes inspection by humans dangerous and time consuming. Hence, by using a robot that is flexible enough to navigate within the large machine and provide video footage back to an operator, maintenance becomes much quicker and safer.

2.4.4 Industrial manipulators

This field dominated research for many years [79]. Industrial robots have been in use since the 1960s, most commonly on car production lines. In the early days, these manipulators were quite simple, and used to automate tasks such as spot welding on car production lines. Here, the robots just did the exact same task over and over, and operated away from people. These robots were hard to re–program and not very flexible. Research and new technology has changed industrial manipulators, and they are now more accurate, more flexible, reprogrammable and much safer.

Modern vision systems, for example, allow industrial manipulators to locate and pick up items without requiring the item to be in a pre-determined position. They also allow for robots to be used in new industrial areas, such as product inspection. Xie et al. [80], for example, carry out tests using a manipulator with a camera mounted on the end-effector, or hand, of the manipulator. The manipulator can identify car wheels on a production line and move itself to inspect certain elements of the wheel, detecting any faults with the wheel.

Other industrial robots are moving away from the traditional idea of a single arm operating away from people. Either by having two arms work in a coordinated manner, providing the ability to perform more tasks and to work in more of a humanoid way, or by being designed to safely work alongside humans.

2.4.5 Dual manipulator systems

As noted above, there is currently a lot of work on human-like [81, 82] and industrial dual manipulator robots [1, 40–42, 58], usually for use in factories. Much research focuses on co-ordinated control of two robotic manipulators [83–85]. Challenges of dual manipulator systems include making the arms work together in a coordinated manner to achieve a task, to work independently of each other, and to prevent the arms from colliding with each other and any objects around them.

In industry, dual arm robots are beginning to be used in place of traditional manual production line or for packing type jobs. These robots can operate the arms independently or in a coordinated manner. They are currently designed to be used together with human workers, to help with simple repetitive assembly type jobs, usually involving small parts. When working with small parts, the robots can often be faster than a human, and do not suffer from fatigue or repetitive strain injuries from doing the same simple task repeatedly. This shift from traditional industrial robots which are operated away from humans, to robots that work alongside humans, introduces a new design challenges. For example, making the robots act as close to a human as possible so they can operate the same tools or work in the same environment as humans [86].

2.5 Concluding Remarks

The present chapter has reviewed selected examples of robotics research from the literature, with a focus on illustrative prototypes and commercial systems that are being used in a nuclear context. Three key technical areas have been considered in quite general terms, namely automation, vision and the human–robot interface. These areas are of particular relevance to the present thesis, which concerns assisted teleoperation of a mobile dual-manipulator system.

The key findings of the literature review are that the nuclear industry does have a desire to increase the use of robotic systems, as this will help improve safety and productivity whilst reducing costs. Traditionally many systems used by the nuclear industry have been bespoke, designed for one job at one site, however there is a drive to create more flexible systems that can perform multiple tasks in different locations. Although full autonomy is being strived for in many areas, for safety reasons it is not desirable within the nuclear industry, but direct teleoperation is highly skilled and slow, so a robotic solution in–between the two would be desirable. For any mobile robotic platform vision and sensing are important, within the context of nuclear decommissioning however the effect of radiation on electronic components has to be taken into account. Also the environment can not be entered to place beacons or sensors to aid in tracking, so sensors need to be on–board the mobile system. The sensors used and the user interface are closely linked, as the sensors determine what data can be presented to the user, which in turn effects how the user interacts with the system. Methods of interacting with for example robotic manipulators is a very active research area with many approaches such as master slave and motion tracking systems being researched.

The findings of this literature review motivate the development of the assisted teleoperation system for a dual-manipulator platform developed within this thesis. Note that specific technical results and algorithms utilised in the thesis (e.g. for parameter estimation and control system design) will be introduced where relevant, with additional citations, in Chapters 4, 5 and 6. In the first instance, however, the following Chapter 3 describes the robotic hardware/software framework.

Chapter 3

Brokk–Hydrolek System

Lancaster University Engineering department has developed a Multi-Arm mobile Robot System for research into Nuclear Decommissioning (occasionally, in some earlier articles, referred to by the acronym MARS-ND) [1–4], consisting of a Brokk 40 base unit coupled with a pair of Hydrolek HLK-7W manipulators, as shown in Figure 1.1 (chapter 1). Detailed information about the hardware/software arrangements are available from two earlier PhD theses [2, 8], whilst Chapter 4 of the present thesis discusses key components of the system in the context of the mechanistic model development. Nonetheless, for completeness, a brief overview of the system is first presented in section 3.1 below, followed by a review of associated research in section 3.2. Finally, section 3.3 considers some essential foundation work completed by the author, including maintenance and calibration issues.

3.1 System Overview

At the time of writing and when the experimental work for this project was completed, the hardware on the Brokk–Hydrolek robot was largely the same as when it was first developed: see [1–3]. Figure 3.1 shows a diagram representing the overall system and showing the distinct subsystems, namely the operators computer, the Brokk base unit and the Hydrolek manipulators.

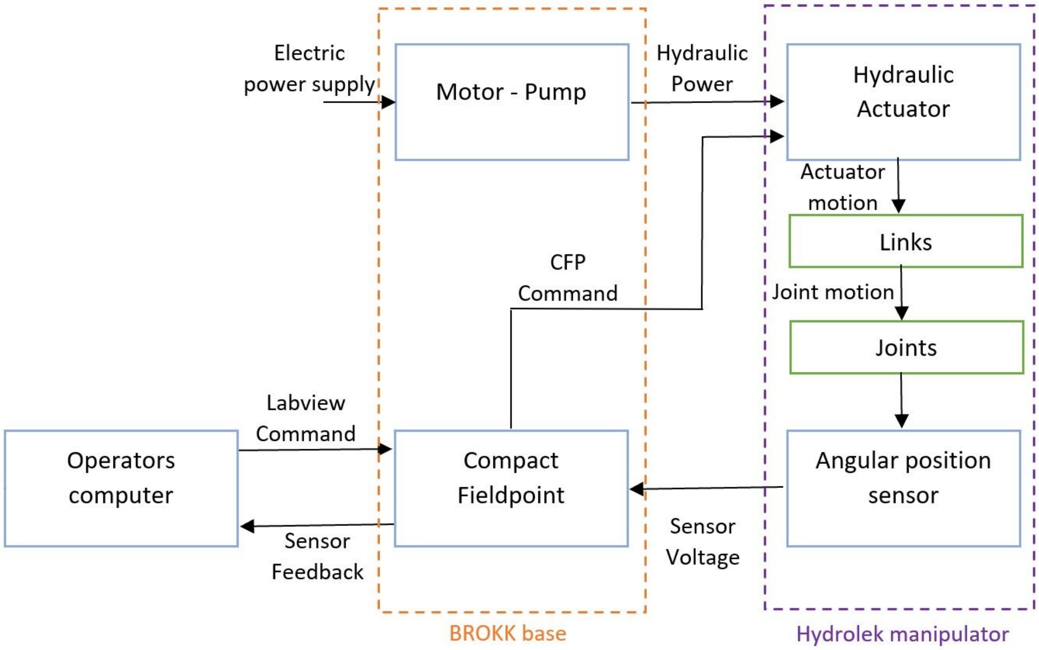


Figure 3.1: *Block diagram representing one manipulator, adapted for dynamic modelling.*

The Brokk 40 base machine is electrically powered so ideal for working inside, as there are no exhaust fumes from an engine. It consists of a moving vehicle, hydraulic tank, remote control system and (initially) single manipulator. It has an onboard hydraulic pump to power the caterpillar tracks and, by means of several hydraulic pistons, the manipulator. At 650mm wide and a height of 950mm, it is small enough to fit through a standard doorway allowing it to move around inside buildings. Being fitted with caterpillar tracks allows for movement across uneven terrain. Note, however, that once positioned, all of the pipe cutting experiments in later chapters are based on a stationary vehicle i.e. only the Hydro-Lek manipulators are moved.

The Hydrolek arms each have 7 Degrees–Of–Freedom (DOF), i.e. 6 joints plus a gripper as shown in Figure 3.2, providing a good range of movement and flexibility to perform tasks in a confined space. In principle, the end–effectors can be equipped with a variety of other tools, such as percussive breakers, hydraulic crushing jaws, excavating buckets and concrete milling heads. Without tools, each arm is 1.5m long, with a weight of 45kg and a lift capacity at full reach of 150kg. The first five joints illustrated in Figure 3.2 (i.e. the azimuth yaw, shoulder pitch, elbow pitch, forearm roll and wrist pitch joints) are all fitted with potentiometer feedback sensors, allowing the position of the end–effector to be determined during operation. The joints are actuated via hydraulic pistons, which are powered via an auxiliary output from the hydraulic pump of the base unit. The sixth joint has a continuous (360 degree) jaw rotation mechanism which, like the gripper, presently operates in open–loop only, with no feedback.

The operators computer runs the user interface, initially the existing open loop joystick teleoperation and later the closed loop assisted teleoperation system developed in this thesis, this output is then mapped to the correct output channel and sent to the National Instruments Compact Field Point (CFP) and its analogue input and output modules. This is an industrial standard Programmable Automation

Controller (PAC) offering programmable logic controller performance, whilst interfacing with a standard PC running National Instruments Labview software. The CFP output controls the proportional solenoid operated spool valves powered from the hydraulic motor, which control the movement of the hydraulic actuators. Note that the hydraulic pistons are controlled by seven pairs of control valves, where each pair has an input for both positive and negative flow. The CFP is a stand alone device running a real–time operating system, allowing for the precise sampling rates needed for discrete–time control. The PC transmits information to the CFP controller via an Ethernet networking connection. For teleoperation, a standard input device, such as a joystick, is also connected to the PC.

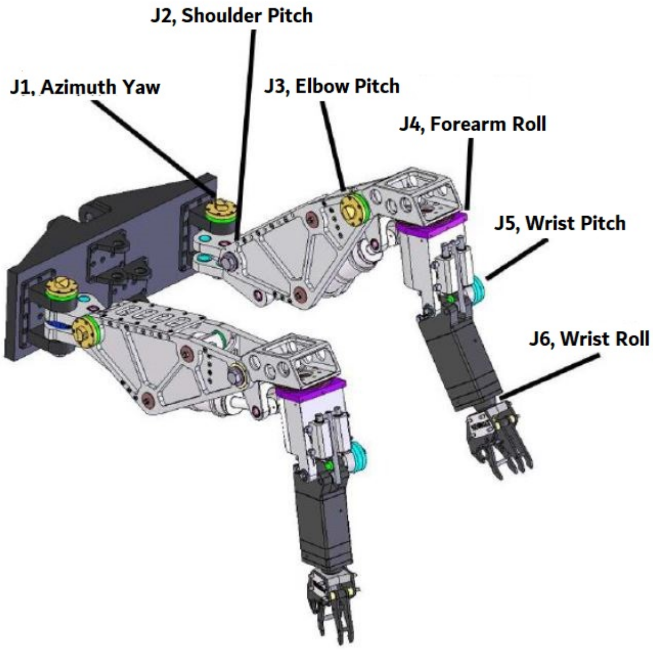


Figure 3.2: *CAD model of the Hydro-Lek manipulators with joints labelled, J1 represents Joint 1 etc..*

3.2 Previous Research

Proportional-Integral-Derivative (PID) control is one of the most widely used control systems in industry. By tuning the three parameters, the control action can be varied to achieve different outputs, improving the stability or transient response time. When the robot was first developed, a PID control system was used to control the manipulators, with the PID gains tuned by trial and error [1]. The PID system was straightforwardly implemented using standard Labview library components, and sometimes gave satisfactory performance in regard to individual joints following a set point. In fact, simple proportional control is often sufficient in this context, since the system can display inherent integral action. This is because the calibration approach used ensures that with zero control input, the joint angle largely remains constant [7]; naturally, this is not necessarily the most robust solution.

A more systematic model–based approach was taken by Robertson [8], who developed Proportional–Integral–Plus (PIP) control systems for each joint. PIP control provides a logical extension of PID methods [11], and has a similar degree of implementational complexity for this system. The same author also developed nonlinear control systems, based on the identification of state-dependent parameter (SDP) models [7, 8, 87–89]. The SDP algorithm was found to yield improved performance in comparison to equivalent PID and PIP control, for the control of individual joints and for simple resolved motion experiments e.g. drawing a figure of eight in a plane using two joints. Such control systems, and the new developments reported in this thesis, are all discussed in more detail in Chapter 6.

Further work involved integrating force control into the gripper, to prevent it from crushing more delicate objects [90]. This was acheived by adding a piezoelectric force sensor to the gripper and integrating it into the control system. The existing PID control architecture was used as a base for developing the force control. However, due to the spike in contact force caused by the gripper coming into contact with an object, the PID system had to be modified into a split range PID controller. When operating under a certain threshold force, one set of high gains are used, then when the threshold force is exceeded, a set of low gains are used. Although the force control was tested on the real gripper, it was not fully integrated and is no longer part of the present system. Finally, the Hydrolek manipulators have been used as a demonstrator for research into inverse kinematics solvers [91, 92].

3.3 Foundation work

In the context of the PhD project, initial work had to be carried out by the author to become familiar with the system, both in terms of the hardware and the existing software framework. Understanding how to send instructions to the manipulators is clearly fundamental to any later practical research. Furthermore, a considerable amount of time was spent identifying and fixing various issues arising with the system, including:

- Replacing faulty potentiometers.

- Rewiring all potentiometers to the Compact Field Point (CFP).

- Calibrating the potentiometer values to joint angles for all joints.
- Expanding basic PID control to the left manipulator (previously applied only to the right manipulator).
- Modifying existing joystick control to improve safety, forcing all valves to close when program is stopped to prevent arm movement.
- Replacing the control PC with a more up to date and capable machine.
- Replacing the main circuit board in the Brokk base unit.

3.3.1 Labview interface

As the instructions sent to the CFP originate from the Labview software, understanding how Labview and the CFP communicate with each other and the type of signals sent between the two devices is essential. From previous work, there were programs available that allowed teleoperation using joysticks, as well as a simple interface for a PID controller that reads the set point from a text file. Analysing these programs helped to develop an understanding of how the communication works, including that each joint has two inputs to the CFP, one for each direction of movement, and that the signal sent to the CFP is a voltage in the range of 0 to 10v. Communication in the other direction, from the CFP to Labview, there is a signal from the potentiometer on each relevant joint. This is an uncalibrated voltage reading, and so needs to be processed to generate a meaningful signal for control.

No record was readily available stating which channel on the CFP controlled which

Table 3.1: *Mapping of CFP channels to joint movement. Directions refer to standing at the base unit and looking out towards the manipulators i.e. the system's first person view.*

	Module		
Channel	@3	@4	@5
0	Right arm J1 left	Right arm J5 up	Left arm J3 down
1	Right arm J1 right	Right arm J5 down	Left arm J3 up
2	Right arm J2 up	Right arm wrist clockwise	Left arm J2 up
3	Right arm J2 down	Right arm wrist anti-clockwise	Left arm J2 down
4	Right arm J3 down	Right arm gripper open	Left arm J4 left
5	Right arm J3 up	Right arm gripper close	Left arm J4 right
6	Right arm J4 right	Left arm J1 left	Left arm J5 down
7	Right arm J4 left	Left arm J1 right	Left arm J5 up

joint. Therefore, controlling a specific joint in a specific direction required a methodical investigation, sending a signal to each channel and observing the result. The CFP consists of different modules, and each module has 8 input or output channels, with 2 channels per joint. Identifying which module and which channel controlled which joint eventually allows a signal from Labview to be addressed to the correct output, allowing the correct joint to be moved in the desired direction. For future reference, Table 3.1 shows the mapping between the CFP channels and the actuated joints. Hence, for example, to move Joint 5 on the right hand side manipulator upwards, a signal should be sent to Module 4 and channel 0.

Finally, to help future researchers, a list of the key Labview files created during this project is provided in Appendix A i.e. the filename and a brief description of functionality. These Labview files are available from the author.

3.3.2 Joint calibration

After some early testing, it became clear that there were some calibration problems, particularly on the left hand side manipulator which had not been used as much as the right hand side manipulator. The signal from the potentiometer on each joint is sent through to Labview as a voltage, and needs processing to give a joint angle output. The joint limits were already known for each joint, so by finding the potentiometer readings at these limits, the scaling between potentiometer voltage V_{meas} and joint angle D_{meas} can be calculated. In the first instance, the voltage per degree, α is found,

$$\alpha = \frac{D_{max} - D_{min}}{V_{max} - V_{min}} \tag{3.1}$$

where D_{max} and D_{min} are the maximum and minimum joint angle limits in degrees respectively, and V_{max} and V_{min} are the maximum and minimum voltages from the potentiometer. These values are used to find the potentiometer voltage at 0 degrees,

$$V_0 = V_{min} + (|0 - D_{min}| \times \alpha) \tag{3.2}$$

which is used to scale between the measured potentiometer value V_{meas} and the joint angle D_{meas} as follows,

$$D_{meas} = \frac{(V_{meas} - V_0)}{\alpha} \tag{3.3}$$

Repeating this procedure for each joint provides the scaling shown in Table 3.2, which represents the ‘current’ calibration framework i.e. the status at the end of the project.

Table 3.2: *Scaling between potentiometer voltage reading and joint angle.*

	Voltage		Angle		
Joint	Min.	Max.	Min.	Max.	Scaling
			Right Arm		
1	3.71	11.43	-27.0	40.6	$D_{meas} = \frac{(V_{meas}-6.792)}{0.114}$
2	1.11	13.54	-10.3	65.6	$D_{meas} = \frac{(V_{meas}-2.794)}{0.164}$
3	2.92	12.24	-42.0	21.4	$D_{meas} = \frac{(V_{meas}-7.024)}{0.144}$
4	2.33	13.67	-114.0	63.9	$D_{meas} = \frac{(V_{meas}-5.767)}{0.064}$
5	7.75	23.52	-81.5	-4.16	$D_{meas} = \frac{(V_{meas}-6.902)}{0.204}$
			Left Arm		
1	6.39	12.99	-40.6	27.0	$D_{meas} = \frac{(V_{meas}-10.77)}{0.108}$
2	4.69	13.59	-10.3	65.6	$D_{meas} = \frac{(V_{meas}-5.901)}{0.117}$
3	2.92	12.24	-42.0	21.4	$D_{meas} = \frac{(V_{meas}-9.091)}{0.147}$
4	2.33	13.67	-114.0	63.9	$D_{meas} = \frac{(V_{meas}-9.596)}{0.064}$
5	1.91	12.63	-81.5	-4.16	$D_{meas} = \frac{(V_{meas}-1.33)}{0.139}$

3.4 Concluding Remarks

The present chapter has briefly reviewed earlier research into the Brokk–Hydrolek system, and essential foundation work by the author. In prior work some years ago, two Hydrolek hydraulically actuated manipulators, each with seven degrees of freedom (7-DOF, i.e. six rotary joints and one gripper) had been attached to a Brokk-40 mobile platform, for research into the decommissioning, repairs and maintenance of nuclear plants. The machine has been a useful demonstrator system in the Engineering Department for many years, such as for undergraduate and taught postgraduate student projects, and for open–days. However, despite the original motivation for its construction, it would be true to say that most of the subsequent academic research focused on use of the system as a case study for the testing of novel control systems, with relatively little attention given to how the user interacts with the robot, or how the robot would interact with the environment in a decommissioning setting, which is the focus of Chapter 5. In the first instance, however, the following Chapter 4 develops a mechanistic model of the entire system.

Chapter 4

Mechanistic Model Development

This chapter discusses the development and testing of a detailed mechanistic model of the Brokk–Hydrolek platform, consisting of all elements of the mechanical, electrical and hydraulic systems. The model is implemented for the MATLAB SIMULINK software environment, and builds on previous work by Montazeri and Ekotuyo [9][1]. However, the previously developed model did not fully reflect the real system behaviour, in part because of several unknown parameter values. Hence, the novel contribution of the present chapter is twofold. In the first instance, research into parameter identification to find these unknown parameter values using experimental data. During this research, it became apparent that there were several problems with the existing model, in particular with regard to the dynamic behaviour of the hydraulic elements. As a result, a second contribution of the chapter is a new mechanistic model for the

[1] The present author is grateful for the earlier contribution of two Lancaster University MSc students supervised by Dr Montazeri, as cited here, namely Maxime Antoine [10] and Joseph Ekotuyo [93].

hydraulic sub system. To clarify more precisely, the mechanistic model introduced in the present thesis, encompasses the mechanical model described by Montazeri and Ekotuyo [9], with some important changes and additions as reported below, and with an entirely new hydraulic sub system.

A parameter estimation step is required since many of the physically-based parameters of the manipulator are unknown, in part due to the age of the manipulators, as well as because of various changes from the original manufacturer specifications caused by repairs and replacement parts. Hence, a sensitivity analysis is first carried out to identify the key parameters that affect the performance of the model. These parameters are subsequently estimated using a Genetic Algorithm (GA). The aim is to find mechanistically realistic parameter values that also yield model responses following the measured (experimental) performance of the real system.

Further background information is provided in section 4.1, including the motivation for the GA approach. The model is developed in section 4.2. Section 4.3 formally presents the parameter estimation problem, while section 4.4 develops the specific GA approach utilised in this thesis. This includes both a simple GA initially used, and a more complex multi–objective genetic algorithm (MOGA) developed subsequently. To optimise the GA in the following section 4.5, settings include the fitness function, crossover and mutation rates, and population size are adjusted for this particular problem, with experimental data used to find the optimum settings. The results are presented in sections 4.6 and 4.7, which focus on the mechanical component of the model and the entire system, respectively.

4.1 Motivation for the model

The mechanistic model should allow consideration of the dynamics, as well as the kinematics of the manipulator. This is important as the manipulators are highly nonlinear, due in a large part to the hydraulic actuators that drive the joints. The model should also allow potential faults to be investigated, for example looking at the effect of one actuator leaking.

The importance of this problem in a robotic context is investigated by, for example, Swevers et. al. [94]. In this regard, previous research for the Brokk-Hydrolek and other hydraulic manipulators has demonstrated the practical utility of a quasi-linear, State Dependent Parameter (SDP) model; e.g. [7, 95]. The parameters of the SDP model are functionally dependent on the measured variables [96], such as joint angles and velocities. The approach is ideal for capturing the essential non-linear behaviour of the system when a relatively straightforward dynamic equation is required for model-based control. However, as discussed later in Chapter 6, such SDP models were initially limited to individual manipulator joints, and they had not been designed to represent the complex mechanistic interactions between various mechanical, hydraulic and electrical components of the Brokk–Hydrolek system.

Early testing of the Hydro-Lek manipulators identified a need for improved control systems. The PID control that was being used did not provide accurate or sufficiently rapid movement. Furthermore, the more advanced control systems alluded to above had not been implemented for all joints, something that would be required for the pipe cutting experiments in the next chapter. Hence, in general terms, the development of a reliable mechanistic model of the dual manipulator platform, aims to facilitate the investigation and development of new control systems in the future.

The approach taken in this thesis, is to decompose the system into several dynamic subsystems and subsequently to integrate the different modules together. The model aims to not only provide an accurate and detailed representation of the complex mechatronic system, but to also provide a grey-box model structure with physically meaningful parameters. In this manner, the model will be a reliable simulator through which the dynamic and kinematic characteristics of the manipulator can be analysed. This is essential for the design of joint-level and supervisory control algorithms, to accomplish complex tasks encountered in nuclear decommissioning (such as remote pick and place, welding and pipe cutting).

Parameter estimation for robotic systems is a challenging task, especially when the number of degrees of freedom and the number of parameters to be estimated are high. The main difficulty is that the underlying problem is nonlinear and non-convex in nature. The most common technique to address the parameter estimation problem is to use an equation error identification approach, which assumes that the dynamic model of the robot is parameterized linearly with respect to the unknown model parameters, and hence the parameters can be estimated using Least Squares (LS) error minimisation criteria [97–99].

Various techniques are reported in the literature to help alleviate the problem of uncertainties from modelling errors and measurement noise. For example, band-pass filtering of joint positions provides one solution, shown in Gautier et al. [100]. Furthermore, Janot et al. [101] compare the performance of this technique with instrumental variables, total LS and output error methods. More advanced robust estimation such as maximum likelihood and extended Kalman filter based techniques have also been proposed e.g. [97].

An alternative approach is to formulate an output error system identification problem in nonlinear LS terms and to use nonlinear programming [100, 102]. When the performance function is smooth, a local optimization technique may then converge to the global optimum. However, Janot et al. [101] show that application of a multi-start Gauss-Newton algorithm for parameter estimation of a 6-DOF robot is not a viable strategy. The problem of multi-modality of the performance surface can be solved by the design of a suitable filter, as shown by Tohme et al. [103], and the technique is efficient when the first and second order derivatives of the performance surface are available analytically. Unfortunately, this assumption is not valid in the present case since, due the complexity of the model, no closed-form analytical equation exists. Furthermore, the performance indices assumed in the optimization process, such as the output error infinite norm, are not necessarily differentiable.

Therefore, a more effective approach to search a large parameter space is to use evolutionary-based algorithms. These include genetic algorithms (GA) [104], related multi-objective approaches [105,106], particle swarm optimization (PSO) [107] and the recently proposed differential evolution approach [108]. Gotmare et al. [109] discuss the suitability of GAs for various optimisation problems in system identification and filter design. Yao [110] and Nyarko [111] provide similar in the context of parameter estimation more specifically. Use of PSO as an alternative to GA optimisation is primarily motivated by the relative simplicity of implementation and the ability to memorize the good solutions by all particles. As a result, PSO has been used for various control and identification problems e.g. [112–114]. However, in contrast to the present parameter estimation problem, PSO is usually considered for real-time and adaptive applications because of its computational simplicity [115].

4.2 The dynamic model

Each HydroLek manipulator has seven DOF, with a continuous (360 degree) gripper rotation mechanism [7]. In total, five linear actuators and two rotary actuators are used, as shown in Figure 4.1. Each joint is fitted with potentiometer feedback sensors, allowing the position of the end-effector to be determined during operation, using the measured joint angles and the forward kinematic equations. The joints are actuated via hydraulic pistons, which are powered via an auxiliary output from the hydraulic pump in the Brokk base unit.

It is known that the dynamic equations of a manipulator can be derived using standard Newton-Euler theory as follows [98]:

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{F_c}\mathrm{sign}(\dot{\boldsymbol{q}}) + \boldsymbol{F_v}\dot{\boldsymbol{q}} = \boldsymbol{\tau} \tag{4.1}$$

where $\boldsymbol{\tau}$ denotes the vector of forces/torques generated by the hydraulic actuators, $\boldsymbol{D}$ is the manipulator inertia matrix, $\boldsymbol{C}$ is the coriolis and centrifugal matrix, $\boldsymbol{g}$ is the gradient of the potential energy, $\boldsymbol{F_c}$ is the coulomb friction and $\boldsymbol{F_v}$ is the viscous friction. The vector $\boldsymbol{q}$ contains all joint angles for the revolute joints. However, it would be extremely complex to analyse the dynamics of the HydroLek manipulator with both linear and rotary joints analytically in this way. Therefore, use of Equation (4.1) would impose several limitations on the derivation of the dynamic model, including infinitely rigid links and joints and backlash free joints [116]. Furthermore, for relatively high DOF, parametrisation of Equation (4.1) becomes extremely complicated, even using symbolic software. To overcome these limitations, and to exploit existing Computer-Aided Design (CAD) models for the manipulator, the present research instead relies on a numerical model to solve the equations of motion. This is achieved by modelling various components of the manipulator using numerical and experimental techniques, and subsequently integrating these into a unified simulation for the purpose of system identification, parameter estimation and the design of end-effector trajectories, as well as the wider control objectives. This process is usually referred to as Robot Calibration in the robotic literature [117].

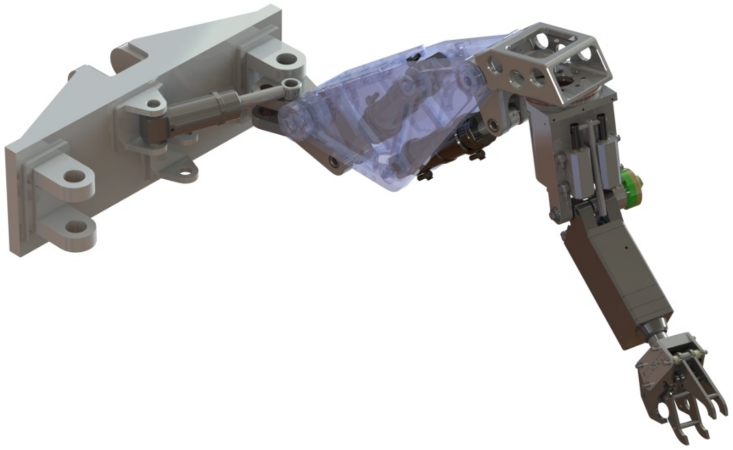


Figure 4.1: *Graphical render of 7-DOF Hydrolek manipulator, with link 2 made transparent to show linear actuators for Joint 2 and 3.*

4.2.1 Operator computer subsystem

The model comprises all elements of the system, as summarised in Figure 3.1 (section 3.1), including the mechanical, electrical and hydraulic parameters of the manipulator. In the physical system a command signal S_{in} is generated by a feedback controller or set directly by the operator in the case of joystick teleoperation, and is processed and calibrated before it can be sent to the National Instruments Compact Field Point (CFP) module.

The role of the operator computer in Figure 3.1 is to convert the signal S_{in} to a calibrated voltage command for the proportional amplifiers i.e. either $V_{amp_{A_i}}$ or $V_{amp_{B_i}}$ for joint i, and to assign the voltage to the correct input on the CFP (Table 3.1 shows the mapping between CFP channel and joint movement). Thus, for each joint i, the voltages $V_{amp_{A_i}}$ and $V_{amp_{B_i}}$ are expressed as follows:

$$V_{amp_{A_i}} = \begin{cases} A_i + \left(|S_{in}|.\frac{B_i - A_i}{100}\right), & \text{if } S_{in} > 0 \\ 0, & \text{if } S_{in} \leq 0 \end{cases} \tag{4.2}$$

$$V_{amp_{B_i}} = \begin{cases} C_i + \left(|S_{in}|.\frac{D_i - C_i}{100}\right), & \text{if } S_{in} < 0 \\ 0, & \text{if } S_{in} \geq 0 \end{cases} \tag{4.3}$$

In Equations (4.2) and (4.3), $V_{amp_{A_i}}$ and $V_{amp_{B_i}}$ are the voltage input at the proportional amplifiers A and B respectively, while the parameters A_i , B_i , C_i , D_i are the joint i calibration coefficients and are tuned to optimise performance.

4.2.2 Mechanical model subsystem

The mechanical subsystem involves two main elements, as illustrated in Figure 3.1, the links and the joints. The input of the mechanical subsystem is the movement of the hydraulic actuators, which applies force to the manipulator links, causing the joints to move, and its output is the angular position of each joint.

Since the analytical solution, using the kinematic equations of each joint, would be extremely complex for the 7 DOF manipulator, a numerical approach using the Matlab SimMechanic toolbox is adopted. The left and right manipulator mechanical subsystems have the same structure i.e. a succession of revolute joint blocks and link subsystems. Revolute joint blocks are actuated by a hydraulic actuator and the angle is sensed as the output variable. Geometric information and properties about each link are extracted from CAD files.

To find the base parameters suitable for estimation, it is assumed that the inertia parameters for each link are calculated using the CAD file information. Therefore, it would be reasonable to consider the mass for each link, and damping and stiffness for each revolute joint, as the unknown parameters for the subsequent optimisation. Finally, a 3D representation of the manipulator is shown in Figure 4.2 using Matlab Mechanical Explorer. Incorporating the CAD model of the manipulator in this way allows for straightforward understanding of the manipulator behaviour. For example, problems such as collision detection and dual-manipulator coordination can be examined using this visualisation.

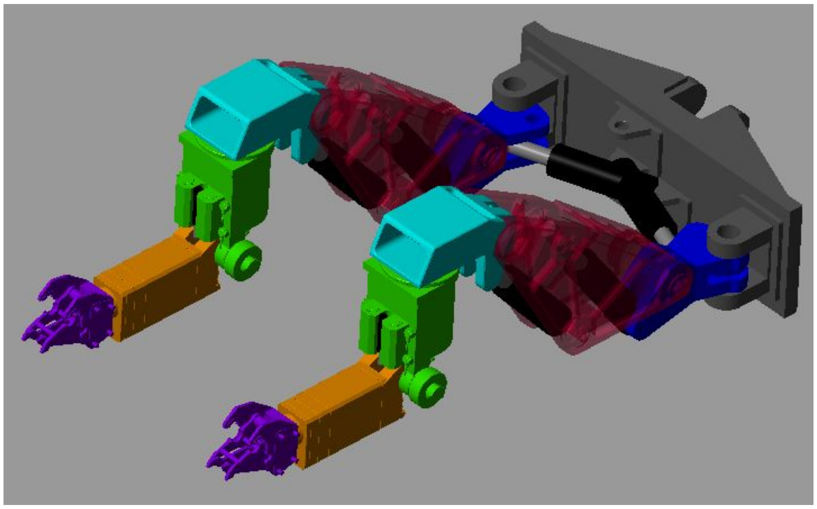


Figure 4.2: *Simscape mechanical explorer 3D view of the manipulator model, with the newly modified hydraulic system model (see section 4.2.4).*

4.2.3 Hydraulic actuator subsystem

The hydraulic actuating subsystem, shown in Figure 4.3, consists of two further subsystems, namely the valves and the hydraulic cylinders. The valve system is decomposed into the proportional amplifier, solenoid and directional valve components. The four-way directional spool valve operates with the solenoid coil currents I_{Coil_A} and I_{Coil_B}. The voltage commands V_{Amp_A} and V_{Amp_B} applied to the proportional amplifiers A and B are converted to the current coil, and energize the solenoids A and B:

$$I_{coil_{H_i}} = \begin{cases} I_{min_i}, & \text{if } V_{amp_{H_i}} = 0V \\ V_{amp_{H_i}} \frac{I_{max_{H_i}} - I_{min_{H_i}}}{10} + I_{min_{H_i}}, & \text{if } 0 < V_{amp_{H_i}} < 10V \\ I_{max_i}, & \text{if } V_{amp_{H_i}} \geq 10V \end{cases} \tag{4.4}$$

where V_{ampH_i} is the joint i voltage signal at the input of proportional amplifier H (in which H is either A or B), I_{coilH_i} is the output of the proportional amplifier H, i.e. the current that will energize the solenoid H of the joint i actuating system, and I_{maxH_i} and I_{minH_i} are the saturation current parameters associated with the amplifiers H of the joint i actuating system. The dynamics of the solenoid system A or B are represented using Newtons law applied to the plunger A or B belonging to the solenoid device, as shown in Figure 4.4 [118],

$$F_{S_{H_i}} = -\frac{\alpha {I_{coilH_i}}^2}{(\beta + \gamma x_p)^2} = m_p \ddot{x}_p + \lambda_p \dot{x}_p + K_p x_p \tag{4.5}$$

where m_p is the mass of the plunger, while λ_p and K_p are the stiffness and the damping coefficient of the plunger. The coefficients α, β and γ are constant and depend on the permeability coefficients, effective magnetic flux areas, magnetic field lengths, and turn numbers. These coefficients are parametrized by selecting two points in the $(F_{S_{H_i}}, x_{P_{H_i}})$ space for the joint i. The variable I_{coilH_i} (again, H represents either A or B) is the current passing through the solenoid and x_{pH_i} is the displacement or stroke of the plunger. Finally, $F_{S_{H_i}}$ is the solenoid force generated by the plunger.

The hydraulic equation links spool displacement x_i and the pressure and flow rate inputs/outputs of the valve i.e. (Q_A,P_A), (Q_B,P_B), (Q_S,P_S) and (Q_T,P_T) for joint i. Here, x_i depends on the displacement of plunger A and B, i.e. $x_{p_{A_i}}$ and $x_{p_{B_i}}$, is defined as follows:

$$x_i = x_{p_{A_i}} - x_{p_{B_i}} \tag{4.6}$$

The volume flow between valve ports in the spool valve is calculated using Bernoullis

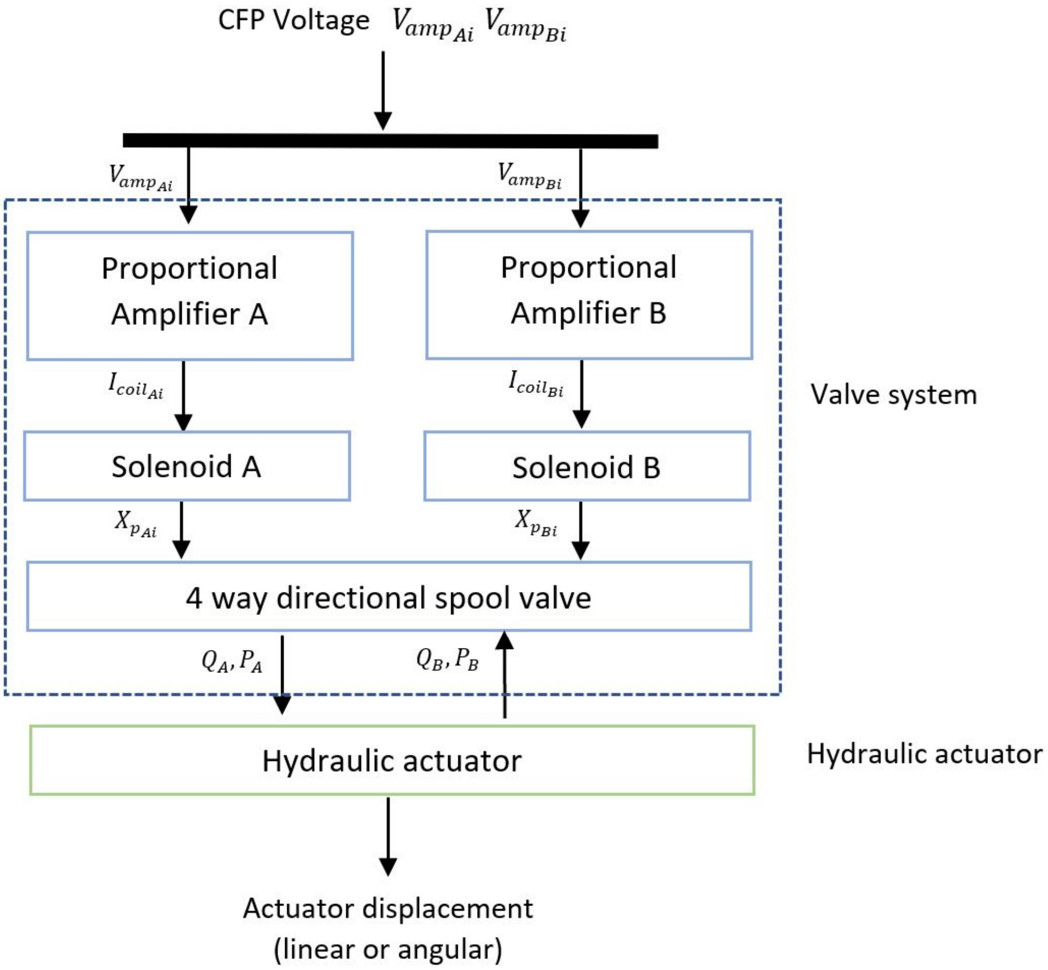


Figure 4.3: *Schematic representation of hydraulic actuator system.*

law. In particular, for a positive deflection x_i, oil flows from ports P to A and B to T are as follows [119]:

$$Q_A = Q_{P\to A} = \begin{cases} C_q.A(x).sgn(P_S - P_A).\sqrt{\frac{(2|P_S - P_A|)}{\rho}} & \text{if } 0 < x_i < w \\ C_q.A_{max}.sgn(P_S - P_A).\sqrt{\frac{(2|P_S - P_A|)}{\rho}} & \text{if } x_i \geq w \end{cases} \tag{4.7}$$

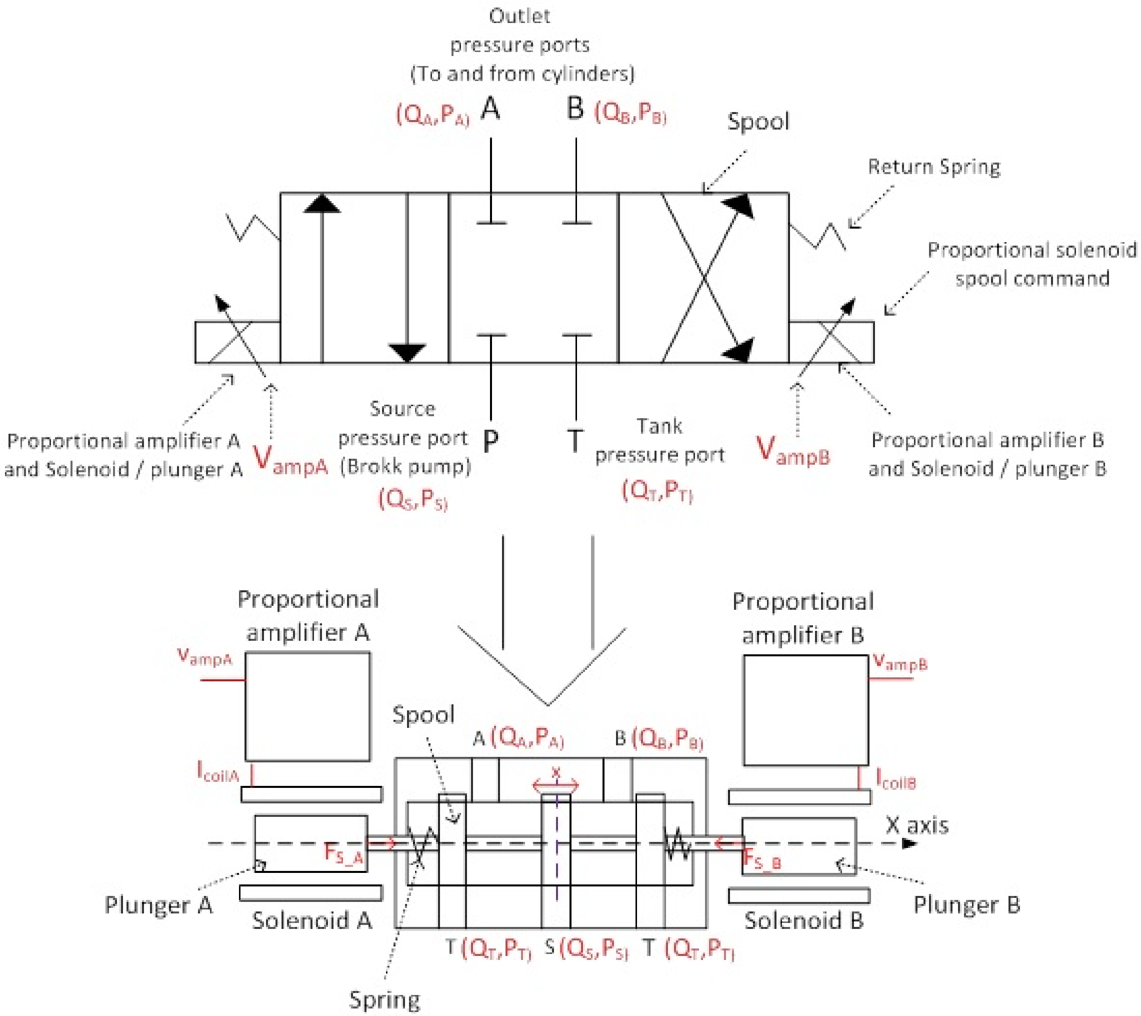


Figure 4.4: *Illustrative model element showing the components of the solenoid device.*

$$Q_B = Q_{B\to T} = \begin{cases} C_q.A(x).sgn(P_B - P_T).\sqrt{\frac{(2|P_B - P_T|)}{\rho}} & \text{if } 0 < x_i < w \\ C_q.A_{max}.sgn(P_B - P_T).\sqrt{\frac{(2|P_B - P_T|)}{\rho}} & \text{if } x_i \geq w \end{cases} \tag{4.8}$$

where P_A, P_B, P_S, P_T are the pressure values at ports A, B, P and T respectively and sgn is the sign function. Here, C_q is the contraction flow coefficient of the valve, w is the width of the slot in the valves sleeve, $A(x_i)$ is the area of the corresponding orifice as a function of spool displacement and A_{max} is its maximum value. The parameter $A(x_i)$ is linked to the valve geometry architecture, so it is not possible to derive $A(x_i)$ as a function of x_i when no information is available about the valve design. Here, it is initially assumed that there is no leakage flow, no viscous forces and fluid is flowing at low speed. However, these parameters can optionally be defined in the corresponding model elements for the hydraulic actuator.

The final elements of the hydraulic model are the linear and rotary cylinders. Joints 1, 2, 3, 5 and 7 on each manipulator are actuated with a linear double acting cylinder. Assuming zero mechanical friction and gravity effects, the force balance on axis x_i of the cylinder rod is:

$$
\begin{aligned}
F_R &= A_A P_A - A_B P_B - F_C = (M_R + M_L)\ddot{x}_i + B_D \dot{x}_i + K_S x_i \\
F_C &= \begin{cases} (x_i - S).K_p.v & \text{if } x_i \geq S, v > 0 \\ x_i.K_p.v & \text{if } x_i \leq 0, v < 0 \end{cases}
\end{aligned}
\tag{4.9}
$$

where F_R is the rod force, F_C is hard stop force, K_p is penetration coefficient, v is Cylinder rod velocity, P_A and P_B are the outlet pressure of the valve system, A_A and A_B are the rod surface areas in chambers A and B respectively, M_R is the mass of the rod, M_L is the mass of the load system, x_i is the rod displacement at joint i, K_S is the load spring constant and B_D represents the viscous damping. For the present research, these hydraulic actuator equations are implemented using the Matlab SimHydraulics library.

4.2.4 Modifications to hydraulic system model

Some major modifications to the implementation of the previously developed hydraulic sub-system model of Montazeri and Ekotuyo [9] were required due to the Simulink implementation of this part of the model being fundamentally flawed in several ways. The original model had the hydraulics as a separate physical system, measuring the hydraulic actuator displacement then attempting to use the geometry of the manipulator to calculate the torque that would be applied to each joint, this torque was then applied directly to the mechanical joint to cause movement. This modelled the system more like an electric manipulator, where the motors do apply torque at the joint, but is not how the hydraulic manipulator operates. The new version of the model places the hydraulic actuators between the links, as can be seen in Figure 4.2, so that the movement of the actuator directly moves the links, applying force at the actuator mounting points. This integrates the mechanical and hydraulic systems into one physical model in Simulink.

The hydraulic systems are modelled using standard SimHydraulics and SimMechanics tools. First the supplied voltage is passed to the relevant solenoid, dependent on whether the signal is positive or negative, where the spool displacement is calculated. This spool displacement controls oil flow at the input to the 4 way directional valve, which directs the flow into the hydraulic cylinder, the oil flow into and out of the hydraulic cylinder causes movement of the piston. The extension or contraction is used to apply a force at the actuator mounting points, along the axis of the actuator, causing the link to pivot at the joint.

This change means the model accurately reflects how the real system works, it also makes the model computationally simpler as a layer of calculations, calculating the torque, has been removed. There were also other smaller changes, such as including gravity in the model, which is not negligible, and adding joint limits to stop the joints extending further than they are physically capable, a problem with the old model.

The cumulative effect of these changes resulted in a model that much more accurately reflects the real system. This was clear before doing any parameter tuning just by looking at the output of a single joint, which was now in the correct range of motion (previously the output of the old full system model was a factor of 100 outside of the real manipulator range of motion, due to incorrect torque calculations).

4.3 Parameter estimation

Due to the age of the manipulators, data sheets containing values for many of the parameters alluded to above are unavailable, and certain parameters quoted by the manufacturer have undoubtedly changed over time through wear and tear, in addition to replacement parts and other practical matters arising. In part for these reasons, in this section the estimation problem to be solved using evolutionary algorithms is formulated. Putting all the parameters of the mechanical and hydraulic subsystems together would lead to a very large search space, making such a high-dimensional and nonlinear estimation problem a challenging task. Instead, by adopting a sequential identification procedure, the estimation is performed in two steps.

The first step is to estimate the parameters of the mechanical subsystem, assuming that the hydraulic subsystem is represented by a scalar gain (K). This ensures that the unknown parameters of the hydraulic model do not affect the tuning of the mechanical system. To illustrate the approach, the focus here is on Joint 2 of the right hand side manipulator (Figure 4.2). To determine the base parameters of the mechanical subsystem, a sensitivity analysis was conducted, details of this are given later. For this purpose, the spring stiffness (S) and damping coefficient (D), as well as the mass of the six manipulator links (M_i $i = 1, \dots, 6$ with link 6 connecting to the gripper) are the set of parameters to be estimated, as the only unknown physically meaningful parameters in the mechanical subsystem. Note that the exact masses of the individual links are unavailable but their geometry is known via the previously developed CAD models. Hence, by adjusting the estimated mass, the inertia of each link can be tuned, which proves to have a significant impact on the performance.

4.3.1 Formulation as a single objective optimization problem

To find a mathematical framework for the estimation problem, the parameters of the mechanical or hydraulic subsystems are considered together as a single vector $\boldsymbol{\theta}$. The input/output data used to solve this estimation problem are the voltage applied to the joint i, i.e. $u_i(k)$, and the joint angle measured from the potentiometer of joint i, i.e. $y_i(k)$. Both experimental and simulation data are sampled at $0.01s$ intervals. To find suitable parameters that relate $y_i(k)$ and $u_i(k)$, the search space Ω is defined to include all prior knowledge about the physical properties of these parameters. As a result, the search space of the parameters for the mechanical subsystem $\Omega_M \subset \mathbb{R}^9$ is defined as:

$$\Omega_M = \{\boldsymbol{\theta}|\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots \theta_9), \theta^M_{min_i} \leq \theta_i \leq \theta^M_{max_i}, i = 1, 2, \ldots, 9\} \tag{4.10}$$

and for the hydraulic subsystem it will be $\Omega_H \subset \mathbb{R}^{16}$:

$$\Omega_H = \{\boldsymbol{\theta}|\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots \theta_{16}), \theta^H_{min_i} \leq \theta_i \leq \theta^H_{max_i}, i = 1, 2, \ldots, 16\} \tag{4.11}$$

Having specified the parameterization of the mechanical (4.10) and hydraulic (4.11) subsystems, the identification error for joint i is defined as:

$$\varepsilon_i(k; \boldsymbol{\theta}) = y_i(k) - \hat{y}_i(k; \boldsymbol{\theta}) \tag{4.12}$$

where $\hat{y}_i(k; \boldsymbol{\theta})$ represents the output of joint i in the model for a specific parameter vector $\boldsymbol{\theta}$.

To judge the quality of a particular value of $\boldsymbol{\theta}$, it is necessary to define a suitable error signal measure. Here we choose two types of cost function. The first is defined as the p-norm of the absolute error signal:

$$J_N(\boldsymbol{\theta}) = \left(\frac{1}{N} \sum_{K=1}^{N} |\varepsilon_i(k; \boldsymbol{\theta})|^p \right)^{\frac{1}{p}} \qquad p = 1, 2, \infty \tag{4.13}$$

and the second performance measure is defined as the p-norm of the relative error signal for the joint i with respect to the p-norm of the corresponding measured data for the joint i:

$$\bar{J}_N(\boldsymbol{\theta}) = \left(\frac{1}{N} \sum_{K=1}^{N} \left| \frac{\varepsilon_i(k; \boldsymbol{\theta})}{\|y_i\|_p} \right|^p \right)^{\frac{1}{p}} \qquad p = 1, 2, \infty \tag{4.14}$$

Optimality may, therefore, be expressed by selecting the parameter values that yield the minimal value of the performance measures above. Hence, in relation to Equation (4.13), the estimated parameter vector $\hat{\boldsymbol{\theta}}_N$ is:

$$\hat{\boldsymbol{\theta}}_N = \arg\min_{\boldsymbol{\theta}\in\Omega_M} \frac{1}{N}\sum_{k=1}^{N} |\varepsilon_i(k;\boldsymbol{\theta})|^p \qquad p = 1, 2, \infty \tag{4.15}$$

A similar argument applies to the cost function (4.14).

$$\hat{\boldsymbol{\theta}}_N = \arg\min_{\boldsymbol{\theta}\in\Omega_M} \frac{1}{N}\sum_{K=1}^{N} \left|\frac{\varepsilon_i\,(k;\boldsymbol{\theta})}{\|y_i\|_p}\right|^p \qquad p = 1, 2, \infty \tag{4.16}$$

4.3.2 Mechanical sensitivity analysis

Before developing a numerical algorithm to solve the optimization problem, an identifiability analysis on the influence of the parameters selected for optimization on the model output $\hat{y}_i(k;\theta)$ is carried out.

This was achieved by looking at the model output $\hat{y}_i(k;\boldsymbol{\theta})$ for different values of the parameters, observed by plotting them numerically. By changing the value of one parameter at a time to cover the full range of possible values and plotting the model output, the effect of each parameter can be seen. Figures 4.5 and 4.6 show the simulation output for varying joint spring stiffness and damping coefficient respectively. This gives a visual indication of the sensitivity of the different parameters.

For joint 2 (again as an example), the results show that the output $\hat{y}_2(k;\boldsymbol{\theta})$ is highly sensitive to the damping and stiffness and hence these are identifiable parameters. By contrast, the sensitivity with respect to the masses is not as high and, in fact, reduces by moving from the mass of link 6 to the mass of link 1. This is addressed by noting that the sensitivity of $\hat{y}_2(k;\boldsymbol{\theta})$ with respect to the parameters in the vector $\boldsymbol{\theta}$ is a function of time. This can be seen in Figure 4.5 where the sensitivity to parameter changes increases with time. This means it is possible to find time segments in which the output $\hat{y}_2(k;\boldsymbol{\theta})$ is more sensitive to some specific parameters than other parameter. To illustrate this more clearly Figure 4.7 shows the sensitivity of some of the parameters for a segment of one experiment. In fact, this property is used in the multi-objectivization of the output error performance index to improve the parameter estimation accuracy when using the proposed multi-objective GA algorithm. This will be explained in more detail in Section 4.4.

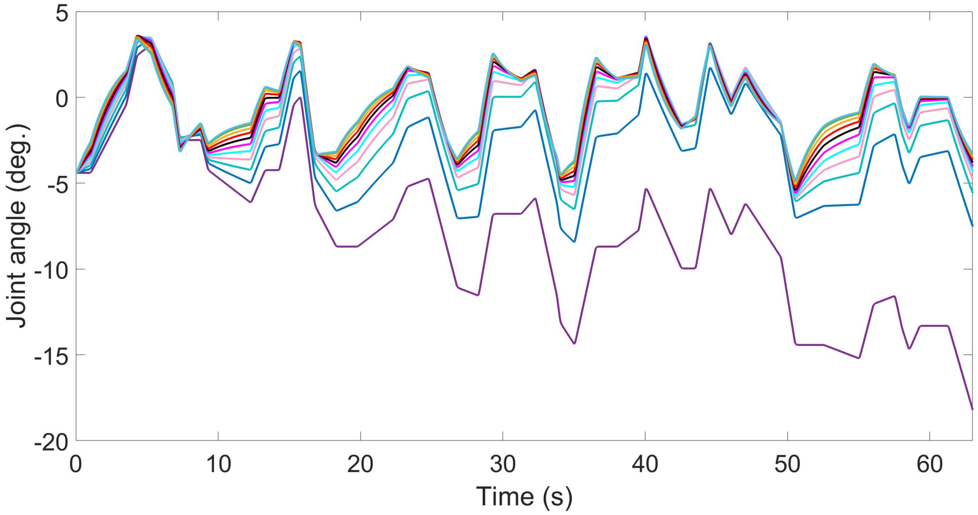


Figure 4.5: *Simulation output, angle of Joint 2, for varying spring stiffness between* 5 *and* $100mN/rad$.

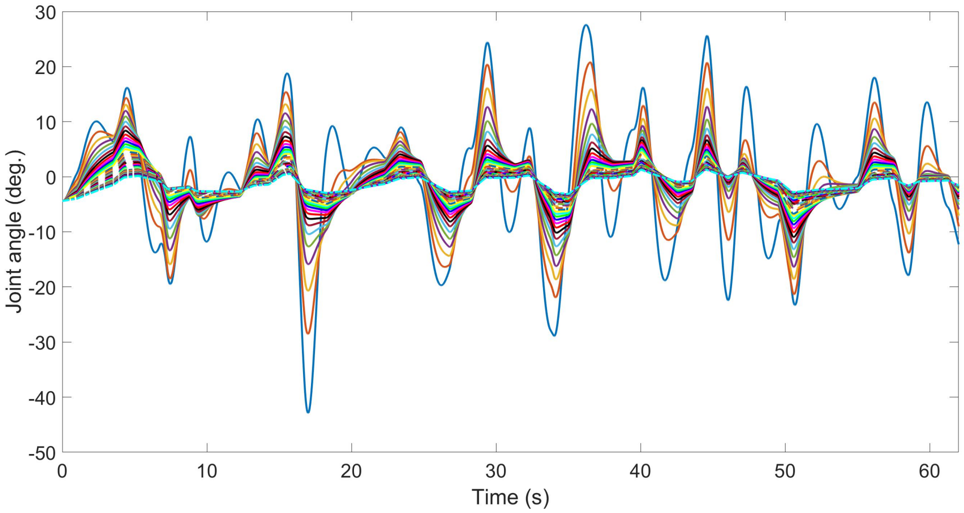


Figure 4.6: *Simulation output, angle of Joint 2, for varying the damping coefficient from* 10 *to* $400 msN/rad$.

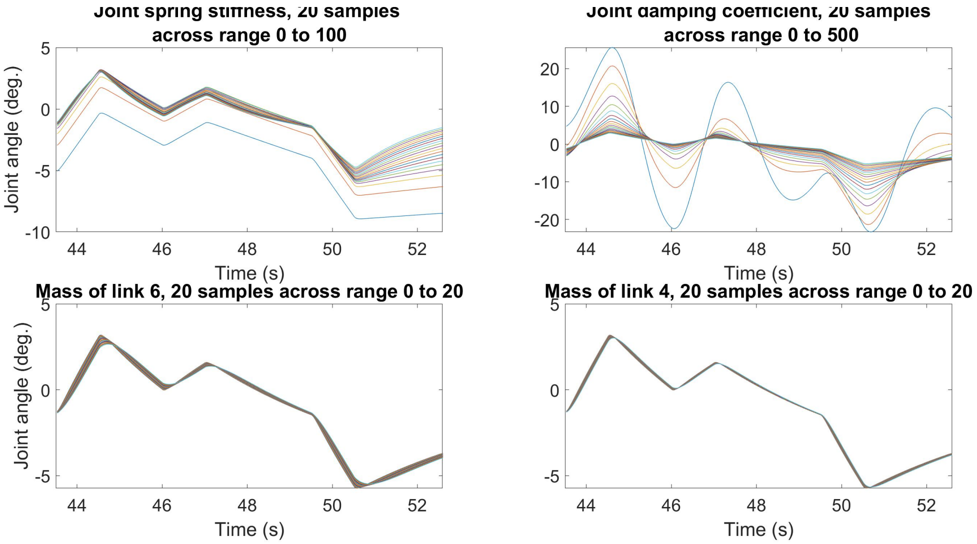


Figure 4.7: *Sensitivity analysis showing the output of Joint 2 around segment 6 (i.e. 43.5s to 52.6s) of an illustrative simulation experiment, for four of the parameters. For each parameter 20 values are plotted.*

4.3.3 Hydraulic system parameter identification

The key parameters of the hydraulic subsystem are identified as the next step. For this purpose, the parameters of the mechanical subsystem are set as fixed values.

For the hydraulic subsystem, there are too many parameters to include in the GA optimisation without encountering significant identifiability and computation issues. Hence, a preliminary sensitivity study is first completed to identify the key parameters. This was done in the same manner as with the mechanical system, changing one parameter at a time to cover the range of possible values and plotting the output to see the affect it has on the output. This was completed for all 63 parameters of each joint of the hydraulic system model. Figures 4.8 and 4.9 show examples of these tests for the Piston area A (A_A in Equation (4.9)) and the Solenoid A force ($F_{S_{A_i}}$ in Equation (4.5)) respectively, plots for all parameters are omitted for brevity. Plots for the parameters that have the most impact on the output are included in Appendix B. In the same way as for the mechanical parameters, the plots showing the model output whilst varying a single parameter are visually inspected to identify which parameters have the greatest impact on model output. An example of parameters not included in the GA are shown in Figure 4.10. Figure 4.10a shows the model output for varying the parameter 'Pump pressure', which is a unit less constant used by Simulink to control the pressure in the 'ideal hydraulic source' block from the Sim Hydrualics library. The pump gauge pressure is directly proportional to this value. As can be seen from the figure the parameter does have some impact on the output, but considerably less than other parameters, as such this parameter was not included in the GA. Figure 4.10b shows an output that was typical of many of the parameters where no noticeable effect on the model output is shown, this type of output allowed many parameters to be discounted straight away.

Inspection of the results show that the sixteen most significant parameters to be

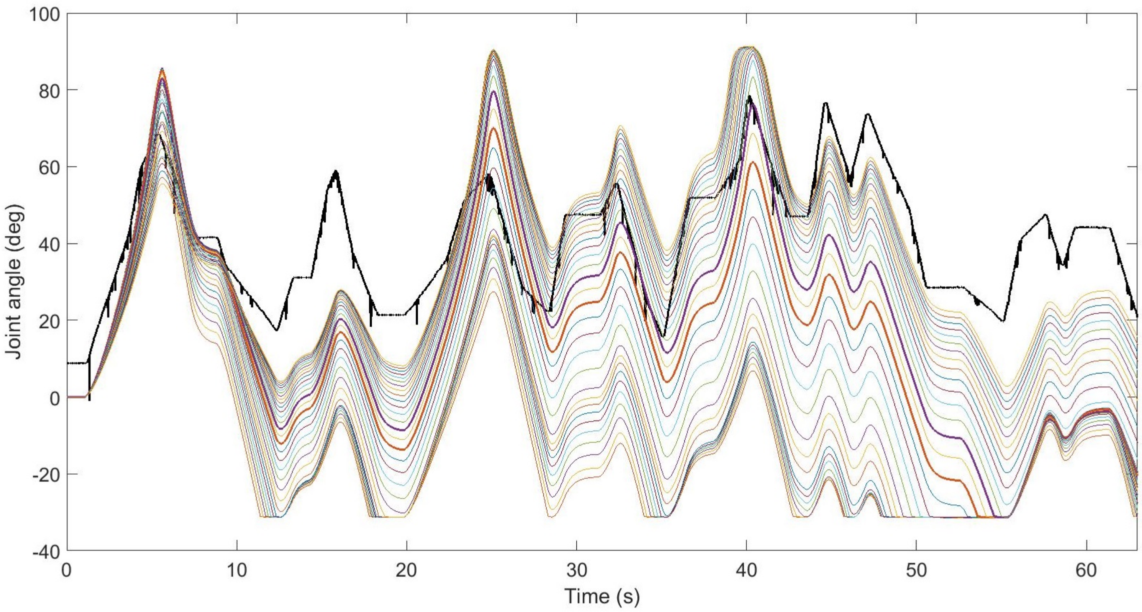


Figure 4.8: *Simulation output using full model, angle of Joint 2, for varying piston area,* A_A *from 500 to 2000* mm^2*.*

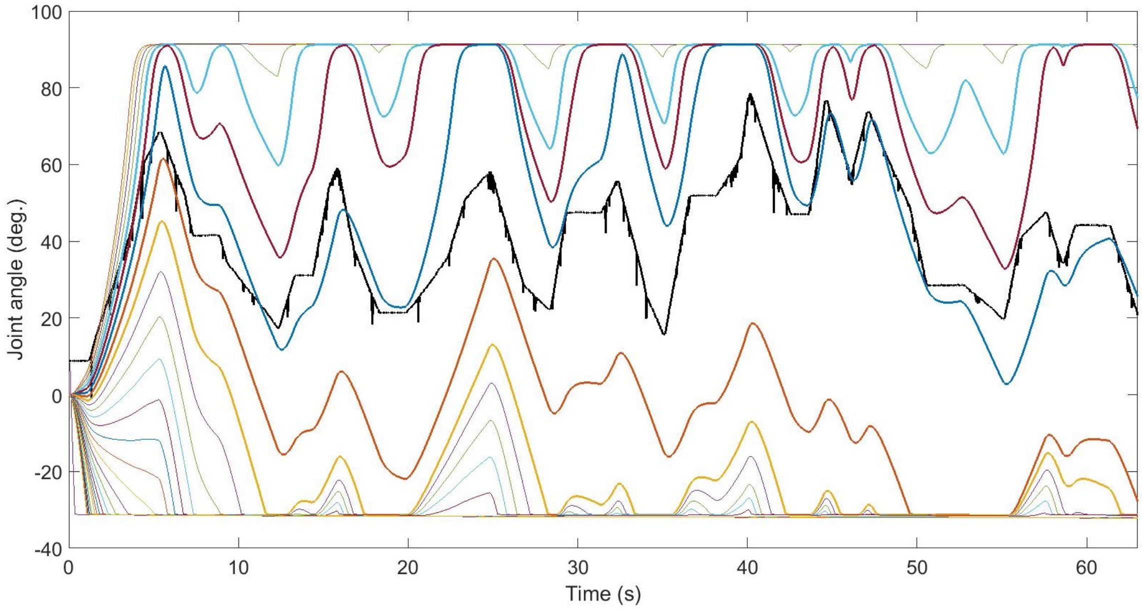
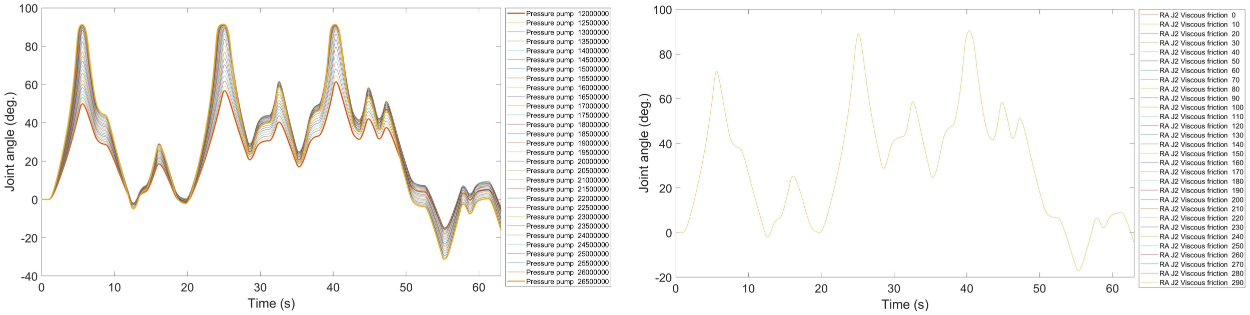


Figure 4.9: *Simulation output using full model, angle of Joint 2, for varying solenoid A Force,* $F_{S_{A_i}}$ *from 1 to 30 N.*



(a) Parameter representing pump pressure, range from 12000000 to 26500000.

(b) Viscous friction, range from 0 to 290 $N/(m/s)$.

Figure 4.10: Example of parameters not included in GA.

considered by the GA are:

- The piston area A_A in Equation (4.9) opposite the rod.
- The piston area A_B in Equation (4.9) at the opposite side of the hydraulic actuator, namely the side with the rod hence with a smaller area.
- The maximum ($I_{max_{H_i}}$) current supplied to the hydraulic valve proportional amplifier in Equation (4.4), for both valves A and B. These are used to adjust the gain provided by the amplifier, directly affecting solenoid movement which subsequently controls oil flow into the actuators.
- The solenoid force ($F_{S_{A_i}}$, $F_{S_{B_i}}$) and stroke ($x_{p_{A_i}}$, $x_{p_{B_i}}$) in Equation (4.5) are both parameterized with two coefficients each. These parameters are used to generate a force stroke curve for the solenoid, so that given an input current a solenoid displacement is calculated. The movement of the solenoid controls oil flow to the actuator valves.

- The solenoid stiffness λ_p and damping K_p from Equation (4.5), for both valves A and B. These parameters are linked to the design of the solenoids, but are also likely to have changed over time.

Other parameters such as piston stroke length and starting position, have a noticeable impact and are different for each joint, but do not need including in the GA as they can be determined from the physical system.

These chosen parameters and the range of values they were tested over are shown in Table 4.1. The range of values for each parameter were chosen to cover a realistic possible range of values when an idea of a realistic range was available, for example with piston area an approximate value is known from the real system, or where no realistic value is known the values were chosen to give a good range around the default parameter value.

Finally, the non-convexity of the problem is investigated by plotting the performance surface $\bar{J}_N(\boldsymbol{\theta})$ for the two and infinity norms as functions of the parameters. For example, Figure 4.11 shows the search landscape for an illustrative pair of hydraulic subsystem parameters. Figure 4.11 and similar plots for other parameters show that the search landscape is not smooth, a situation that is worse when considering the combined dynamics of the actuator and mechanical subsystems.

4.3.4 Formulation as a multi objective optimization problem

In Section 4.3.1, the problem of finding an estimate $\hat{\boldsymbol{\theta}}$ of the parameter vector $\boldsymbol{\theta}$ was formulated as minimization of the objective function $J_N(\boldsymbol{\theta})$ in Equation (4.13) or

Table 4.1: *Significant Hydraulic parameters and the range of values tested.*

Parameter	**Min**	**Max**	**units**
Piston area A_A	500	3500	mm^2
Piston area A_B	500	2000	mm^2
Proportional amplifier current A I_{max_A}	0.30	3.20	A
Proportional amplifier current B I_{max_B}	0.25	3.15	A
Solenoid force A $F_{S_{A_i}}$ F1	1.00	30.00	N
Solenoid force A $F_{S_{A_i}}$ F2	0.10	3.00	N
Solenoid force B $F_{S_{B_i}}$ F1	1.00	30.00	N
Solenoid force B $F_{S_{B_i}}$ F2	0.50	9.50	N
Solenoid stroke A $x_{p_{A_i}}$ X1	0.50	15.00	mm
Solenoid stroke A $x_{p_{A_i}}$ X2	3.00	32.00	mm
Solenoid stroke B $x_{p_{B_i}}$ X1	0.50	15.00	mm
Solenoid stroke B $x_{p_{B_i}}$ X2	2.00	31.00	mm
Solenoid A stiffness λ_p	100000	3000000	N/m
Solenoid B stiffness λ_p	100000	3000000	N/m
Solenoid A damping K_p	0.00	15.00	$N/(m/s)$
Solenoid B damping K_p	0.00	15.00	$N/(m/s)$

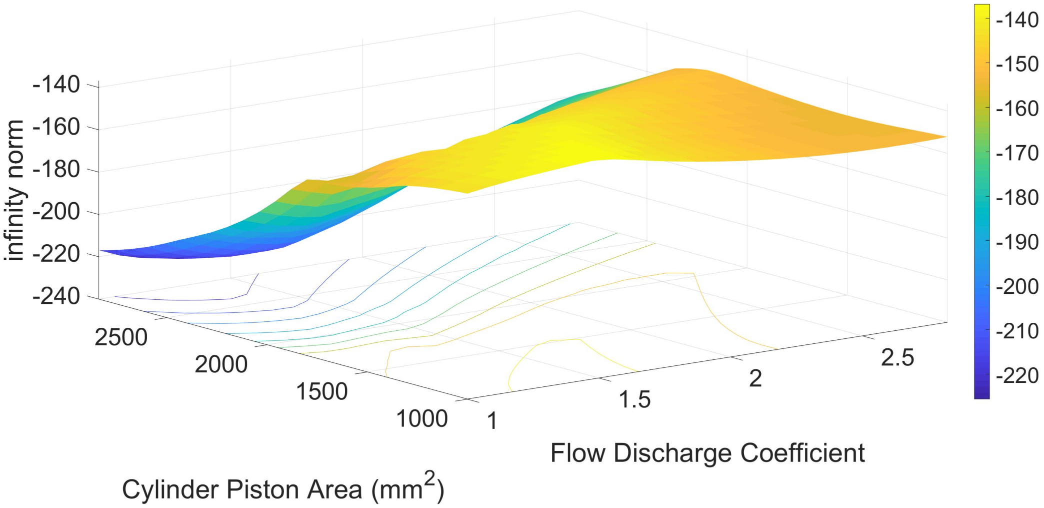


Figure 4.11: *Performance surface showing the infinity norm of the output error as a function of flow discharge coefficient and cylinder piston area of the hydraulic actuator.*

$\bar{J}_N(\boldsymbol{\theta})$ in Equation (4.14). To improve the estimation accuracy of the parameters, the approach taken in determining the global optimum solution is to use the so called multi-objectivization method, see e.g. [120, 121]. In this technique, a single-objective problem is first formulated as a multi-objective optimization problem and, by solving the reformulated problem, it is possible to provide a solution which minimizes the original single-objective cost function. It has been shown by Lochtefeld and Ciarallo, [121], that on average a multi-objective approach outperforms single-objective methods. Another advantage of using multiple segments is that it introduces cross validation in parallel to parameter estimation when the GA is running. For each segment, the estimated parameters are validated against the fitness value for other segments. A good parameter set in the sense of the multi-objective performance index should have a low fitness value across all segments. Since, in the ideal case, there is only one parameter set that ensures the minimum fitness value across all segments, i.e. the global optimum solution, it is expected that the algorithm achieves an improved convergence to the optimum parameter values in this way.

To explain the rationale, consider the following definition, from Marler and Arora [122]. Assume $\mathbf{F}(\boldsymbol{\theta}) \in \mathbb{R}^n$ is a multi-objective function $\boldsymbol{F(\theta)} = [F_1(\boldsymbol{\theta}), F_2(\boldsymbol{\theta}), \ldots, F_n(\boldsymbol{\theta})]$. The point $\mathbf{F}^0 \in \mathbb{R}^n$ is called a utopia point if and only if $F_i^0 = min\{F_i(\boldsymbol{\theta})|\boldsymbol{\theta} \in \Omega_M\}$ for $i = 1, 2, 3, \ldots, n$, where Ω_M is the search space of feasible parameter values. Here, by splitting the measured output into segments, the single objective optimization problem is converted to the minimization of a multi-objective cost function as follows:

$$J_{mul}(\boldsymbol{\theta}) = \left[J^1_{N_1}(\boldsymbol{\theta}), J^2_{N_2}(\boldsymbol{\theta}), \ldots, J^s_{N_s}(\boldsymbol{\theta})\right] \tag{4.17}$$

where $J^s_{N_s}(\boldsymbol{\theta})$ is the objective function for the segment s defined as:

$$J^s_{N_s}(\boldsymbol{\theta}) = \left(\frac{1}{N_s}\sum_{k=1}^{N_s} |\varepsilon^s(k;\boldsymbol{\theta})|^p\right)^{\frac{1}{p}} \qquad p = 1, 2, \infty \tag{4.18}$$

in which $\varepsilon^s(k;\boldsymbol{\theta}) = y^s(k) - y(k;\boldsymbol{\theta})$ and $y^s(k)$ is the measured output at segment s. For the present problem, $\mathbf{F}^0$ corresponds to the value of the objective function $J_{mul}(\boldsymbol{\theta})$ at the desired $\boldsymbol{\theta}_0$ which is unattainable in general. However, using evolutionary algorithms, an approximation of $\mathbf{F}^0$ which is close to $\mathbf{F}^0$ will be achieved. Such a solution is called a compromised solution and is Pareto optimal. The challenge is how the word close is defined for the compromised solution. In the present work, two criteria are considered: $1-$ sum of the square error of the objective function at the solution point with respect to the utopia point; and $2-$ sum of the absolute value of the relative error of the estimated parameters. These two criteria are considered for the three different p-norm types, i.e. $p = 1, 2, \infty$, as shown in Equation (4.18).

After the GA algorithm has completed a specific number of iterations, a Pareto set indicating the best parameters estimating the Pareto-optimal solution is achieved. The solution is achieved by finding the minimum of $F_i(\boldsymbol{\theta})$ for all Pareto set solutions and $i = 1, 2, 3, \ldots, n$. This leads to a best solution for each segment. Subsequently, the average of these n points is taken as the final solution.

4.4 Genetic algorithm

The main challenge in dealing with the optimization problem (eqs. (4.13) and (4.14)) is that the error signal $\varepsilon(k;\boldsymbol{\theta})$ cannot be formulated analytically as a function of the parameter vector $\boldsymbol{\theta}$. This is due to the fact that a closed form representation of the dynamic equations of the 7-DOF manipulator is very mathematically complex. Moreover, the performance index $J_N(\boldsymbol{\theta})$ in equation (4.13) for $p = 1$ and $p = \infty$ is not differentiable. As a result, finding a mathematical expression showing the gradient of the performance surface with respect to the parameter vector $\boldsymbol{\theta}$ is not practical. Hence, in the following, an initial straightforward GA and a more complex, multi-objective GA are described.

4.4.1 Simple genetic algorithm

The GA approach is a very well-known, evolutionary computational global optimisation method, based on the biological principle of natural selection, where the fittest individuals will survive and reproduce. In the GA the individual parameters are encoded as strings of numbers called chromosomes and so, for example, one chromosome will contain a value for each of the parameters being investigated. The process starts by creating a random population of potential solutions which are subsequently evaluated using a fitness function. The initial population is randomly selected from the search space Ω_M with a uniform probability distribution. There are various methods for selection and reproduction [104]. Here parent selection is based on the weighted roulette wheel with replacement i.e. the selection probability of each individual is proportional to its fitness. In the present work , two crossover approaches are compared i.e. uniform crossover and pointwise crossover. Mutation is the final stage, where single elements may be randomly swapped to create a more diverse population. The process is repeated with a new population, until either a minimum fit is reached or a specific number of iterations are passed. Figure 4.12 [123] shows a flow chart of the GA approach used here.

In Section 4.5, each component in Figure 4.12 is investigated to tailor the algorithm

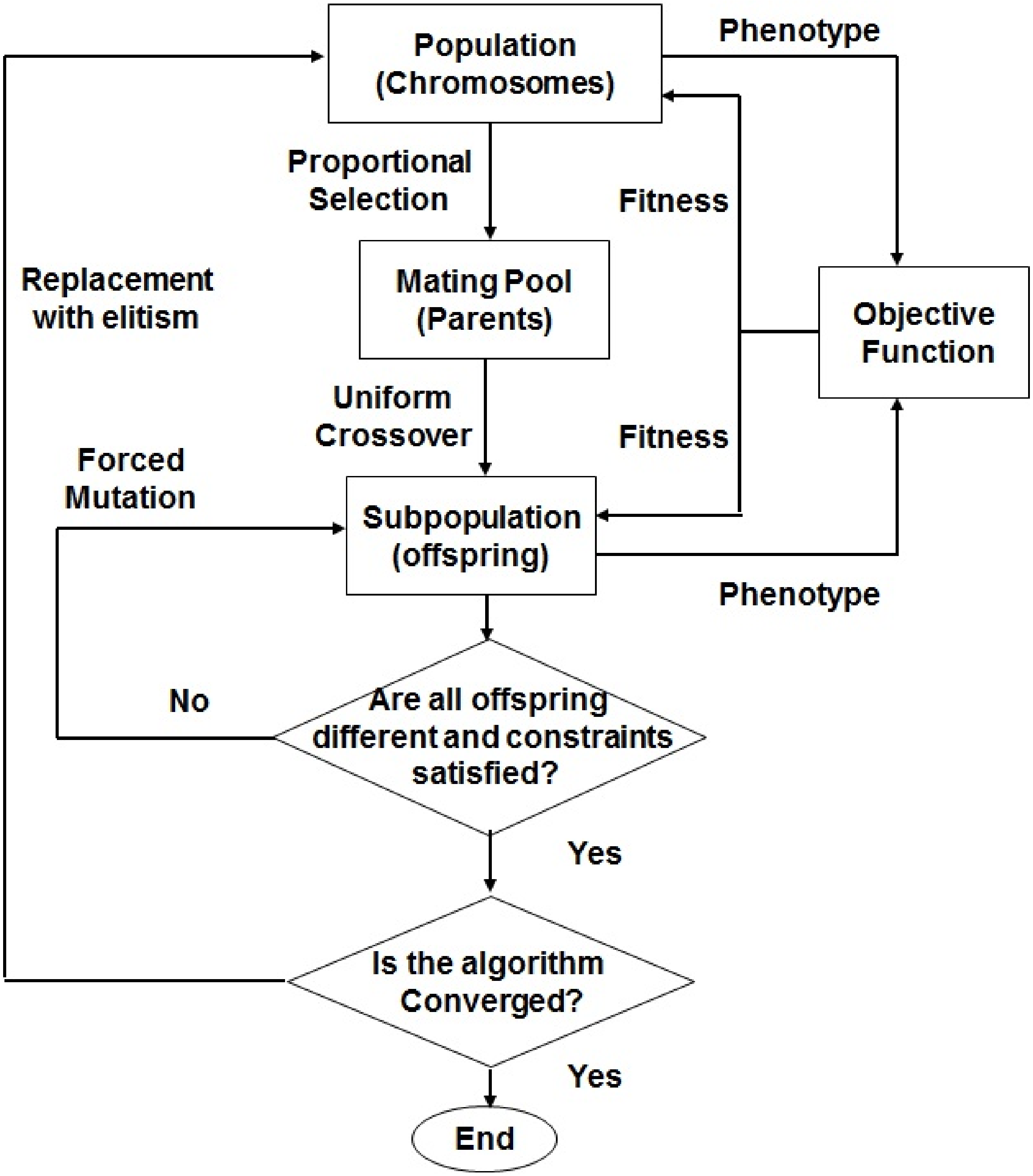


Figure 4.12: *Flowchart showing GA procedure.*

to the specific problem at hand. Two coding schemes are investigated, i.e. integer and multivariable binary string (MVBS). The performance of both coding schemes in terms of finding a better approximation of the optimal solution for the estimation problem is investigated. Using either coding scheme, the chromosomes themselves look the same. For example, equation (4.19) shows the form of the chromosome for optimisation of the mechanical model,

$$\text{Chromosome} = [K, D, S, M_1, M_2, M_3, M_4, M_5, M_6] \tag{4.19}$$

where K is the gain representing the hydraulic subsystem, D is the joint damping coefficient, S is the joint spring stiffness and M_1 to M_6 are the link masses. In the MVBS scheme, each element of the chromosome is represented by a 16 bit binary number, whereas in the integer coding scheme each element is represented by an integer. For each iteration of the algorithm, the simulation model utilises the parameters in that chromosome and the output is compared to experimental data. A fitness value is used to assess the strength of the chromosome, in which the closer the simulation output is to the experimental data, the smaller the fitness value. Selection of the fitness function plays an important role in the convergence behaviour of the proposed GA, hence a number of options are evaluated later in Section 4.5, to determine the one which is most suitable for the present application.

4.4.2 Multi-objective genetic algorithm

Gotmare et al. [109] presents an overview of evolutionary algorithms for system identification and filtering. In the context of multi-objective evolutionary optimization, Marler and Arora [122] suggest that none of these approaches have been applied in a system identification framework. In fact, quite a few attempts are reported to use multi-objective optimization for the trade-off between the order of the model and the quality of the identification. However, the motivation for the present work is quite different. Here the concept of multi-objectivization [121] is used to convert a single-objective output error identification problem into a multi-objective one.

GAs are also used by [124,125] for tuning the model parameters of a 3-DOF serial manipulator powered by electric motors and gearboxes. A multi-objective GA is also used in [126] to estimate and tune the parameters of a PID controller for a 2-DOF robot arm. However, other options include replacing the GA with particle swarm optimisation (PSO) or differential evolution (DE).

Key challenges associated with the initial single-objective optimisation problem are:

1. Avoiding local optima.

2. Keeping the diversity of the population at a reasonable level.

3. Ensuring the algorithm identifies and keeps good solutions.

To address these a multi-objective GA is implemented. Using the concept of multi-objectivization, the single objective problem (eqs. (4.13) and (4.14)) is converted to a multi-objective optimization problem by splitting the measured data into a number of segments and defining a similar objective function for each segment. The developed algorithm utilises the well-known NSGA II based on non-dominant sorting and Pareto optimal solutions. As noted by Lochtefeld & Ciarallo [121], the majority of multi-objective evolutionary algorithms, especially those used for multi-objectivization, share a more or less similar algorithmic framework to that of the non-dominated sorting GA and, in engineering applications, no single approach is always superior [106]. The same argument is true for application of evolutionary algorithms in system identification and filter design [109].

Selection of a specific method depends on the type of information provided in the problem, the solution requirement and user preference. The main purpose of the multi-objective GA is to evaluate and order the offspring in a more effective way, with the aim to improve the performance in regard to the convergence of the estimated parameters to the true values.

4.5 Performance optimisation

Further to the fitness function and the coding scheme explained earlier, the performance of the GA in finding the best estimation result for parameters of the developed model is heavily influenced by several other factors, including the crossover rate, crossover type, mutation rate and population length. Hence, it is necessary to train the simple and multi-objective GA with the most appropriate settings for the present identification problem.

For the analysis in this section, the estimated parameters are compared with a set of illustrative parameter values, i.e. numerical values that are physically realistic but are not yet optimised for the real device. These values are listed in the first few rows of Table 4.2, and are used in the model to generate simulation data for the simple and multi-objective GA optimisation (i.e. as a surrogate for experimental data but with known parameter values). For this purpose, the mechanical subsystem is initially considered in isolation using a gain K to represent the hydraulics.

This approach allows us to evaluate how close the estimated values of the parameters are to the true values already set in the mechanistic model. It also makes clear under which circumstances the algorithm has the capability to converge to these real parameter values, rather than generating a solution based on a different local minimum within the global search space. The latter can happen when the fitness value is relatively low, indicating a close match between the simulated and experimental (here also simulated) output response, but for which the estimated parameters remain far from the true values. The following sections initially focus on the simple GA with MVBS coding, before considering the integer coding scheme and the multi-objective GA.

4.5.1 Evaluation of different crossover types

Initially using MVBS, pointwise and uniform crossover are investigated. For the pointwise crossover scheme, chromosomes are broken into several segments and each two-parent chromosome swaps segments between the points. Uniform crossover works by swapping every other gene of one of the parent chromosomes with the other parent, so each child ends up with 50 percent of each parent chromosome.After testing both approaches the conclusion is that using the MVBS coding scheme, for this particular application, the pointwise approach consistently yields significantly improved performance compared to uniform crossover.

Table 4.2: *Initial versus true (simulation) parameters values for the mechanical model, and relative error of the estimated parameters (i.e. difference between the true and estimated values, divided by true value) for selected fitness functions using the simple GA.*

				Mass of link (kg)					
	Gain	Damping	Stiffness	1	2	3	4	5	6
Parameter	**Value**								
Initial	0.1	100	20	0.6	1.3	2.8	1.2	2.2	1.7
True	0.2	231.8	53.4	4.6	5.1	22.5	1.6	4.1	4.7
Fitness Function	**Relative Error**								
Equation (4.14) with $p = \infty$	0.5	0.26	0.42	0.19	0.05	0.43	0.12	0.22	1.06
Equation (4.14) with $p = 1$	1.0	0.82	0.43	0.84	0.94	0.96	2.50	1.46	0.72
Equation (4.14) with $p = 2$	0.5	0.57	0.29	0.23	0.09	0.84	0.93	0.19	0.21
Equation (4.15) with $p = \infty$	0.5	0.32	0.47	0.23	0.80	0.78	3.93	0.24	0.12

4.5.2 Evaluation of different fitness functions

One of the most important issues in mathematical optimization is selection of the fitness function. Selection of a suitable fitness function enables proper discrimination of the individuals using the proposed GA-based method. Of the six possible fitness functions introduced previously, two are immediately found to yield rather erratic performance. Hence, only four options are considered in more detail. These are based on Equation (4.13) for $p = 1, 2, \infty$ and Equation (4.14) for $p = \infty$. Figure 4.13 illustrates the simple GA performance for the MVBS coding scheme using each of these fitness functions. Table 4.2 shows the relative errors of the optimised parameter values using these different fitness functions and Table 4.3 the lowest fitness values and error indices.

These results demonstrate that the convergence of the estimated parameters to their true values requires particular attention in the present context. For this example, Figure 4.13 shows that norm two (the Euclidian norm) reaches the final fitness value with the least number of iterations searched. Although norm two has the fastest convergence rate, it is evident from Table 4.3 that it has a relatively high fitness value, whereas the two infinity norm fitness functions yield very low fitness values. The infinity norm (eq. (4.13)) results in both the lowest output identification error and lowest parameter estimation error, and so is the most obvious choice for further evaluation below.

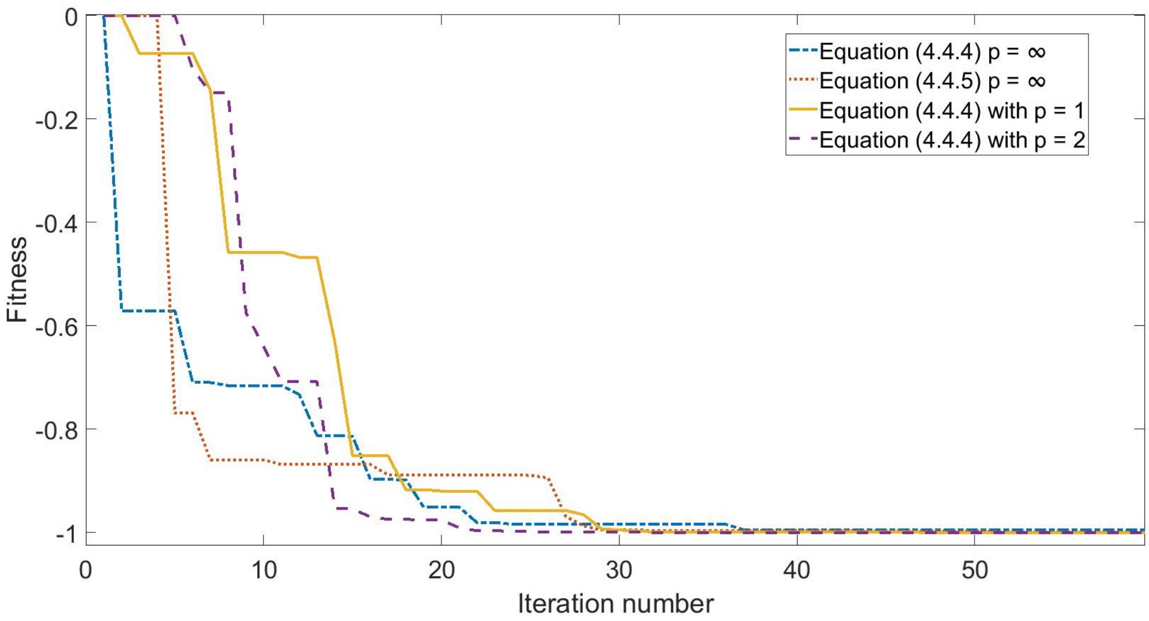


Figure 4.13: *Comparison of the learning curves of four different fitness functions (normalized between 0 and 1) for the simple GA using the MVBS coding scheme.*

4.5.3 Evaluation of population, crossover and mutation rates

For the present application, population size, crossover and mutation rates are found to have a significant effect on the speed of convergence and final achievable value for the fitness function. Comparing various population sizes, the simulation study suggests that a population size of ~ 70 should be utilised since it generates a diverse population and has the capability to reach a lower value for the performance index.

With regard to crossover, a value of 1 means that, at every iteration, the parent chromosomes will create new child chromosomes, hence every iteration will contain different chromosomes to the one before. In Figure 4.14, for example, crossover values of 0.6 and 0.8 take almost the same time to converge while, despite 0.8 having a higher initial value of fitness, it converges to the lowest value. These results suggest

Table 4.3: *Comparison of each fitness function associated with the simple GA optimisation shown in Table 4.2. The output error indices are based on the infinity-norm of the difference between the simulated and optimised dynamic response. The parameter error indices are calculated as the sum of absolute value of relative estimation errors for the set of nine mechanical model parameters.*

Fitness Function	**Lowest Fitness Value**	**Output Error Index**	**Parameter Error Index**
Equation (4.14) with $p = \infty$	-0.1275	0.651	743
Equation (4.13) with $p = \infty$	-0.6458	0.645	327
Equation (4.13) with $p = 1$	-836.54	0.850	969
Equation (4.13) with $p = 2$	-14.5172	0.82	389

that a crossover value of 0.8 tends to yield the most promising performance. This crossover value represents an empirically derived balance between the requirement for a diversity of population and the need to force the output to determine the fittest individuals. Similar exercises are used to determine the mutation rate and the study shows that the value 0.05 yields the best result in terms of a compromise between the diversity of the population and the pressure on the selection of the highest fitness values. Hence, the best tune for the binary coding scheme is listed in the middle column of Table 4.4.

4.5.4 Evaluation of different coding schemes

A similar study (i.e. changing one parameter at a time and comparing the learning curves) is carried out for the simple GA with integer coding. In this case only a uniform crossover is considered. The previous results suggest that the best tune for the parameters of the simple GA with integer coding is population size 70, crossover rate 0.8, and mutation rate 0.5. Finally, the performance of the algorithm for the integer and MVBS coding schemes are compared. With the MVBS scheme, the present anal-

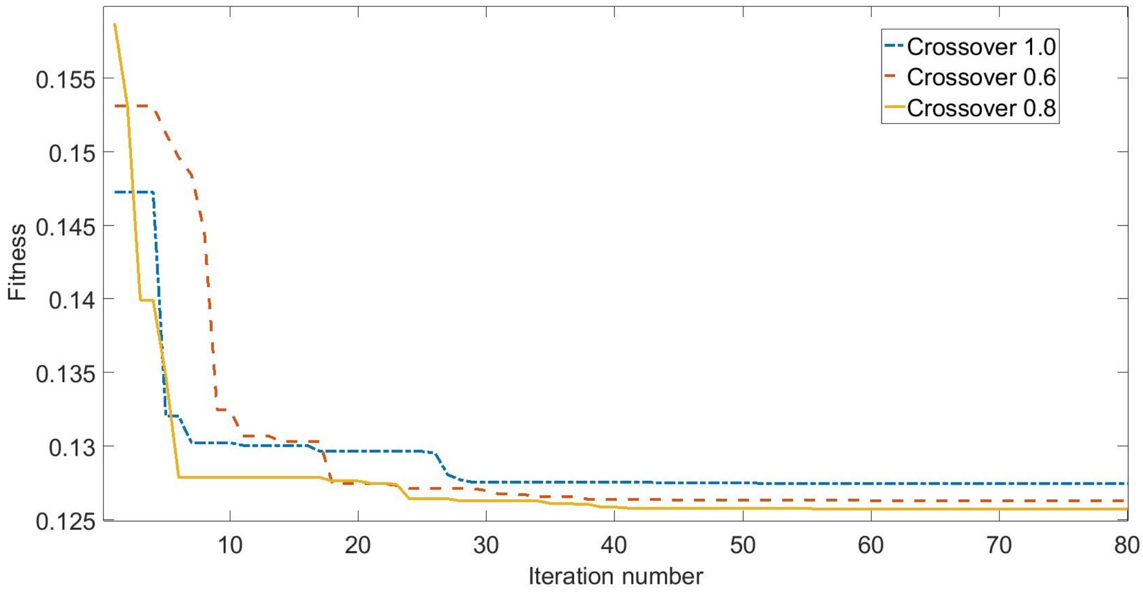


Figure 4.14: *Fitness comparison for three different crossover rates for MVBS coding scheme.*

ysis uses grey code to prevent sudden erratic jumps in the parameters when crossover occurs, allowing the algorithm to explore the whole search space more effectively. Running the simple GA for both coding schemes and with the attained best tune of the parameters, shows that the MVBS scheme consistently performs better for this robotic manipulator identification problem. The learning curves of the simple GA with the best tune for both MVBS and integer coding schemes are compared in Figure 4.15. In all cases, the learning curves are plotted after taking the average over 10 runs.

4.5.5 Evaluating the parameters of the multi-objective GA

Although the simple GA developed in the previous section forms the reproduction engine for the multi-objective optimization algorithm, the best parameters for this

Table 4.4: *GA settings used to generate the parameter estimation results.*

Setting	**Simple GA value**	**Multi-Objective GA value**
Coding scheme	Multivariable binary coding	Multivariable binary coding
Crossover rate (Pc)	0.8	0.4
Mutation rate (Pm)	0.05	0.2
Parent selection	proportional	proportional
Crossover type	pointwise	pointwise
Population size	70	20
Fitness function	infinity norm	infinity norm

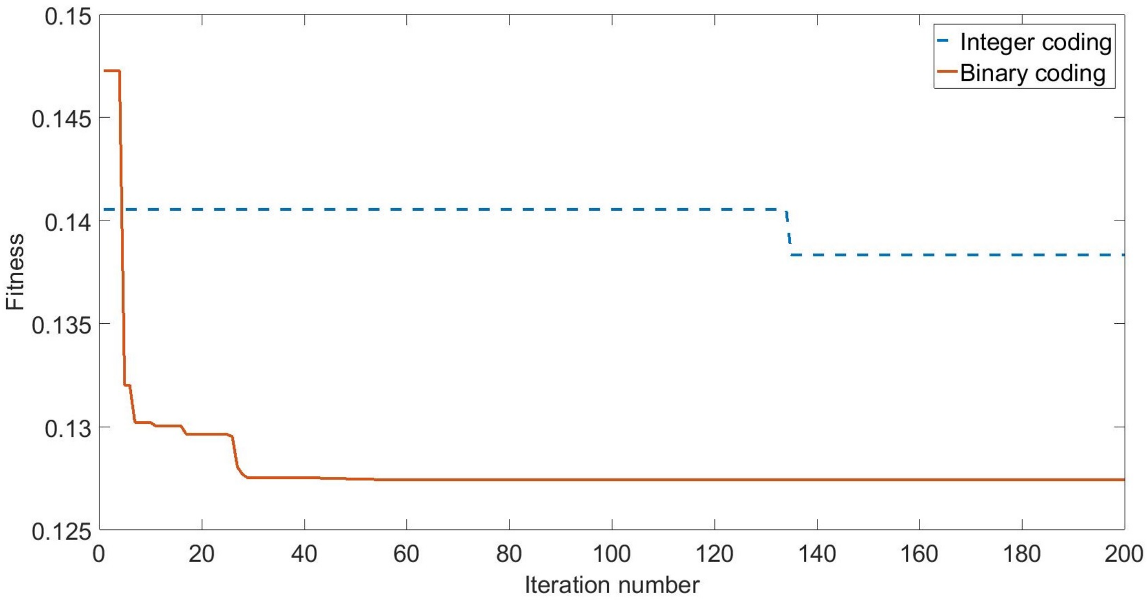


Figure 4.15: *Fitness comparison for binary and integer coding schemes.*

algorithm, nonetheless, require further study. Selected results from the systematic study into this issue are presented in Tables 4.5 and 4.6, while the final column in Table 4.4 summarises the conclusions. For this purpose, the crossover, mutation, and population size of the multi-objective GA are all varied and the sum of relative parameter estimation error, as well as the sum of square errors, are calculated for each parameter. The accuracy of the estimated parameters shown in Table 4.5 is listed for selected crossover values, whilst the sum of square error and output error index are listed in Table 4.6 for selected crossover, mutation, and population values. The results in Table 4.5 show that the crossover values 0.8 and 0.4 both give acceptable results in terms of the estimation accuracy.

However, from Table 4.6 it can be inferred that the crossover values 0.6 and 0.4 yield the best results in terms of the mean squared error. This necessitates reaching a compromise for the parameters of the multi-objective GA in terms of the parameter estimation accuracy and reaching the minimum value for the sum of square error. Comparing the index values listed in Tables 4.5 and 4.6 suggests that the crossover value of 0.8 is the right choice for further study in the present context. Following a similar analysis, a mutation rate 0.2 and population size 70 is found to yield the best performance for the proposed multi-objective GA in order to achieve better estimation accuracy for the parameters.

Finally, to evaluate how the number of segments in the multi-objectivization process affects the performance of the multi-objective GA, the parameter estimation problem is solved for different numbers of segments. In particular, Tables 4.6 and 4.7 compare use of two and eight segments, and show that both the mean square error of the output and the parameter estimation accuracy are significantly improved by having eight objective functions. In fact, use of eight segments is a pragmatic choice based on a visual examination of the experimental time series.

Table 4.5: *Evaluating the affect of crossover on the performance of the multi-objective GA.*

				Mass of link (kg)						
	Gain	**Damping**	**Stiffness**	1	2	3	4	5	6	
Crossover				**Relative Error**						**Sum**
1.0	0.80	0.36	0.27	0.35	0.02	0.28	2.60	0.78	0.48	**5.94**
0.8	0.05	0.08	0.15	0.22	0.77	0.02	4.05	2.81	0.31	**8.46**
0.6	0.35	0.26	0.12	0.28	1.03	0.10	4.70	3.05	1.83	**11.72**
0.4	0.05	0.08	0.15	0.21	0.77	0.03	4.05	2.81	0.31	**8.46**

Table 4.6: *Comparison of the mean squared and output error indices for various multi-objective GA settings.*

Population	**Crossover**	**Mutation**	**Mean squared error**	**Output error index**
			8 Segments	
70	0.8	0.05	0.3133	1.322
50	0.8	0.05	11.2987	7.862
20	0.8	0.05	0.2283	1.2141
70	1.0	0.05	0.2217	0.9240
70	0.6	0.05	0.0580	1.2141
70	0.4	0.05	0.08636	1.2157
70	0.8	0.1	0.0693	1.2148
70	0.8	0.2	0.0945	1.2151
			2 Segments	
70	0.8	0.05	0.1297	1.214

Table 4.7: *Evaluating the affect of the number of segments on the performance of the multi-objective GA.*

				Mass of link (kg)						
	Gain	**Damping**	**Stiffness**	1	2	3	4	5	6	
True value	0.2	231.8	53.4	4.6	5.1	22.5	1.6	4.0	4.7	
Simple GA estimates	0.4	343.2	89.4	4.6	10.6	0.8	10.0	19.5	5.9	
Multi-objective 8 segments average	0.4	343.2	89.4	4.6	10.6	0.8	10.0	19.5	5.9	
			Relative Error							**Sum**
Simple GA	1.0	0.84	0.67	0	1.08	0.96	5.25	3.76	0.26	13.46
Multi-objective 2 segments	0.9	0.86	0.48	0.28	0.77	0.01	4.97	2.66	1.82	12.75
Multi-objective 8 segments	0.5	0.52	0.29	0.50	0.55	0.41	2.19	0.46	0.04	5.46

4.6 Results for mechanical model only

The present section considers the mechanical model only. In the first instance, the optimisation performance of the simple and multi-objective GA's are compared in simulation (i.e. to estimate assumed 'known' parameters). This is followed by their application to experimental data i.e. to estimate the actual unknown parameters of the mechanical subsystem for each joint.

4.6.1 Testing with simulated data

The values in Table 4.4 are utilised. Figure 4.16 compares the dynamic response of the simple GA and multi-objective GA optimised model with the simulated data for Joint 2. Simulated data is used initially in place of experimental data, so that the parameters are known and the performance of the parameter estimation can be evaluated. The voltage input used for this simulation experiment, illustrated in Figure 4.17, is based on the laboratory experiments considered later in this thesis and represents a realistic input signal.

As Figure 4.16 shows, both the simple GA and multi-objective GA estimated simulation model output tracks the original simulation output rather well, albeit with a small over shoot at the peaks. However, Table 4.7 shows that some of the simple GA parameter estimates are significantly distant from their true (simulated) values, motivating the use of the multi-objective GA approach. To implement the latter, the output is split into eight segments, hence eight objectives to minimise,The use of eight segments was found to work well in practice. It matches the number of peaks and troughs in the example time series (when the joint angle was at the greatest deviation from the initial reference point) and is compatible with the sensitivity analysis reported earlier (e.g. Figure 4.7).

In the case of the multi-objective GA, after completion of a specific number of iterations, a Pareto set indicating the best parameters estimating the Pareto-optimal solution is achieved. The final value of the estimated parameters $\hat{\boldsymbol{\theta}}$ is achieved by finding an estimation of the Pareto-optimal solution that minimizes each of the eight objective functions, and subsequently determining the average. The parameter vector is utilised to generate a new output response, as illustrated in Figure 4.16 The average optimised response follows the simulation data very well for most of the experiment.

Table 4.7 compares the simple and multi-objective GA optimisation results. For this example, it can be seen that the multi-objective GA with eight segments generally yields more accurate parameter estimates. In fact, the sum of the relative error for the simple and multi-objective GA with both two and eight segments are 13.46, 12.75, and 5.46 respectively. This demonstrates significant improvement in the estimation accuracy of the parameters for the multi-objective GA with eight segments compared to the others.

4.6.2 Testing with experimental data

The simple and multi-objective GAs are applied to experimental data collected from the manipulator, as illustrated in Figure 4.18. For direct comparison with the simulation benchmark examples above, the shoulder joint is again chosen to demonstrate the feasibility of the GA approach. The experimental data is collected using a sim-

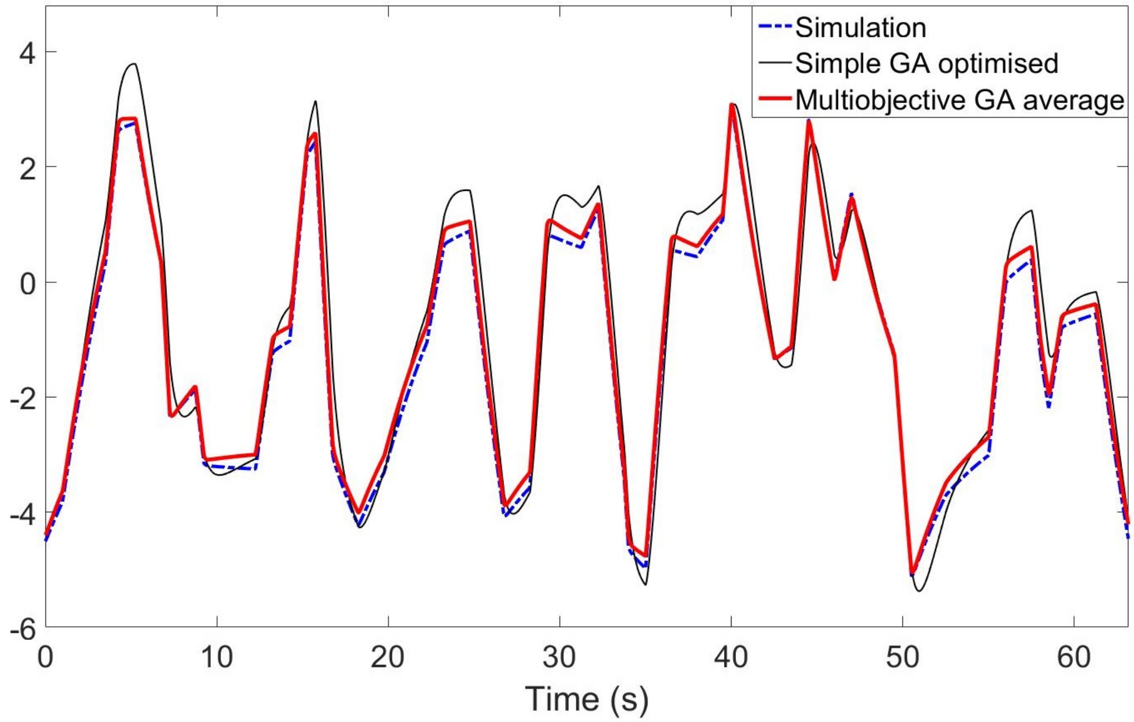


Figure 4.16: *Simulation response of the mechanical subsystem model using the simulated 'true' output (dashed trace), simple GA (thin trace), and multi-objective GA estimated parameters.*

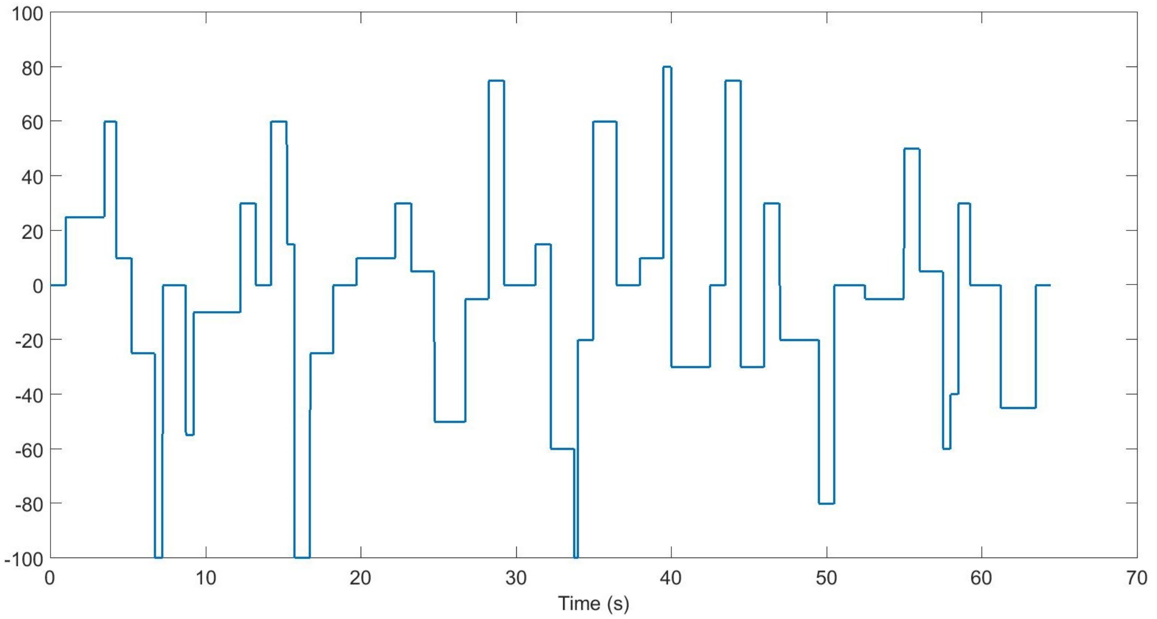


Figure 4.17: *Input used to generate both the simulated and experimental data.*

ple open loop technique, where either step or pseudo-random input signals can be used. The signals are generated using a Matlab script, and are treated as a controller output signal. An example of a pseudo-random signal is given in Figure 4.17, this signal would then be passed through the calibration before being split into two voltage signals and sent to the corresponding channel on the CFP.

Table 4.8 shows the mechanical model parameter estimates returned by both the simple GA and multi-objective GA. Figure 4.18 shows that the optimised simulation output for both algorithms generally follows the dynamic behaviour of the device, although there are some significant underestimates of the joint angle in the case of the multi-objective GA. This may be a result of using a constant gain value instead of the hydraulic sub system, which is non linear.

In regard to the parameter estimates, the results for the mechanical subsystem can be partially validated by consideration of the known total mass of the manipulator i.e. $45kg$. In this regard, the sum of the six estimated link masses in Table 4.8 is $43.9kg$ and $79.7kg$ for the multi-objective and simple GA respectively. This result supports the earlier conclusions of the simulation study, i.e. the multi-objective GA provides closer estimates to the true parameter values than the simple GA.

Finally, Table 4.9 shows that the parameters optimised using the laboratory data in Figure 4.18 also yield a satisfactory output response when they are applied to new experimental data in a simple validation exercise. Here, two different input signals are utilised to generate simulation data (with the same set of parameters as optimised above) and the joint angle responses in each case are compared with the equivalent experimental data collected from the real machine.

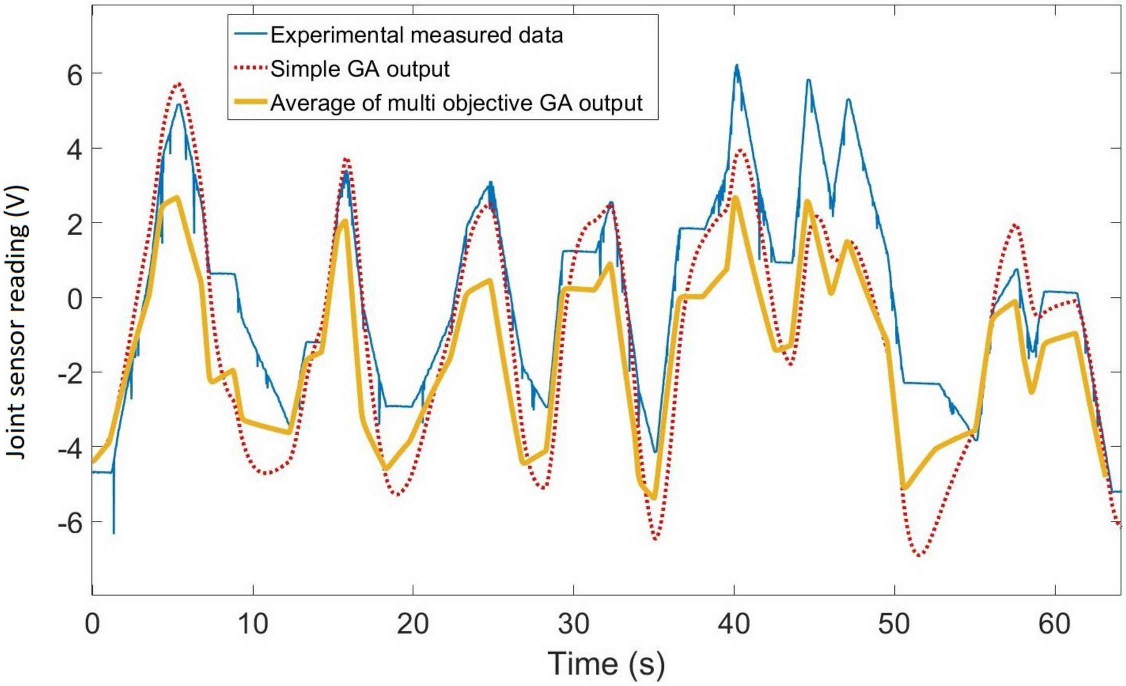


Figure 4.18: *Experimental data from the manipulator, compared with the simulation response for the model optimised using the simple (dashed) and multi-objective (thick) GA.*

Table 4.8: *Estimated parameters returned by the simple and multi-objective GA for the experimental data for the mechanical model subsystem.*

				Mass of link (kg)					
	Gain	Damping	Stiffness	1	2	3	4	5	6
Simple GA estimates	0.1441	90.52	26.34	1.11	7.80	27.73	10.37	21.86	10.82
Multi-objective 8 segment average	0.3581	314.27	39.06	2.97	4.99	16.08	5.78	7.12	6.99

Table 4.9: *Output error indices associated with the optimised model applied to three different laboratory experiments, i.e. based on different input sequences. Only the first data set is used for GA optimisation hence the other two experiments represent test validation experiments. The output error indices are based on the mean absolute error between the simulated and optimised dynamic response.*

Data Set	**Multi-Objective**	**Simple GA**
1	1.7528	1.4301
2	2.5661	2.5414
3	1.3659	1.7591

4.7 Results for full model

Following the changes made to the hydraulic system model discussed in section 4.2.4, parameter estimation could be carried out, focussing on the sixteen parameters identified in Section 4.3, and setting the mechanical parameters to the values already estimated, as shown in Table 4.8. The process for this was the same as for the mechanical subsystem, just modifying the chromosomes used in the GA to be for the hydraulic model parameters rather than the mechanical subsystem. As it was known that the GA approach worked, testing went straight to using real experimental data rather than simulated data.

Table 4.10 shows the estimated values of the previously identified key hydraulic parameters for each joint. For the first three joints these values are all similar, and seem physically realistic without knowing the true real values. This is expected as the first three joints all use similarly sized hydraulic actuators. Joint 4 is the only rotary joint, and as such has a noticeably different value for piston area (note the area is not strictly piston area, being a rotary joint, but for ease of presentation the terminology is kept consistent across all joints).

Figures 4.19 to 4.23 show a comparison of the measured experimental data from the manipulator with the model output for each joint, following a series of step inputs covering the entire range of motion for each joint. The input was designed to be challenging but realistic, with varying step sizes in both directions, and to cover the entire range of motion of the joint. An existing controller was used to both control the real manipulator and generate the voltage input for the model so that the same input was used for both the model and real system. The idea at this stage is just to match the behaviours not optimise set point following.

The results shown in Figures 4.19 to 4.23, although not perfect, are promising and show that both the GA approach for parameter estimation is working, and that the newly developed full system model is working and representative of the real system. The results show that in some areas the model is particularly strong, such as the time delay. As can be seen particularly clearly in Figures 4.20 and 4.21, the time taken for movement as a response to the input is almost identical between the model and measured data. The area where the model is weaker currently is in matching amplitude change in response to the step input, in most cases causing the model to give a higher or lower joint angle output than the measured data. For several of the joints this difference in output seems particularly prevalent at the limits of the range of motion, for example Joint 2 (Figure 4.20), the model tracks the experimental data well in the mid range of movement but is poor at the upper and lower limits. The results are particularly poor for Joint 5, Figure 4.23, which is not a surprise as this joint has hardware faults causing erratic behaviour, such as the drifting that can be clearly seen in the experimental measured data in Figure 4.23.

In future research, expanding the GA to include other less dominant parameters would likely help with improving on these results. Other potential methods of improving these results would be to retune the GA parameters, such as the crossover and mutation rates, with the higher number of parameters, as this may have an affect on the GA performance. Nonetheless, the present version of the model yields reasonably realistic responses, suitable for preliminary development and simulation work of assisted teleoperation. Furthermore, whilst the first goal of the chapter was to develop the modelling equations and implement these in MATLAB, the second goal was to investigate the potential use of GAs for this type of parameter estimation problem, and this has been demonstrated.

Table 4.10: *Hydraulic system parameter estimates.*

Parameter	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5
Piston area A	450.010	416.370	251.920	9.488e-09	300
Piston area B	399.940	107.880	157.370	1.060e-08	400
Solenoid current I_{max_A}	0.034	0.032	0.035	0.032	0.070
Solenoid current I_{max_B}	0.060	0.048	0.079	0.079	0.070
Solenoid force F_{S_A1}	5.626	5.627	1.361	1.857	5.800
Solenoid force F_{S_A2}	3.789	3.788	0.642	0.857	3.750
Solenoid force F_{S_B1}	1.664	1.663	2.341	1.793	1.500
Solenoid force F_{S_B2}	0.793	0.793	0.666	0.796	0.790
Solenoid stroke x_{p_A1}	0.345	0.344	0.435	0.449	0.344
Solenoid stroke x_{p_A2}	0.369	0.369	0.461	0.499	0.365
Solenoid stroke x_{p_B1}	0.452	0.451	0.443	0.452	1.500
Solenoid stroke x_{p_B2}	0.4834	0.484	0.469	0.484	0.793
Solenoid stiffness λ_{p_A}	5e+05	5.47e+05	1.58e+06	5.47e+05	5.47e+05
Solenoid stiffness λ_{p_B}	1.56e+06	1.58e+06	1.58e+06	1.58e+06	5.47e+05
Solenoid damping K_{p_A}	100	120.177	16.467	120.177	120.177
Solenoid damping K_{p_B}	1000	17.178	16.467	17.177	17.178

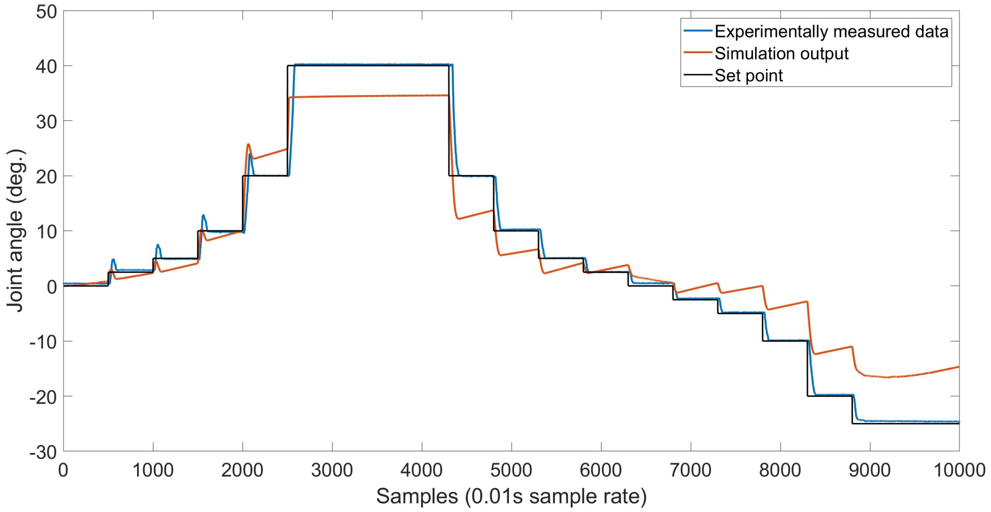


Figure 4.19: *Model response and experimental measured data for set point following across range of motion, for Joint 1 of the manipulator.*

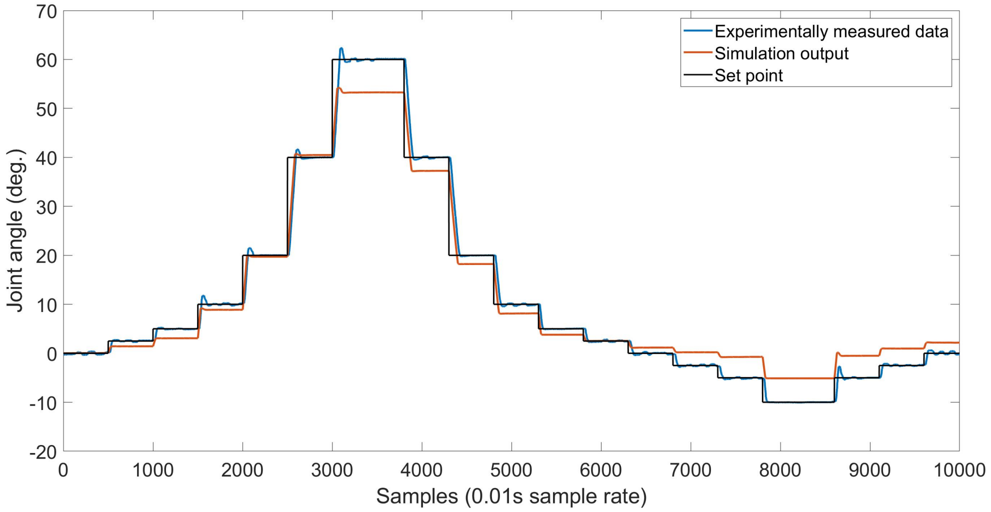


Figure 4.20: *Model response and experimental measured data for set point following across range of motion, for Joint 2 of the manipulator.*

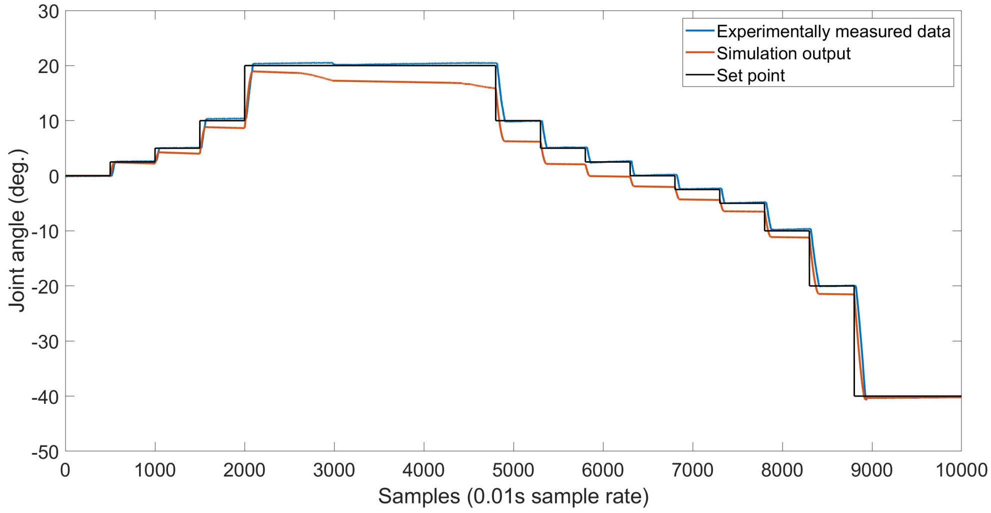


Figure 4.21: *Model response and experimental measured data for set point following across range of motion, for Joint 3 of the manipulator.*

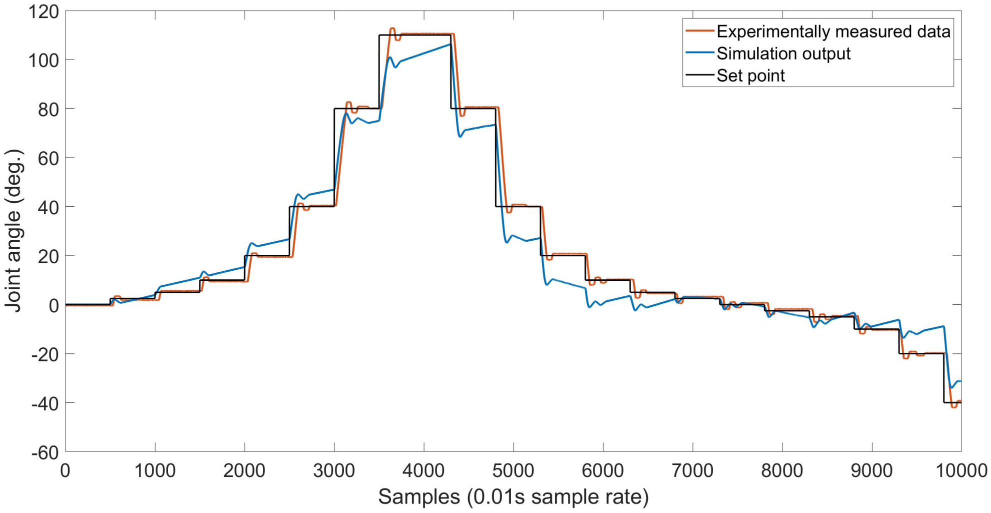


Figure 4.22: *Model response and experimental measured data for set point following across range of motion, for Joint 4 of the manipulator.*

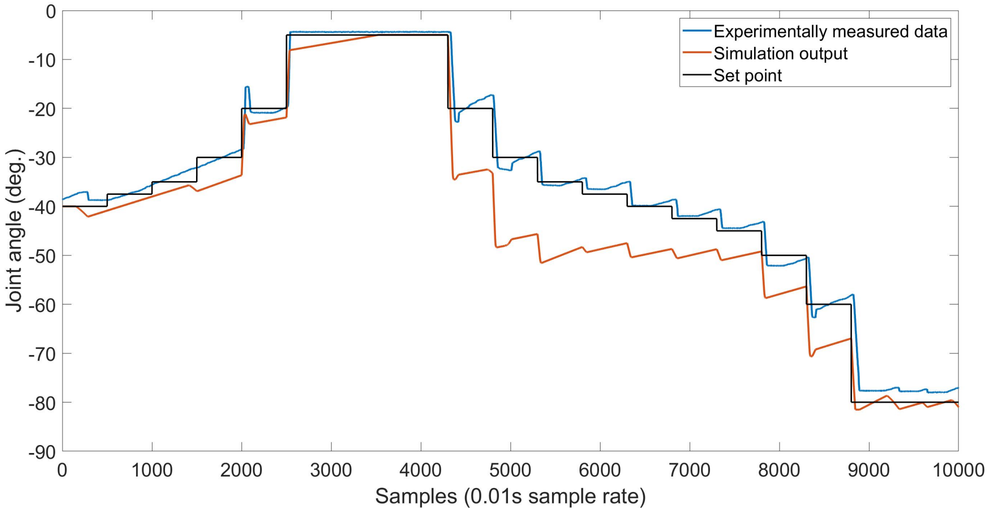


Figure 4.23: *Model response and experimental measured data for set point following across range of motion, for Joint 5 of the manipulator.*

4.8 Concluding Remarks

A comprehensive mechanistic model of the hydraulic manipulator has been developed and implemented for the MATLAB SIMULINK software environment. Different GAs for parameter estimation were tuned and compared, with the overall aim of both determining the unknown parameter values of the system, and ensuring that the estimated model matched the real system performance. The optimum settings for the GA were identified for this particular problem, with a simple GA and a multi objective GA compared. These results show that the multi objective GA outperformed the simple GA, both in yielding accurate parameter values and in matching the real world dynamic behaviour of the manipulators.

Initially, a simplified version of the mechanistic model was utilised for the research into GA optimisation, one that only modelled the mechanical aspects of the manipulator. This approach simplified the problem, reducing the number of parameters involved. Once the GA approach was proven to work with the mechanical model, it was extended to the detailed full model. These results are also promising, with the model generally providing a good match to the experimental data collected from the real manipulator.

Chapter 5

Assisted Teleoperation Development

Whilst earlier research using the Brokk–Hydrolek had been limited to either open-loop experiments, or closed-loop control for which the end-effector follows a previously determined trajectory, the present chapter presents the new assisted teleoperation system. Background information and existing technological developments in regard to robotic vision and the human-machine interface are discussed in section 5.1. This is followed in sections 5.2 and 5.3 by a description of the assisted teleoperation system and associated kinematic equations respectively, and experimental testing in the laboratory in section 5.4.

5.1 Vision systems and user–interface in robotics

Vision systems are widely used in robotics, in both teleoperated and autonomous systems. Commonly vision systems used with robotic manipulators involve cameras fixed in place, to give multiple views of the workplace. This approach is not practical for a mobile nuclear decommissioning system as it would need people to enter the dangerous work environment and install cameras in precise locations. As such, methods that involved cameras mounted to the mobile platform using manipulators were investigated. Again these often involve multiple cameras, for example one fixed on the body and one on the end effector. This gives a better view of the work place and makes control simpler.

In a nuclear environment where radiation can damage sensors and electronics, there are two different directions that could be taken. The first is to design for redundancy, i.e. having multiple sensors that can act as backups. The issue with this approach is that it is expensive, and as all the sensors are in the same location they will likely be damaged at an equal rate. The second option is to minimise sensors, and to make them easily replaceable. This was the option chosen in this work, and the reason for choosing to use a single camera, i.e. one that can be easily swapped in case of damage. For pragmatic reasons, the Microsoft Kinect is chosen for research and development purposes, in part because of the readily available image processing algorithms associated with it.

The following subsections look at vision systems in general and operator feedback methods, as well at the Microsoft Kinect system.

5.1.1 Vision in robotics

Robot vision and sensing is a key part of both automated and tele-operated robots. The ability to both see and understand what is around opens up a lot of possibilities and technologies using vision, such as object identification and object tracking, which can allow automation of many tasks. For example, Kehoe et al. [127] use object identification technology combined with pose estimation technology to allow a dual arm robot to sort items. They look at using the technologies to allow the robot to pick up and organise household objects. The system is able to identify what the object is, then what pose it is oriented in, so that it can pick it up and place it in the right location.

The University of Birmingham has carried out substantial work using vision systems with manipulators, with a particular focus on grasping and manipulating objects. As an example of recent work with a nuclear focus, Talha et al. [128] investigate the current technology of tele-operating systems, with multiple buttons and joysticks, and with multiple cameras giving different views of the workspace. The results confirm that the current method is extremely challenging, especially for inexperienced users; this article concludes that current systems will not allow the complex tasks needed in decommissioning to be completed. Marturi et al. [48] compares remote tele-operation with a semi autonomous system using visual servoing, with the results showing the semi autonomous system improves task performance and reduces operator workload.

5.1.2 Operator feedback and user interface

To allow effective control the operator needs to receive some form of feedback from the robot so they know where it is, and the state of the work environment. For example, with a robot entering a nuclear power station part way through decommissioning, the operator may not know exactly where they are going or what to expect, and so relies on the feedback from the robot to navigate safely. Likewise if a robot is operating in close proximity to other robots or people, the operator needs to know what is going on around it to allow safe operation.

The simplest way to do this is to have a camera mounted to the front of the robot so the user can see what is in front of the robot, however this is very limiting. Humans dont purely rely on looking straight ahead to know what is going on around them, they rely on their other senses such as hearing and feeling the temperature. Due to this, to allow immersive and effective teleoperation, often more sensors are desirable. Feedback from these sensors needs to be given to the user in an effective and usable way to allow them to be beneficial and not just overwhelm the user with information.

Sensing and measuring lots of data isn't any use if there isn't an effective way of communicating it back to the operator. Giving the user all the raw data isn't always helpful and can actually make the operation more difficult. A good user interface makes a big difference to the usability of a robot manipulator; a screen full of text and numbers is harder to comprehend than images and graphical representations of data.

There are several ways of providing feedback to the operator, most commonly:

displaying images, videos or words on a screen [129]; playing audio, either directly replaying the sounds recorded near the manipulator [130], or giving audio cues for example beeping if the manipulator is moving towards an object and might collide; and haptic feedback, for example vibrations through a joystick [129].

When displaying images on a screen for the operator, thought needs to be put into the design of the graphical user interface, GUI. For example screen layout, use of colours, clarity of images, and fonts used all need to be carefully thought about [131]. When choosing colours there are standard conventions, such as green meaning go, red being stop or error, and yellow being a warning, which are intuitive even to inexperienced users. These colour conventions are used amongst many interfaces and so are common to most people. Layout conventions are also common across many interfaces, and this is a whole subject of research on its own.

Haptic feedback can be used at different levels, from the user wearing a device that provides force or vibrations to the fingers to mimic what a robot manipulator is touching [132,133], to just giving vibration through a controller [55,129]. With control that uses motion tracking, haptic feedback may be more difficult to deliver than traditional control methods such as joysticks, however it is still a valuable feedback method that greatly increases the sense of user immersion.

5.1.3 The Microsoft Kinect

The release of the Microsoft Kinect, a low cost camera capable of depth sensing and generating 3D point cloud data, helped encourage a lot of research and development into gesture recognition and markerless motion tracking. The original Kinect worked by projecting a pattern of infrared dots; using an infrared camera the deformation of the pattern was measured and used to establish a depth map. The technology, called Light Coding, was developed by a company called PrimeSense, which has since been bought out by Apple. Each Kinect was calibrated during manufacturing by placing it a set distance from a flat wall, and using that known pattern of the infrared dots as a reference point [134].

A lot of work using the Kinect has focused on its ability to track a users' skeletal position, for example work by NASA [57] uses a Kinect to track a users' hand position and then move a robot arm to mimic the users movement. Another example of using a Kinect to track the full skeleton is the project by Wasielica et al. [135], which takes the users' skeletal position to control the joint positions of a small humanoid robot. The robot combines the data from the Kinect with self-balancing and collision avoidance algorithms to mimic the users movements whilst staying upright and not having any of its limbs collide. The advantage of using the Kinect for motion tracking is that no physical markers or sensors are needed on the user, so the users' motion is in no way restricted. It also makes it quick to set up as the camera just needs turning on; in other motion tracking systems a user may need to strap an array of sensors to their body to allow tracking.

As well as direct skeletal tracking, the Kinect can be used for gesture recognition, with certain gestures corresponding to actions from a robot. For example Biao et al. [136] the Kinect monitors hand gestures and the robot moves into pre-set orientations depending on the gesture, and in Qian et al. [137] gestures such as waving move a robot arm in the direction of the wave.

The Kinect can also be used for navigation of mobile robots in indoor environments. For example Benavidez [138] uses a Kinect to allow navigation and object tracking for a mobile robot. Navigation and mapping of indoor environments using the Kinect depth sensor is shown by Correa et al. [139]; here, the depth sensor detects walls and obstacles and the position of these is graphed as the mobile robot moves around the indoor environment, building up a map of the area. This can work well indoors, and as the depth sensor uses infrared light, the Kinect can be used to navigate even in the dark. However, it would be unusable outside as the depth sensor is sensitive to external light.

Another way the Kinect can be used for robotics applications is to identify and track objects, generating coordinates to move a manipulator too. In Siadjuddin et al. [140] a blob detection algorithm is used to separate an object from the background based upon its colour; it then gets the 3D position of the centre of that object and uses it to position the end effector of a robotic manipulator. Similarly Husain et al. [141], depth data is used to segment an object from the background and track it moving within the workspace; the manipulator then moves to the target object until it can grasp it. This is similar to the approach developed later in this chapter.

The work presented here also has similarities to recent work by Kent et al. [142], in which a user clicks a desired grasp position on screen and is subsequently presented with possible grasp positions to select from. It was shown that the point and click interface performed significantly better than the currently available user interface. A major difference with the work in this thesis, however, is that Kent et al. uses a single fixed manipulator, with two 3D cameras, one mounted in front of the workspace and one above, giving a lot of information about the workspace. By contrast, the present research uses dual manipulators on a mobile platform, with a single 3D camera mounted on the platform not fixed in the workspace.

5.1.4 Levels of autonomy

Across robotics in many sectors there is a drive towards autonomy. However, in high risk environments, such as nuclear and search and rescue, full autonomy is often not desirable due to potential risks, and lack of trust in the emerging unproven technology. It is recognised though that direct teleoperation is slow and requires the operator to have lots of training and experience to be effective. As a result, techniques that lie in-between the two are increasingly looked at as a solution, where a level of autonomy is introduced to lower the operator work load but the operator remains in control.

In Bruemmer et al. [143] a robot used for characterisation in nuclear environments is the focus, and a 'mixed-initiative' control scheme is developed to improve over direct teleoperation by allowing the level of automation to shift, giving the operator or the robot more control. In this case, for example, if communication is lost the robot can become more autonomous and carry on the operation, returning control to the operator when communication is restored.

Work by Marturi et al. [48] has investigated the benefit of adding a level of autonomy during manipulation tasks. Direct teleoperation is compared with a visual guided semi-autonomous system for a block stacking task. Results show the semi-autonmous system performed better than direct teleoperation, improving task completion time, precision and repeatability.

5.2 The proposed assisted teleoperation system

In-line with the current industry trend to improve on direct teleoperation methods used in decommissioning tasks, a method that incorporates a level of autonomy has been developed in this work. The new method replaces complex movements using for example joysticks to control each joint independently, with 4 mouse clicks to complete a pipe grasp and cut operation. Full autonomy is undesirable in the nuclear industry, so levels between the two extremes were looked at. It was decided to develop an assisted teleoperation system, where the user maintains control, but low level tasks are automated.

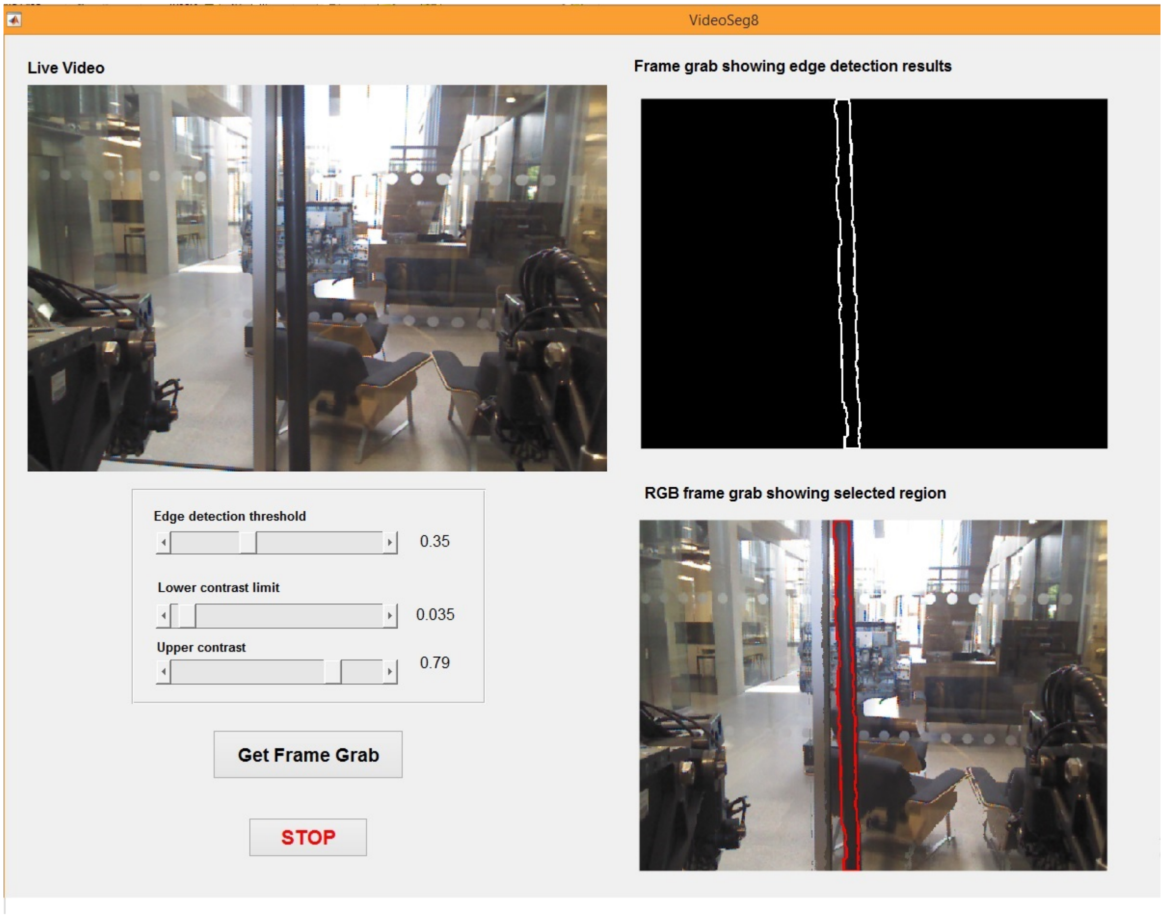
Initially the developed system focuses on pipe cutting, as this was identified as being a commonly carried out and repetitive task in decommissioning. Using the system the user selects the target pipe as well as the grasp and cut positions; the arms then move to these target locations and grasp and then cut the pipe at the selected points. This replaces traditional systems where a user would use joysticks to move each joint independently to carry out the grasping and cutting; this is a highly skilled job requiring experienced operators, and is very time consuming.

The idea behind the system is to have separate modules for different tasks, for example one for pipe cutting and one for pick and place tasks, where the level of autonomy is dependent on the selected task. This would simplify the system, and allow the interface and level of autonomy to be tailored for each task. The present thesis focuses on pipe cutting as an example of the approach.

The GUI developed for this system, shown in Figure 5.1, was developed in Matlab using GUIDE (GUI Development Environment) provided with Matlab. This allowed the appearance of the interface to be set graphically, for example positioning the buttons and graphical displays for the camera video feed. Once the interface layout is set Matlab creates the code which is used as the framework for developing the program, this results in two files that are necessary to run the GUI a '.fig' file for the graphical appearance and a '.m' file for the underlying code. The created code positions and initialises everything and creates handles to the buttons and graphical displays, so that these can be interacted with using normal Matlab programming techniques. For example a normal IF statement can be written that uses the state of the button on the GUI as it's input condition. The object handles created by Matlab can also be used by the MS Kinect interface, to publish the camera image to the display objects.

For the MS Kinect to work with Matlab several things are needed, first is the 'Kinect for Windows SDK', this allows the Kinect to interface with the PC, second is the image acquisition (imaq) toolbox for Matlab, and finally the 'Kinect for Windows Sensor support package' from Matlab. The sensor support package means you can just tell Matlab that the video input device is a MS Kinect, as shown in the first line of code shown in Figure 5.2. The imaq toolbox provides the functions for initialising the Kinect, shown in Figure 5.2, and using the data produced by the RGB and depth sensor allowing it to be used with standard Matlab functions.

An overview of the steps involved in using the interface to achieve a pipe cut are listed here, with a process flow chart shown in Figure 5.3 giving an overview of the process. More detail is discussed in each of the following subsections.

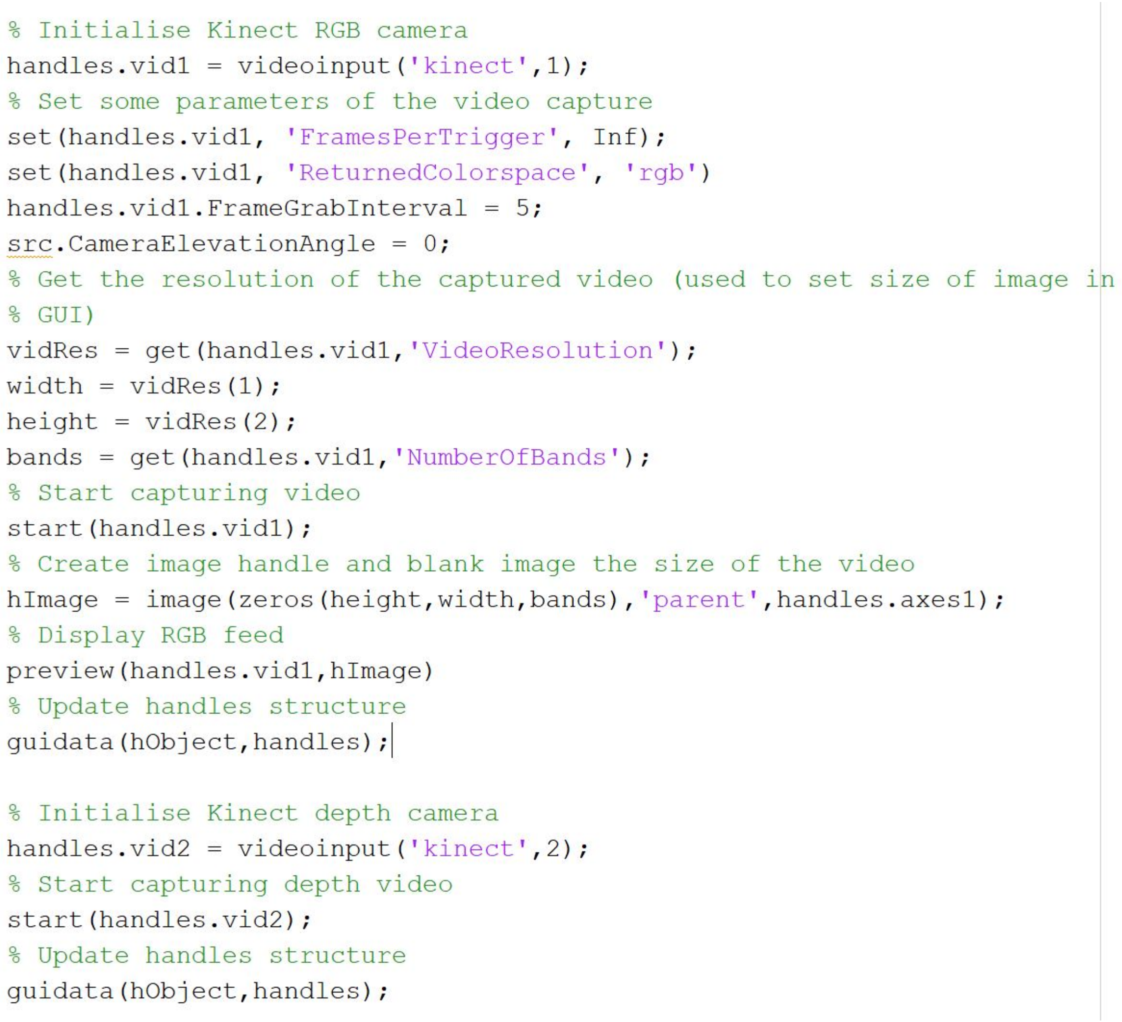

VideoSeg8
Live Video
Frame grab showing edge detection results
RGB frame grab showing selected region
Get Frame Grab
STOP

Edge detection threshold 0.35

Lower contrast limit 0.035

Upper contrast 0.79

Figure 5.1: *The graphical user interface, GUI, presented to the user.*



```
% Initialise Kinect RGB camera
handles.vid1 = videoinput('kinect',1);
% Set some parameters of the video capture
set(handles.vid1, 'FramesPerTrigger', Inf);
set(handles.vid1, 'ReturnedColorspace', 'rgb')
handles.vid1.FrameGrabInterval = 5;
src.CameraElevationAngle = 0;
% Get the resolution of the captured video (used to set size of image in
% GUI)
vidRes = get(handles.vid1,'VideoResolution');
width = vidRes(1);
height = vidRes(2);
bands = get(handles.vid1,'NumberOfBands');
% Start capturing video
start(handles.vid1);
% Create image handle and blank image the size of the video
hImage = image(zeros(height,width,bands),'parent',handles.axes1);
% Display RGB feed
preview(handles.vid1,hImage)
% Update handles structure
guidata(hObject,handles);

% Initialise Kinect depth camera
handles.vid2 = videoinput('kinect',2);
% Start capturing depth video
start(handles.vid2);
% Update handles structure
guidata(hObject,handles);
```

Figure 5.2: *Matlab code used to initialise the MS Kinect camera using standard functions in image acquisition toolbox.*

- **Stage 1:** Position platform in target work area and ensure target object is visible in live video feed.
- **Stage 2:** Click 'take snapshot' button, to capture a still image from live video and carry out edge detection.
- **Stage 3:** Check target object is visible in edge image, and use sliders on interface to adjust edge image if necessary.
- **Stage 4:** Select target object in edge image. This is then superimposed on colour image to confirm correct selection.
- **Stage 5:** Click a grasp and cut location on selected object.
- **Stage 6:** Press return, and manipulators move to given positions carrying out the grasp and cut operation.

5.2.1 Stage 1: positioning

Initially, while the operator moves the mobile platform to the target work area, the MS Kinect is providing live colour video at all times, displayed on the developed Graphical user interface (GUI), to allow the operator to see where they are going. To position the platform the operator has control of the tracks on the base, as well as the Brokk boom arm to position the shoulder plate of the dual manipulators at different heights, or for example to reach forward over an obstacle. The MS Kinect is fixed in place on the plate between the two hydraulic manipulators, so positioning in this way

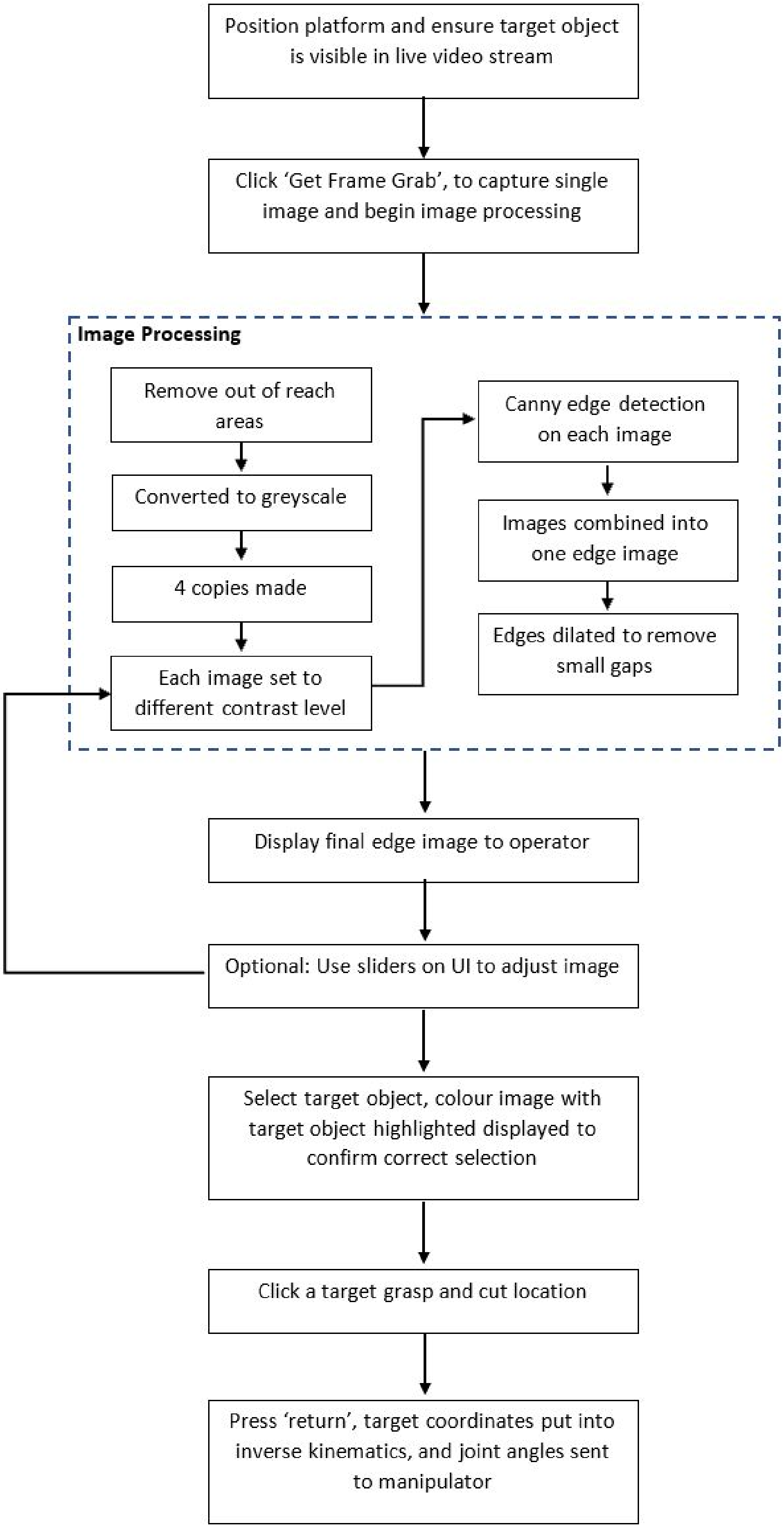

Position platform and ensure target object is visible in live video stream
Click 'Get Frame Grab', to capture single image and begin image processing
Image Processing
Remove out of reach areas
Converted to greyscale
4 copies made
Each image set to different contrast level
Canny edge detection on each image
Images combined into one edge image
Edges dilated to remove small gaps
Display final edge image to operator
Optional: Use sliders on UI to adjust image
Select target object, colour image with target object highlighted displayed to confirm correct selection
Click a target grasp and cut location
Press 'return', target coordinates put into inverse kinematics, and joint angles sent to manipulator

Figure 5.3: *Flowchart showing overview of using the system.*

does not impact on later coordinate system transforms between the camera and the manipulators.

5.2.2 Stage 2: image capture and edge detection

Once the platform is in the desired position, the operator presses a button on the interface, as shown in Figure 5.1. This takes a screen-shot of the live video and performs some processing and presents an edge image to the operator. The screenshot is captured by taking a single frame of the live video when the button is pressed, this is done using the 'getsnapshot' function in Matlab, provided by the image acquisition toolbox. The live colour video continues being shown to the operator, to provide constant visual feedback to the operator, and the captured screenshot goes through several stages of editing and manipulation before being displayed to the operator alongside the live video.

The image is processed to show just the edges of the objects in view, and has objects out of reach removed. This gives a simplified view of the workspace, showing separate objects each enclosed by a continuous solid line. This can be seen in the simple example in Figure 5.1 where a single pipe is enclosed by a continuous solid line. In this example, the pipe extends out of view, and the edges of the camera view are treated as object edges. This is important to be aware of when positioning the platform, as long objects will be segmented by the system based on what the camera can see.

Once the user presses the button to take a screen-shot, multiple steps are carried out in the background that lead to the final edge image. Initially the RGB and depth images are aligned using the ‘alignColorToDepth’ function from the imaq toolbox, this needs doing due to the offset between the sensors on the Kinect, this creates a mirror image of the scene so needs to be flipped using the ‘flip’ function. The image is next converted to greyscale, as the edge detection algorithms used later needs a greyscale image as the input, again the imaq toolbox has a standard tool for this called ‘rgb2gray’. To reduce the size of the image and computational complexity, as well as to only present useful information to the user, all areas that are either out of reach of the system or that have no depth data available are removed.This is achieved by creating a binary mask of the image with each pixel set to a 1 if there is depth data and it is within reach, and a 0 if there is either no depth data or the depth value is out of reach. This binary mask is then multiplied element wise with the now greyscale image to keep the parts within reach and remove the parts out of reach. This leaves the image only containing reachable areas, as illustrated in Figure 5.4, with out of reach areas just being black. Depth data may not exist for a number of reasons, such as surfaces not reflecting the structured infrared light back to the sensor, scattering of the light so that it cannot be returned to the sensor, or objects simply being shadowed from the sensor light. If the object the operator intends to grasp is not shown, they can reposition the manipulator system and try again.

The next stage involves the creation of three copies of the image, each set to a different contrast level using the ‘imadjust’ function, as demonstrated in Figure 5.5. Sliders on the GUI allow the user to adjust the contrast levels, for two of the images the inputs to the ‘imadjust’ function are the values from the sliders, and for the third image preset values are used. These three images, together with the original, are forwarded to the edge detection algorithm. Noting that the working environment of the manipulator system is likely to be poorly lit, this approach lessens the impact of shadows and highlights and is found to capture more detail. Figure 5.6 illustrates such an improvement in captured detail, allowing edges that would otherwise be missed to be captured. This approach was inspired by bracketing techniques used in photography, where multiple shots of a subject are taken with different camera settings and then combined.

The four images are passed through a Canny edge detection algorithm [144], implemented in Matlab, using the ‘edge’ function with ‘canny’ set as the second input (the first input is the image name). The Canny algorithm is a widely used algorithm in image processing in many fields, for example [145,146], and it has been shown that the Canny algorithm performs better than other edge detection algorithms in most situations but is more computationally expensive [147]. Reliable detection of solid edges is a key requirement of the present system, hence the Canny algorithm was chosen. The Canny algorithm allows a sensitivity threshold to be set to alter how strong the edge needs to be to be detected. A third slider on the developed GUI allows the user to adjust this sensitivity threshold, allowing for adjustments to which edges are detected. This can be useful if for example there are markings on the target object which are being detected, by increasing the sensitivity threshold these markings can be ignored and just the edges of the object seen.

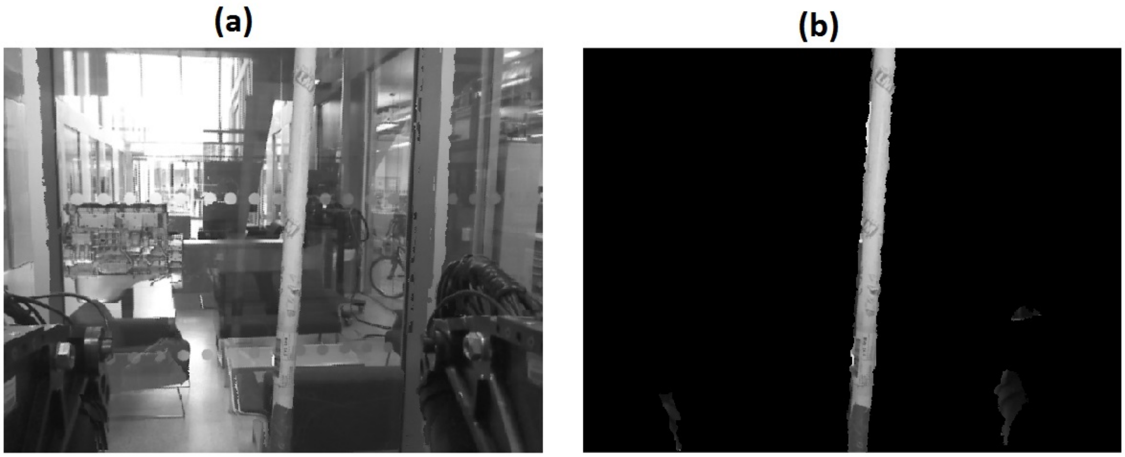


Figure 5.4: *Preliminary image processing to remove background: (a) Original image and (b) Simplified image with out of reach areas removed.*

5.2.3 Stage 3: Operator image adjustment

The four edge-detected images are combined to yield one edge-detected image that contains more detail than if just the original image had been used, as illustrated in Figure 5.6. Due to the edge images being binary images, combining them is a simple case of adding the images together as you would a numerical array. Finally, the combined edge-detected image has the lines dilated in order make the displayed image clearer to the operator. Dilating the lines consists of a couple of stages, first the ‘strel’ function is used to set a morphological structuring element for dilation. The structure used in this case is a line 2 pixels long, so each pixel is checked and if it’s neighbour is set to a 1, representing a line, the pixel being checked is also set to a 1. This is performed in both the horizontal and vertical directions. The purpose of doing this is to close small gaps in the detected edge, and by keeping the structuring element as a line only 2 pixels long false edges aren’t created but small holes in the edge image

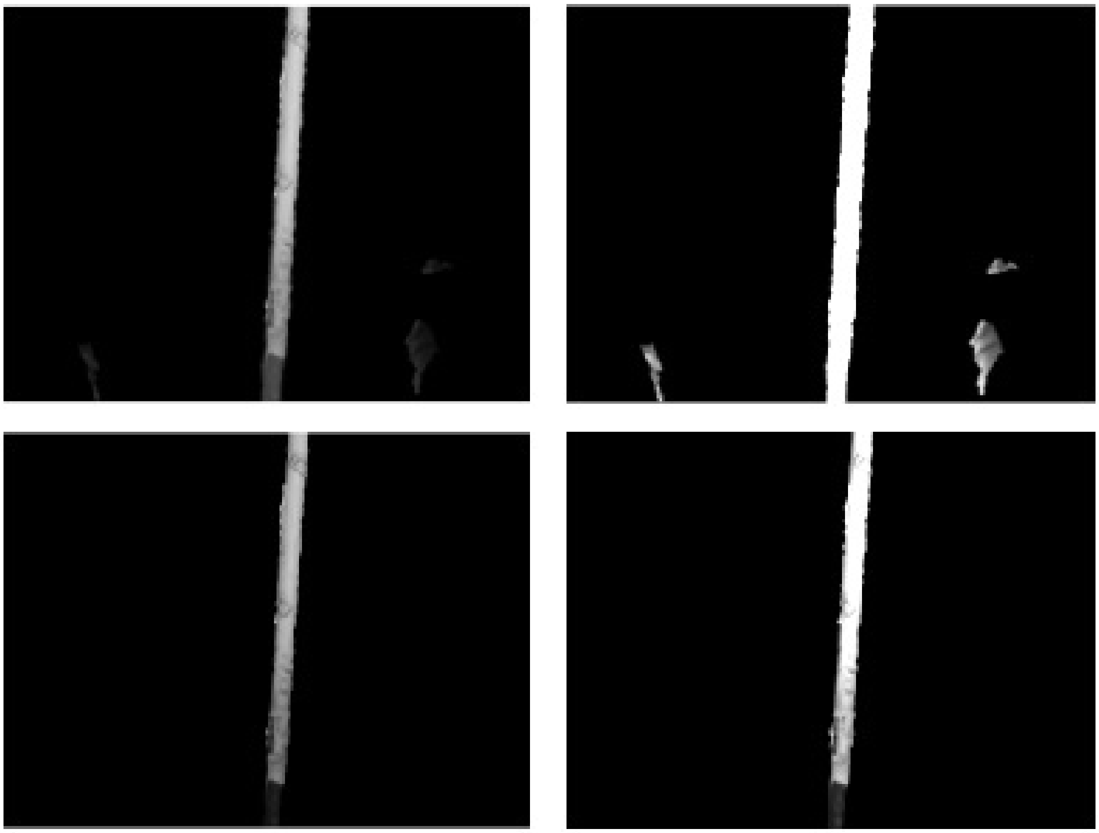


Figure 5.5: *Image contrast examples: Original (top left) and the three different levels of contrast used for edge detection.*

are filled in. The choice of structuring element shape and size was chosen through trial and error, looking at the output of the edge image. The ‘strel’ function can work with a variety of shapes including discs and squares, and the size of that element is set by the user. Pipes tend to have straight edges so it was found the line option worked well, and performing it in both horizontal and vertical orientations allows pipes of any orientation to be found. The edges are then dilated using the ‘imdilate’ function, with the two (horizontal and vertical) ‘strel’ function outputs used as inputs to the dilation function. This is to make the edges clearer when displayed to the operator rather than having very thin lines, and also because the edges detected on the 4 images with different contrast levels don't perfectly overlap, as shown in Figure 5.6, so dilating the edges results in one solid edge.

Note that the three sliders on the GUI, to control the image high and low contrast and the edge detection sensitivity, update the displayed edge detected image in real-time, with a change in slider value triggering the image manipulation function to restart. This allows the operator to adjust the image, potentially removing superfluous edges or filling in missing edges, without having to reposition the system.

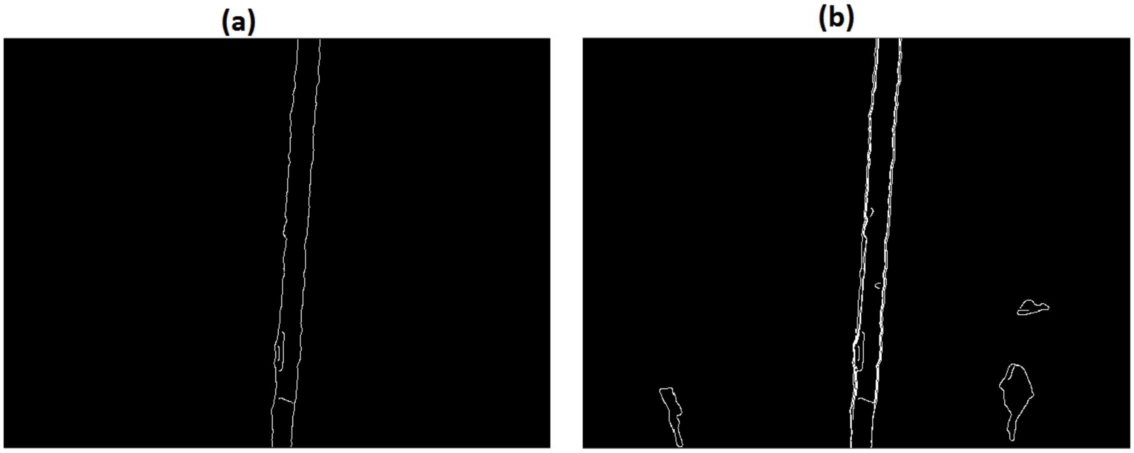


Figure 5.6: *Edge detection results: (a) Original image (b) combining the four different contrast levels.*

5.2.4 Stage 4: Operator target selection

Once the edge-detected image is showing the object the user wants to grasp as an enclosed area, the operator selects it by clicking within this area with the mouse. The image is then updated to show only the selected area. An image showing the colour screen-shot with the selected object highlighted is simultaneously shown, providing visual confirmation to the operator that the correct object has been selected.

To select just the object that the user selected, first the position that the operator clicked is found as an x and y pixel coordinate using the ‘get’ function to get the ‘CurrentPoint’ (input to get function, other options are available) of the cursor. This function is triggered by the mouse click so returns the cursor position where the operator clicked. These coordinates are then rounded to the nearest integer as the ‘get’ function returns a level of accuracy not needed here. With these coordinates available the next step is to check that the operator clicked within the edge image on the GUI, and not anywhere else on the GUI. Assuming they have clicked on the edge image, the area enclosing the clicked point is filled in to create a solid object. This is done using the ‘imfill’ function, which takes the edge image (with dilated edges) and the the mouse click coordinates as inputs and acts as a flood fill function would in image processing software. The image with the filled in area is stored as a new image, and has the edge image subtracted from it (again treating the binary images as numerical arrays) to leave just the selected object as a solid block. For this work at this stage there will usually be a solid rectangle left as that is what sections of pipe look like after passing through an edge detection algorithm, as can bee seen in Figure 5.6.

With the object selected and separated from the rest of the image it’s perimeter is found, this is overlay over the original colour screenshot and shown to the operator as a validation step that they have selected the correct object. To find the perimeter the ‘bwperim’ function is used, which finds the perimeter of objects in a binary image.

This perimeter is then further dilated, in the same way as before, to ensure it is clear when shown on the colour image. To combine this perimeter shape with the colour image, the colour image is first split into it's separate RGB channels giving 3 versions of the image. Then the pixels of the colour image that correspond to the perimeter can be changed, in this case they are set to be red so only one of the images just created needs modifying, this is done as follows 'image(perimeter)=255' where image is the name of the red channel separated from the original image, and perimeter is the binary file of the object perimeter. Combining the three separated channels back into one image gives the original colour image, but with the pixels showing the selected object perimeter changed to red. This image is then displayed on the GUI so the operator can confirm they have selected the correct object before continuing.

5.2.5 Stage 5: Grasp and cut location

If the correct object is selected, the user proceeds to click on the point they determine the manipulator should grasp the object, a pipe in the present context, with a second mouse click at the location they wish the cutting operation to occur. The coordinates of the mouse clicks are found in the same way as before. At this stage, the selected positions for grasp and cut are snapped to the major axis of the object i.e. the centre line of the pipe. The centreline and orientation are determined using the 'regionprops' function in the Matlab image processing toolbox, which can calculate a number of properties about an enclosed region, in this case it us used to find the centre coordinates, the orientation and the major axis length. Using these three things the centreline of the pipe can be found by using Equation (5.1) to find the coordinates of the two ends of the centre line, where (X_1, Y_1) and (X_2, Y_2) are the coordinates of the two end points, length is the major axis length and orientation is the major axis orientation. By creating a linear vector of points between these coordinates the centreline is identified.

$$\begin{aligned} X_1 &= X_{co} - \frac{length}{2}\cos(\theta) \quad Y_1 = Y_{co} + \frac{length}{2}\sin(\theta) \\ X_2 &= X_{co} + \frac{length}{2}\cos(\theta) \quad Y_2 = Y_{co} - \frac{length}{2}\sin(\theta) \end{aligned} \tag{5.1}$$

The operator selected points are snapped to the corresponding points on the centreline with the smallest absolute difference between the selected point and centreline point. To do this first the orientation has to be considered as this will determine if the closest point is along the horizontal or vertical axis. If the orientation is under 45 degrees from the horizontal, the translation will be along the X-axis, so the absolute difference between the selected X coordinate and each X coordinate element in the vector describing the centre line is found. The element in this array with the smallest absolute difference is then identified and the user selected coordinates are changed to be the identified coordinates from the centreline vector. If the orientation is over 45 degrees the same process happens but with the Y coordinates instead of the X coordinates. Hence, the user does not need to accurately click the centre line of the pipe, making the operator task easier. Furthermore, the grasp target is centred on the major axis of the pipe so that the gripper fingers will not hit the pipe, as it might for example if the user clicked off to one edge of the pipe.

The start and end positions for the cutting operation are determined such that the path of the cutter is perpendicular to the major axis of the pipe, at the position the operator selected. The start and end-points also take a pre-set distance from the edge of the pipe, by finding the width using the minor axis length property of the 'regionprops' function and adding a set value either side. This is to allow any width of pipe to be cut, and ensures clearance for the cutting tool before and after the cut. The coordinates for the cut start and end positions are found in a similar way to that shown in Equation (5.1), but using the orientation plus 90 degrees to make it perpendicular and the length value is the minor axis length plus the pre set offset.

By calculating the grasp location and cutting path based on the major axis of the selected pipe, any orientation of pipe can be addressed, such that the system is able to cut pipes that are horizontal, vertical or at any angle in-between. Since depth data are available at both the selected points, the pipe does not even have to be in one plane, the grasp and cut points could be at different distances. To reduce the possibility of collisions between the end-effector and the pipe being grasped, and to help position the gripper in the target orientation, the end-effector first moves to a position directly in front of the target grasp position; subsequently, when it has reached a set error tolerance for that position, it is moved into the final grasp position.

5.2.6 Stage 6: Operation

At this stage there are four goal positions i.e. the cutting operation start and end positions, the initial position directly in front of the grasp location and the final grasp location. Until this stage, these points have all been addressed as pixel coordinates i.e. the pixels that the user clicked and the corresponding depth values. To be used as input arguments to the Inverse Kinematics (IK) algorithm these are converted into the manipulator coordinate system. Keeping everything in pixel coordinates until this stage allows the system to be modular. The vision processing and user interface is one module, and the kinematics algorithm is a separate module. This allows the user interface to be applied to different manipulator systems, simply by changing the kinematics algorithm.

To convert from pixel coordinates to the manipulator coordinate system, a trigonometric approach is taken. The number of pixels in the image from the Kinect is known, i.e. 640 × 480, and the field of view of the camera is 57° horizontal and 43° vertical hence it is a straightforward matter to determine the degrees per pixel: 0.089° for both horizontal and vertical. By shifting the origin of the camera image from the upper left corner (default position) to the centre of the image, the angle between the camera centre and the user-selected point can be determined. The transformation from the original pixel coordinate system, to shifting the origin to the centre of the image is simply:

$$x_1 = x - 320 \qquad y_1 = y - 240$$

where x and y are the pixel values where the user originally clicked. This formulation ensures that points to the right and above the centre are positive and points to the left and below the centre are negative, which yields the same positive and negative directions as the manipulator coordinate system. Using these coordinates and the depth data, the real world coordinates relative to the camera origin are determined as followed

$$X = Z \times tan\,(0.089 \times x_1)$$

$$Y = Z \times tan\left(0.089 \times y_1\right)$$

where Z is the depth value and 0.089 is the degrees per pixel value determined previously. For the Kinect, points on a plane parallel to the sensor all have the same depth value (rather than the straight line distance from the point to the sensor) and this defines the trigonometric identity to use. At this stage of processing, the coordinates are relative to the camera centre. Hence, a straightforward translation is used to convert these into a usable form for manipulator movement, by translating the camera origin to the manipulator system origin, which is the shoulder of the right side manipulator. The transformation from the user-selected pixel, to the manipulator coordinate input is, Equation (5.2):

$$\begin{bmatrix} X_t = Z\tan\left(0.089\left(x-320\right)\right)+275 \\ Y_t = Z\tan\left(0.089\left(y-240\right)\right) \\ Z_t = Z \end{bmatrix} \tag{5.2}$$

where x and y are the pixels the user clicked on, and the arrays X_t, Y_t, Z_t are the manipulator target coordinates, determined for each of the four goal positions. A Jacobian transpose based IK solver is utilised to find the associated target joint angles. These joint angles represent set points for the feedback control algorithms that control the hydraulic actuators.

Everything on the interface and the calculations up to this point have been implemented in Matlab, however the hardware requires control signals from NI Labview. As such the joint angles are sent via a TCP/IP link to Labview and used as the set points to control systems implemented in Labview to generate a voltage signal to send to the CFP that controls the hydraulic valves. This architecture allows the GUI to be on one computer, and the controllers to be on a separate machine, further facilitating remote operation.

5.3 Manipulator Kinematics

The present section considers the kinematic equations necessary to resolve the motion planned above.

5.3.1 Forward Kinematics

Forward kinematics is used to find the end effector position, in cartesian coordinates, relative to the manipulator origin, given the angles of each joint in the manipulator. To solve the forward kinematics problem two things are needed, the vectors of the dimensions of each link, see Table 5.1, and the rotation matrix for each joint, Table 5.2. The rotation matrices define the direction and axis of rotation, for a given angle θ. Joint 6 is the gripper rotation, which doesn't impact on the kinematics and so the rotation matrix is simply an identity matrix. In this case, due to the geometry of the manipulator, all the rotation matrices are elemental rotations, as they each only rotate about a single axis of the coordinate system.

By starting at the shoulder end and working towards the end effector, following the chain of translations multiplied by rotations, equation (5.3) below can be used to find the end effector position if joint angles are known. Appendix C shows the

Table 5.1: *Vectors of link lengths for Hydrolek Manipulator.*

Link	X **(mm)**	Y **(mm)**	Z **(mm)**	**Name**
J_{01}	0	0	0	L_0
J_{12}	70.00	0	0	L_1
J_{23}	523.48	0	0	L_2
J_{34}	165.00	-212.60	0	L_3
J_{45}	44.45	0	0	L_4
J_{56}	284.80	-18.47	0	L_5
J_{67}	0	0	0	L_6

Forward Kinematics equation for the Hydrolek manipulator.

$$P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = L_0 + R_1 \left(L_1 + R_2 \left(L_2 + R_3 \left(L_3 + R_4 \left(L_4 + R_5 \left(L_5 + R_6 L_6\right)\right)\right)\right)\right) \tag{5.3}$$

5.3.2 Inverse Kinematics

Inverse kinematics is the opposite of the forward kinematics i.e. the calculation of the necessary joint angles required to reach a given end effector position (or position and orientation). There exist several common methods for doing this and, for some manipulators, an analytical solution exists. However, commonly a numerical solution needs to be found. This is generally an iterative process, with two approaches

Table 5.2: *Rotation matrices for each joint.*

Joint	**Axis of rotation**	**Rotation matrix**	**Name**
J_1	Y-axis	$\begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) \\ 0 & 1 & 0 \\ -\sin(\theta_1) & 0 & \cos(\theta_1) \end{bmatrix}$	R_1
J_2	Z-axis	$\begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix}$	R_12
J_3	Z-axis	$\begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix}$	R_3
J_4	Y-axis	$\begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) \\ 0 & 1 & 0 \\ -\sin(\theta_4) & 0 & \cos(\theta_4) \end{bmatrix}$	R_4
J_5	Z-axis	$\begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 \\ 0 & 0 & 1 \end{bmatrix}$	R_5
J_6	X-axis	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	R_6

dominating: the Jacobian transpose and Cyclic Coordinate Descent, CCD, method. Typically the CCD method is used in animation and video game applications and the Jacobian method is used on robotic applications.

The Jacobian is a matrix of partial derivatives of the forward kinematics equation. It is typically an $m \times n$ matrix, where m is degrees of freedom used to position the end effector (i.e. 3 DOF for position only and 6 DOF for position and orientation), and n is the number of joints. Equation (5.4) shows an example of a typical 6 by 6 Jacobian matrix. The Jacobian for the Hydrolek manipulators is shown in Appendix D.

$$J(\theta) = \begin{bmatrix} \frac{\delta f_x}{\delta \theta_1} & \cdots & \frac{\delta f_x}{\delta \theta_6} \\ \vdots & \ddots & \vdots \\ \frac{\delta f_w}{\delta \theta_1} & \cdots & \frac{\delta f_w}{\delta \theta_6} \end{bmatrix} \tag{5.4}$$

The Jacobian matrix gives a linear approximation of the system at θ, as shown in Equation (5.5). From Equation (5.5) it is clear that it can be rearranged to express this in terms of $d\theta$ by using the inverse of the Jacobian. However it is known that using the inverse Jacobian is unstable and gives poor performance around singularities [148].

$$dX \approx J.d\theta \tag{5.5}$$

An alternative to using the inverse Jacobian is to use the Transpose, since this removes the need for computationally expensive inverses and performs better near singularities. Equation (5.6) shows $d\theta$ using the transpose method, where α is a small scaling factor and $\alpha > 0$.

$$d\theta = \alpha J^T.dX \tag{5.6}$$

Both the transpose and the inverse technique are iterative processes making small changes to the joint angles i.e. each loop repeating until the end position is within a predefined tolerance. The scaling factor for the transpose method is calculated at the start of each iteration of the loop. Equation (5.7) shows how the scaling factor is calculated, where dP is the difference between the target and current end effector position. A good scaling factor can reduce the number of iterations needed for the end effector to converge to the target.

$$\alpha = \left(\frac{|J^T dP|}{|J.J^T dP|} \right)^2 \tag{5.7}$$

Previous work at Lancaster University on the inverse kinematics problem for the Hydrolek arms is reported in the MSc project of Pierre Besset in 2013 [149]. In his work, he developed several Matlab implementations of the Jacobian Transpose method, one using just position so being 3 DOF, one using position and orientation, hence 6 DOF, and one able to generate trajectory data between two positions. These solutions worked, however they were very slow to compute and so unable to be used for any real time control, with the 6 DOF solver taking on average around 13 seconds to generate the joint angles for each target position.

Using this existing Matlab algorithm as a starting point, a solution was created for the 6 DOF Inverse Kinematics problem, which calculates the joint angles both fast and accurately enough to be used for real time control. This was mainly achieved by re-structuring the code and performing some calculations in different more efficient ways. One modification that caused the biggest improvement was the removal of symbolic variables. Besset [149] had calculated the Jacobian matrix using symbolic variables in place of the joint angle numerical values. By removing the symbolics and moving the Jacobian matrix calculation inside of the main loop of the algorithm, a significant increase in performance was achieved.

The average time for Bessets code to calculate joint angles given a target position and orientation was 13.67 seconds. Using the same timing technique, the average time for the new algorithm to calculate the joint angles was 0.0203 seconds. Figure 5.7 shows an example of the run time for the new IK algorithm on 400 randomly generated target positions, with a euclidean position error tolerance of 2 mm on the IK algorithm, the mean run time is 0.0158 seconds. As can be seen some, points take significantly longer to run, this is due to the positions being randomly generated, so sometimes there are very large changes between the current position and the target position. This is a worst case scenario as under normal operating conditions the arm wouldn't often be moving from one extreme to another, it would likely be making small movements or following a trajectory.

Table 5.3 shows a comparison of mean run time and position error for different error tolerances. The final column shows the average position error if errors greater than 10 mm are removed. As can be seen from Figure 5.8, which corresponds to the second row in the table with a tolerance of 2 mm, this doesn't remove many points, but does have an impact on the mean as the outliers tend to be a long way out. The number of outliers is in part due to the random sample points, and the Jacobian

Table 5.3: *Comparison of mean run time and position error for various tolerances.*

Position error tolerance (mm)	**Mean run time (s)**	**Mean position error (mm)**	**Mean position error ignoring outliers $>$ 10 (mm)**
5	0.0230	6.3496	4.8278
2	0.0429	2.9087	1.9315
1	0.0553	2.7949	1.0274
0.5	0.0873	1.6577	0.5166
0.1	0.1817	1.3907	0.1397

method getting stuck moving between certain points. In a real world situation, such as following a trajectory this would be much less pronounced. By ignoring the outliers it can be seen that the mean error is very close to the set tolerance. It can be seen the Jacobian Transpose method performs reasonably well, and the modifications to the original algorithm have resulted in an algorithm that is both fast and accurate enough to be used in real time applications, as needed here.

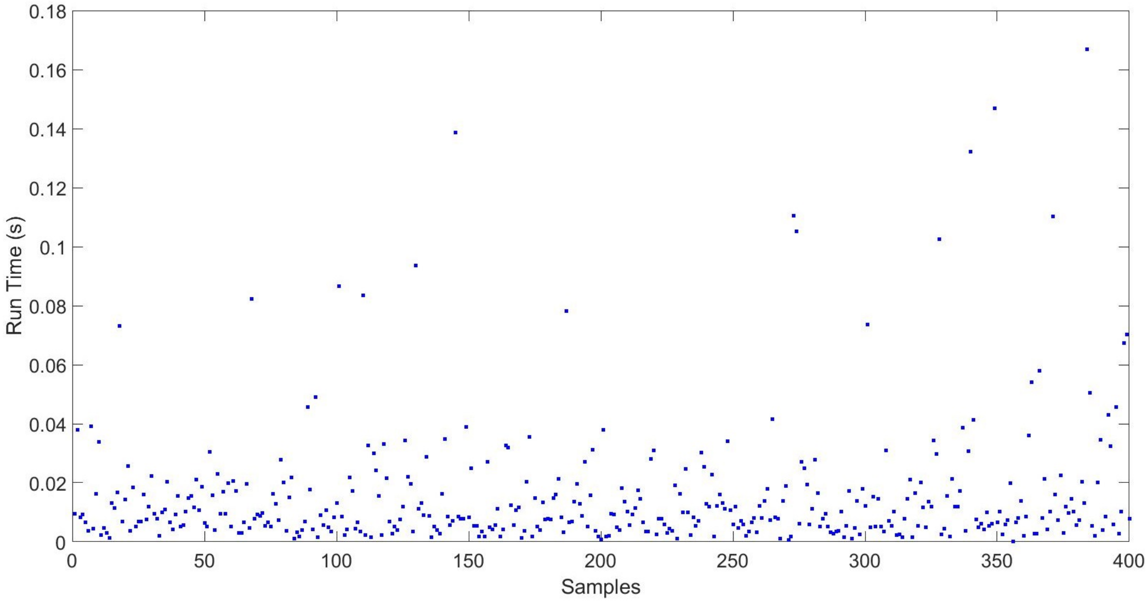


Figure 5.7: *Execution time for* 400 *random samples, error tolerance of* 2*mm.*

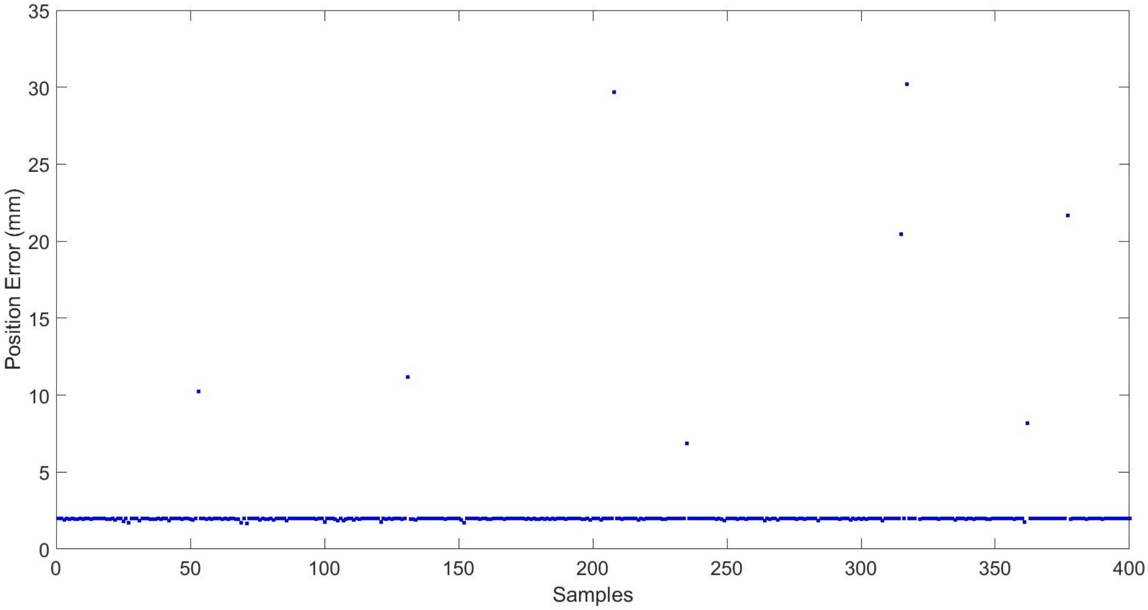


Figure 5.8: *Calculated position error for* 400 *random sample, error tolerance of* 2*mm.*

5.4 Testing of assisted teleoperation system

Initial evaluation tests show that the system works effectively, but that the manipulators struggled to reach the target position accurately, highlighting a need for an improved control system. The tests showed the operator can see and select a pipe on the GUI, and the manipulator successfully grasps and moves as though cutting through it (pre-empting the use of appropriate cutting tools in future research). Tests were repeated from various random starting positions, and successful grasping was achieved each time. Figure 5.9 shows an example of the right hand side manipulator moving to a target grasp location. It shows small movements around the initial target position, before proceeding to the final grasp location, the final end-effector position has a Euclidian distance error of 17.1 mm from the target, and takes 5.4 seconds to complete the movement. This is within acceptable tolerances for the grasping, however for cutting improved accuracy is needed. When a cutting tool is moving between two points, the errors stack up and could become significant. This is particularly an issue if a mechanical cutting tool such as a circular saw is used, where the errors could cause for example a horizontal line to be non horizontal, stressing the cutting tool and potentially causing a breakage.

Since the new system actuates all the joints in parallel, significantly faster moving times are possible, compared to conventional teleoperation via joystick control. The latter approach requires different buttons to be pressed in various configurations to control each joint. To test this some basic tests were carried out using both experienced and inexperienced operators, each performing grasps with both the traditional joystick direct teleoperation and the vision based autonomous system.

The results are given in Table 5.4 for 12 experiments similar to that shown in Figure 5.9. Each experiment starts with the manipulators and pipe in the same position. First the operator uses the joystick method to grasp the pipe. This is

Table 5.4: *Comparison of direct teleoperation and vision based assisted teleoperation, showing time to complete a pipe grasp. Times shown in Min:Sec.*

Teleoperation	**Assisted teleoperation**	**Reduction**	
Inexperienced User			
2:02	0:31	1:31	74.6%
3:09 (Failed, damaged pipe)	0:22	2:47	88.4%
2:28	0:27	2:01	81.8%
Experienced User			
0:57	0:20	0:37	64.9%
1:04	0:21	0:43	67.2%
1:08	0:20	0:48	70.6%

repeated three times. The same user subsequently operates the new vision system, again three times. These tests are repeated for both an experienced and inexperienced operator. The inexperienced user had never used the system before but received 15 minutes training and practice with the joysticks. The experienced operator was already familiar with the joystick control method.

The times taken to complete the task are listed in Table 5.4, which shows that the new system yields considerably faster performance for the grasping task for both operators, with an average reduction in task completion time of 81.6% and 67.6% for inexperienced and experienced operators respectively. It is interesting to note that the execution time is only a few seconds longer for the inexperienced user when using the vision system, suggesting that it is relatively straightforward to learn how to use the new system. These early results show the potential of this system, as even with an experienced operator a large reduction in operating time is achievable, and as discussed earlier for the nuclear industry improving efficiency and reducing costs are important requirements.

It should be stressed that these preliminary tests are quite limited in scope and, to provide a more thorough comparison, more users and more detailed test scenarios need to investigated and carried out in future research. One approach to this would be to use the NASA task load index (NASA TLX) assessment. This assessment gets users to subjectively rate a task on six different areas, and is used widely in human factors research. This approach is used in Marturi et al. [48], where the effect of adding a level of autonomy to direct teleoperation systems is investigated, in particular for point to point movement and a block stacking task. However, these initial results provide an indication of the potential for improved performance using the new assisted teleoperation approach, compared to the directly teleoperated system presently used on nuclear sites.

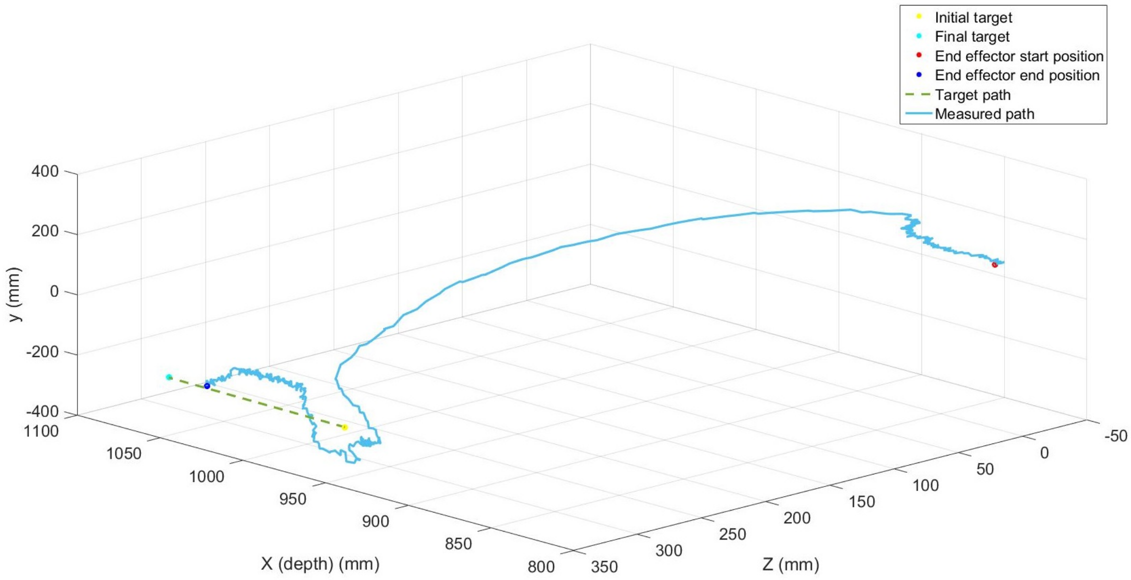


Figure 5.9: *End effector trajectory during grasp motion.*

5.5 Concluding Remarks

This chapter has developed and evaluated a vision based semi-autonomous object grasping system. In a similar manner to the research on parameter estimation for robotic systems in Chapter 4, the Brokk-Hydrolek system is again used as the case study example. However, it should be stressed that the concept is applicable to future dual–manipulator nuclear decommissioning robots in general. The system presents a straightforward GUI to the operator, who with just four mouse clicks can select target positions for each manipulator to perform a pipe grasp and cut action. Throughout the process, the user can view the live colour video and terminate manipulator movements at any time. The system was tested in a laboratory environment. It is shown to work successfully, outperforming the traditional joystick-based teleoperation approach. The system keeps the user in control of the overall system behaviour but significantly reduces user workload and operation time.

Further research is required to support these conclusions, including additional experiments with more users, and particularly in relation to attaching suitable cutters and working with real pipes and other objects. Other possible future experiments include using both manipulators to grasp the same object, in order to move particularly heavy objects. In addition, one limitation of the present prototype is the lack of a sophisticated collision avoidance system i.e. to prevent the manipulators colliding with each other or the work environment.

Another limitation of the results reported in this chapter is the use of standard PID control algorithms for low-level control. For brevity and to avoid repetition with the following two chapters, analysis of the closed-loop control performance has not been discussed above, but it suggests there is considerable scope for improving the control of joint angles. Hence, research into the low-level joint control problem and the development of novel control systems are discussed in the following two chapters.

Chapter 6

Control System Development

Previous research (see references below) and recent testing of the new assisted teleoperation system, as reported in Chapter 5, motivates research into the development of improved control systems. Chapters 6 and 7 investigate various novel design methods for low-level control of each manipulator joint. The present Chapter 6 considers model-based methods evaluated in simulation, whilst Chapter 7 reports on the experimental results and also introduces an alternative, fuzzy logic approach. Since the interactions between each joint are relatively small in practical terms, single- input, single-output controllers are developed, one for each degree of freedom of the manipulator (i.e. the final developed algorithm consists of a multiple-loop control system).

The control objective is to automatically determine the required voltage to each manipulator valve, so that the associated joint angle follows a pre-specified set point or desired angle. The experiments reported below utilise fixed set point sequences for research and development purposes, including step inputs and sine wave signals.

However, it is assumed that for real-time applications, these set points will be provided by the high-level assisted teleoperation system introduced in Chapter 5. Relatively straightforward design approaches are preferred, so that these can be quickly updated when necessary and transferred to other manipulators.

As such the design objectives for the controllers developed in this chapter are:

- To have a rise time of less than 2 seconds. This will ensure rapid movement, which is one of the main advantages of the semi autonomous system over direct teleoperation

- To have an overshoot of under 10 percent of the step movement. Having rapid movement often results in an overshoot, but due to the potential of the manipulator to cause damage to the mobile platform or the work environment, it is necessary to keep this to a minimum.

- To have a steady state error of under 0.5 degrees. Each joint may have a position error and these add up to effect the end effector position error, as such each joints steady state error needs to be kept to a minimum. However the fact that these are real hydraulic actuators and that a number of assumptions are made about them during the SDP model development process also needs taking into account.

Although the main focus is on achieving accurate and fast set point following, a secondary aim is to generate a “clean” (low variance) voltage signal. A noisy or rapidly oscillating voltage causes rapid valve opening and closing, leading to excess wear on these components and increasing the likelihood of failure. However this value is difficult to quantify as a reference ideal signal does not exist to compare against, so this measure is judged relative to the other controllers, with the lowest variance value being the best.

To help gain more information about the system, and to inform decisions about future controller development, the dynamic behaviour of the Hydrolek manipulators are first systematically investigated by the present author through a set of open-loop experiments for each joint. These data are used to develop low order and relatively straightforward, although necessarily nonlinear, state-dependent parameter (SDP) models. In contrast to the mechanistic model described in Chapter 4, such SDP models can be directly utilised for control system design. Although not reported here, it should be noted that the mechanistic model has been used to generate simulation data that can yield similar SDP model forms i.e. by applying system identification methods to the mechanistic model. However, since the hardware Brokk-Hydrolek system was readily available in the laboratory for the purpose of this project, it was directly utilised for open-loop data collection, SDP model identification and subsequent estimation of the appropriate numerical values of the control coefficients. As a result, the identified SDP models are also used for initial control evaluation (Chapter 6) whilst, of course, the Brokk-Hydrolek is directly used for the hardware evaluation of the approach (Chapter 7).

The present chapter utilises these SDP models for model-based control system design within a non- minimal state-space (NMSS) framework [11]. This includes investigations into conventional linear Proportional-Integral (PI), linear Proportional-Integral-Plus (PIP) control, and a nonlinear approach based directly on the SDP model.

SDP models, see for example Young et al. [150], are estimated from data in a stochastic state-space framework. The SDP model parameters are functionally dependent on measured variables such as joint velocities and positions. An SDP model approach was chosen over other methods, such as the commonly used Inverse Dynamic Identification Model and linear Least Squares (IDIM-LS), for a number of reasons. In the first instance, SDP models have already been used for modelling hydraulic systems in related heavy plant robotic applications [95]. Secondly, recent work by Janot et al. [151] has demonstrated the advantages of the SDP approach in comparison to the IDIM-LS approach, applied to both an electro-mechanical positioning system and a TX40 robotic manipulator. Finally, previous work at Lancaster has applied SDP models to the Brokk-Hydrolek platform and successfully used these for control system design (see e.g. [7, 87]).

The present chapter builds on all these prior developments. However, the above cited research into SDP control of the Brokk-Hydrolek has a number of limitations. In particular, a new way of parameterising the SDP model is developed in this chapter, one that better addresses the dead-zone limits of the system. Although a voltage between $\pm 10V$ can be sent to the CFP to control actuator movement, it is known from previous work that there is a dead-zone where low voltages do not cause actuator movement. Taylor and Robertson [7] and Robertson et al. [89] (and other articles from this time cited earlier) all utilised a scaled input signal that approximately eliminated this dead-zone in a rather *ad hoc* manner. Unfortunately, the calibration for such scaling is time consuming and needs repeating as the hardware is serviced and maintained, and because properties such as friction characteristics can change over time. By contrast, the new approach developed here focuses on the joint angular velocity. This yields a different SDP model structure that more clearly highlights the asymmetric behaviour of the hydraulic actuators, and implicitly estimates the dead-zone and velocity saturation points. Although developed independently, the approach has some similarities to the recent friction analysis work by Janot et al. [151] (albeit for the case of robots that are not hydraulically actuated). One advantage of the new SDP model form is that it immediately facilitates the use of an Inverse Dead- Zone (IDZ) approach to the control problem e.g. Fortgang et al. [152]. The new control approach developed in this chapter exploits an IDZ control element to address the nonlinearity, while straightforward linear methods handle the dynamic response and set point tracking.

Also, the earlier PhD thesis on control of the Brokk-Hydrolek [7] focused on the use of just two joints, whilst the present thesis necessarily extends these results to additional joints as required for the pipe cutting task. Nonetheless, it should be noted that due to hardware failure, joint 5 of the right hand manipulator is not included in the present analysis, and is left fixed in position for the resolved motion and grasping tests. Also, as joints 6 and 7 for the gripper rotation and closing do not presently have sensors fitted to give position feedback, these are also omitted. As a result, the present work focuses on controller design for 4 joints, which is sufficient for positioning the end-effector. Finally, although SDP models have been estimated for both manipulators, it was decided to focus on the right hand manipulator for the purpose of the present chapter i.e. the manipulator performing the grasping task.

This decision is in part due to the fact that the choice of cutting tool, to be fitted to the left manipulator, is a subject for future work, and this may require hardware modifications that have a knock on effect on the controller design. However, it should be stressed that the present chapter aims to develop a generic approach to control that is readily applicable to other manipulators.

Section 6.1 discusses the development of the new SDP model form and its estimation from experimental data. This is followed in section 6.2 by the design and preliminary evaluation of each control algorithm. Various implementation structures and design approaches (e.g. closed-loop poles) are compared in simulation. The investigation starts with a simple PI controller and incrementally adds a level of complexity. This yields five different but similarly structured controllers, namely PI control, PIP with no dead-band scaling, PIP using the original approach to the dead-band based on a scaled input signal, PIP control with the new IDZ control element, and finally a forward path PIP implementation of this. Section 6.3 systematically compares these designs using SDP based simulations, with the conclusions drawn in Section 6.4.

6.1 Model development

6.1.1 Data collection

Open loop step tests were used to investigate the behaviour of each joint, Figure 6.1 shows an example of the output of these tests for joint 2 of the right hand manipulator, normalised for initial conditions of zero. Each joint was investigated separately, with a series of step inputs covering the $\pm 10V$ input range, in increments of 2 volts, with increments of 0.5 volts used between $\pm 2V$ to better capture details about the deadband. These tests were carried out with an initial value of 0 before the step input, i.e. the joint being static before the input, and with an initial steady voltage before the step, i.e. the joint moving before the step. These pre step voltages covered the range of possible voltages again in increments of 2 volts, and with both positive and negative initial values for each step. This gave a large data set of open loop step tests covering all possibilities, the joint static then receiving a step input, the joint moving and receiving a step input to continue in the direction of movement, and the joint moving and receiving a step input to move in the opposite direction. This resulted in 150 data files for each of the five joints fitted with potentiometers, of both manipulators. Thus creating a large experimental data set that will aid future research. For the work discussed here only the data files with the zero initial input, i.e. the joint starting from static, are used.

6.1.2 Transfer function model

Using this gathered open loop data and the RIVID function in the Captain toolbox (see Young et al. [153]) in Matlab a model structure that fit the data was identified. The results show a first order linear difference equation, in the form of Equation (6.1) was suitable for modelling the joints.

$$y_k = -a_1 y_{k-1} + b_\tau v_{k-\tau} \tag{6.1}$$

with $\{a_1, b_\tau\}$ being time-invariant and the time delay, τ, being dependent on the sampling interval Δt.

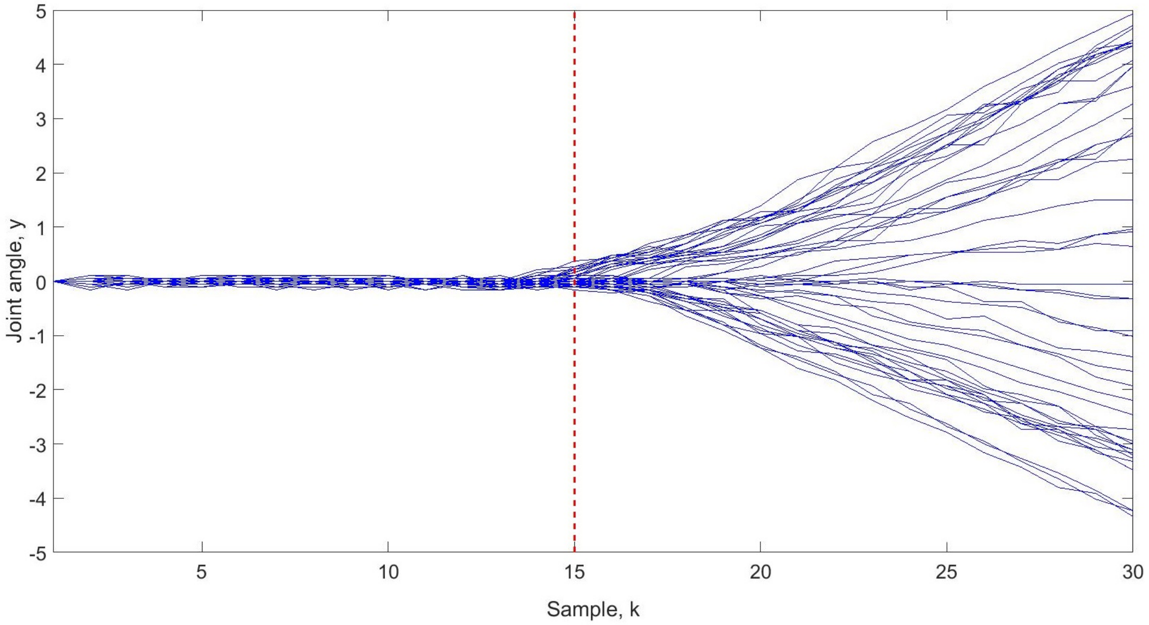


Figure 6.1: *Joint angle* y_k *plotted against sample* k*, for* 40 *experiments each with a step input of different magnitude at* $k = 0$*, with* $\Delta t = 0.01s$*. The estimated time-delay* $\tau = 15$ *samples is shown by the dashed line. This is illustrative example is for joint 2 of the right hand manipulator.*

Analysing the open loop data and interpreting the model in Equation (6.1) it was determined that a value of $a_1 = -1$ was appropriate, but it was clear that there were different values of b_τ for each step experiment depending on the amplitude of the input.

6.1.3 SDP model development

For the next stage RIV estimates of b_τ were found for each step experiment, again using tools in the Captain toolbox. For this the sampling rate $\Delta t = 0.01s$ was used to match the raw data sampling rate, with $a_1 = -1$ and $\tau = 15$ determined from visual inspection of the data (estimated time delay is shown in Figure 6.1). The value of $a_1 = -1$ being determined by the ramp response to a step input, visible in Figure 6.1, indicating that the system behaves as an integrator. Figure 6.2 shows these RIV estimates of b_τ showing the clear state-dependency on the input, V.

Although this illustrative example is for joint 2 the same process was repeated for each joint, with the time delay being approximated by eye from the step response data plotted similarly to in Figure 6.1. Visual inspection showed that there is a range of time delays for different inputs, with the time delay noticeably increasing as input voltage approaches the dead-zone. Developing methods that deal with varying time delays was outside the scope of this project, and so a compromised approach was taken where a single time delay value for each joint was chosen by eye from the data. For joints 1 to 3 of the right manipulator a time delay value of 15 samples, with a sampling rate of 0.01s, was chosen and for joint 4 25 samples was used. As well as having the longest time delay joint 4 also exhibited the widest range of different time delays, whereas the first 3 joints were all similar. This may be connected to the fact that the first three joints are linear actuators and joint 4 is a rotary actuator. This time delay issue as an area that could be further investigated in future work.

Using Equation (6.1) with $a_1 = -1$ gives $b_\tau v_{k-\tau} = y_k - y_{k-1}$, i.e. $b_\tau v_{k-\tau}$ gives an

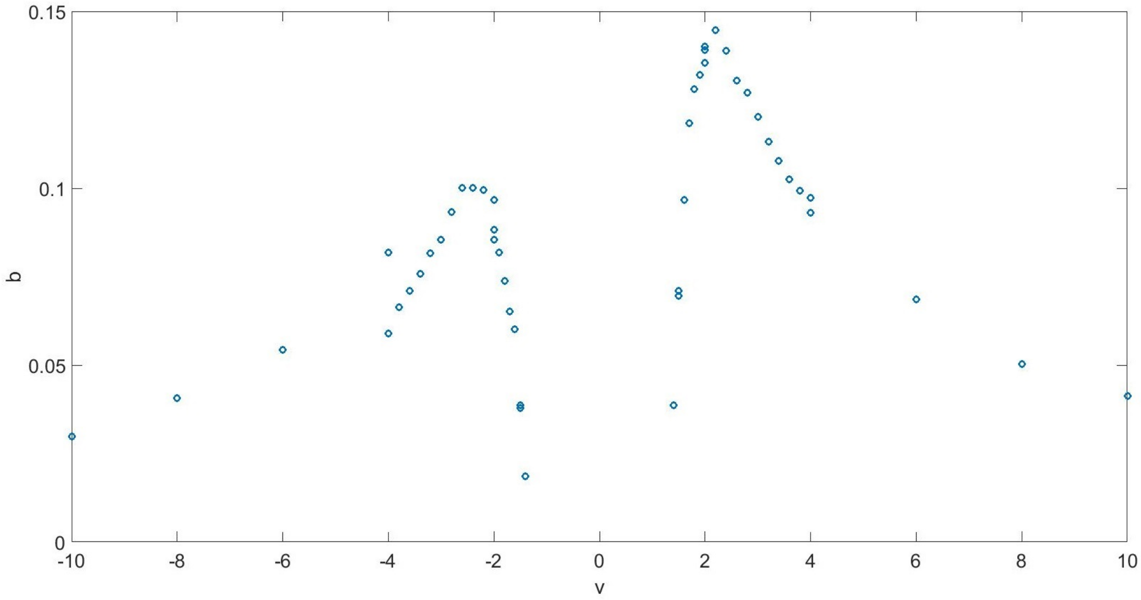


Figure 6.2: *State-dependent parameter plotted against input magnitude (i.e. voltage to NI-CFP) showing RIV estimates for b_τ for each step experiment.*

estimate of the differenced sampled output signal. Dividing this by the sampling rate, $b_\tau v_{k-\tau}/\Delta t$, gives an estimate of the joint angular velocity in degrees/s, as plotted in Figure 6.3. The circles in Figure 6.3 are the RIV estimates and the solid line is from an fminsearch estimation of the SDP model using the raw open loop data (v_k, y_k), i.e. the joint angles from Figure 6.1. Processing the data in this manner, and plotting it as in Figure 6.3, clearly highlights the dead-zone and saturation points, as well as highlighting the asymmetrical behaviour.

Exponential functions could be directly fitted to the RIV estimates shown in Figure 6.3. Due to the asymmetrical actuator behaviour, e.g. the two movement directions have different saturation velocities, clearly visible in Figure 6.3, two functions are required for each joint. To fit the exponential functions to the data the 'lsqcurvefit' function in Matlab is used. This uses a nonlinear least-squares solver to find the

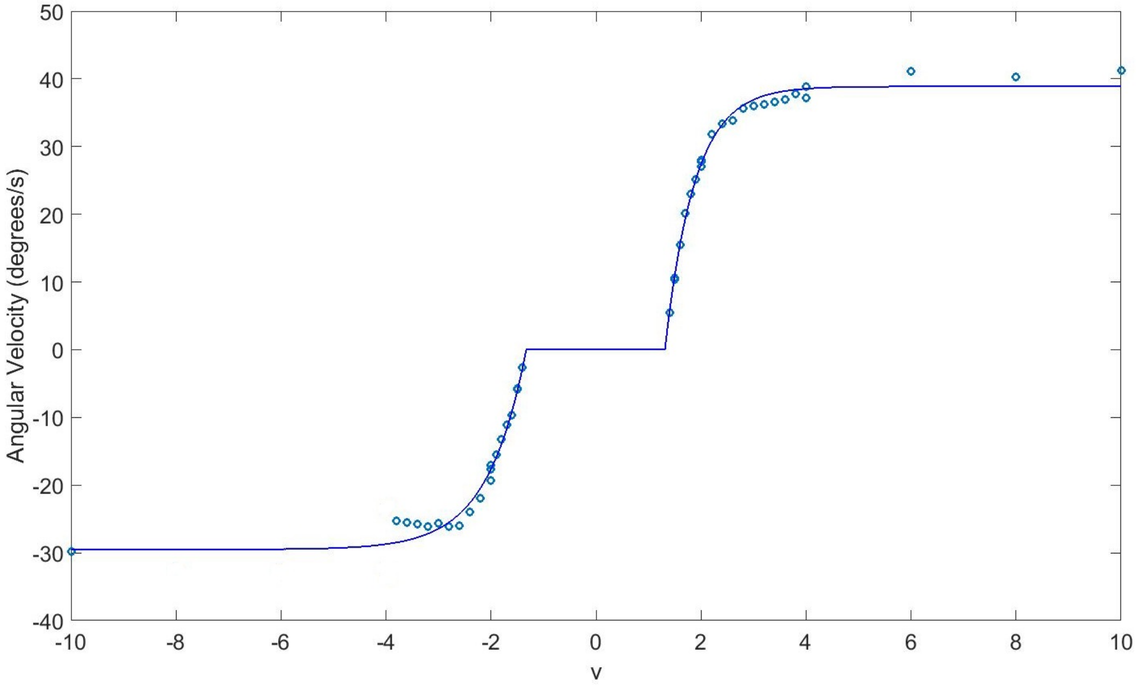


Figure 6.3: *Angular velocity (degrees/s) against input magnitude (voltage to the NI-CFP), showing RIV estimates $b_\tau \times v/\Delta t$ for individual step tests (circles) and SDP estimate (solid line) for Joint 2 of the right hand manipulator.*

coefficients that best fit a defined function to the data, in this case the RIV estimates. By denoting $q\{v_k\} = b_\tau\{v_k\} \times v_k$ the SDP model can be expressed as

$$y_k = y_{k-1} + q\{v_{k-\tau}\} \tag{6.2}$$

where

$$q\{v_k\} = \begin{cases} (1-\alpha_1)\, e^{\alpha_2(\alpha_3 - v_k)} & \text{for} \quad v_k < a_3 \\ 0 & \text{for} \quad a_3 \leq u_k \leq a_6 \\ (1-\alpha_4)\, e^{\alpha_5(v_k - \alpha_6)} & \text{for} \quad v_k > \hat{a}_6 \end{cases} \tag{6.3}$$

and α_1,α_4 are the minimum and maximum angular velocity saturation points, α_2,α_5 are curve coefficients and α_3,α_6 are the voltage dead-zone limits, as shown in Table 6.1.

Table 6.1: *Alpha values for each joint, where α_1,α_4 are the minimum and maximum angular velocity saturation points, α_2,α_5 are curve coefficients and α_3,α_6 are the voltage dead-zone limits.*

	α_1	α_2	α_3	α_4	α_5	α_6
Joint 1	-0.5448	1.8702	-1.2739	0.4101	1.6841	1.1454
Joint 2	-0.2951	1.3560	-1.3298	0.3887	1.7710	1.3124
Joint 3	-0.2474	1.8918	-1.2618	0.3409	1.6934	1.1993
Joint 4	-0.7808	2.8107	-1.4426	0.6048	1.0569	1.1650

As a validation step to confirm these α values the same exponential functions are used and the model output compared with the raw data from the step tests, the sum of the least squares output error is determined and the mean of these errors is used as the objective function of a ‘fminsearch’ in Matlab, to find optimised α coefficients.

6.1.4 Validating SDP model

To check that the developed SDP model was providing an accurate representation of the physical system, a set of open loop pseudo random voltage inputs were used with both the SDP model and the real hardware. Two sets of tests were carried out for each joint, for the first the voltage amplitude was limited to keep it out of the saturation region. This meant that as long as the input wasn't in the deadband each change in the input would have an effect on the joint. If the input wasn't limited it would likely spend a lot of time in the saturation region, where input changes have no effect and so the results would be less meaningful. The second set of tests kept the amplitude limit but added a second limit so that the input couldn't be in the ±0.5v region, this reduced the amount of inputs in the deadband without totally removing it. Both sets of tests were carried out for five minutes, with the pseudo random multi level voltage input switching at varying frequency. Figure 6.4 shows an example of the validation results for joint 2 of the right hand manipulator. As can be seen for joint 2 example the SDP model output is very close to the experimental output, particularly when the joint is moving. The biggest disparity between the two is when the joint is stationary at the limits. For brevity an example for one joint is shown here with similar results for the other joints contained in Appendix E. Table 6.2 shows the mean squared error values between the SDP model and experimentally measured outputs, for both test types. This value is the average over 3 runs each lasting 5 minutes. These results show that for the first 3 joints the SDP model gives an acceptable and usable output, but joint 4 there is room for improvement. The 3 iterations tested for joint 4 showed a large range of MSE value, ranging from 400 in one run to over 2000 in another run, whereas the other joints were much more consistent across the iterations, this can be seen in the example plots shown in Appendix E . This inconsistency makes modelling joint 4 a challenge, and is part of the motivation for the alternative linear SDP model approach discussed in Section 6.2.6.

6.1.5 Tuning for Inverse Dead-Zone approach

In contrast to the previous work, a formal method of dealing with the dead-zone is developed and embedded within the controller structure. The method used for

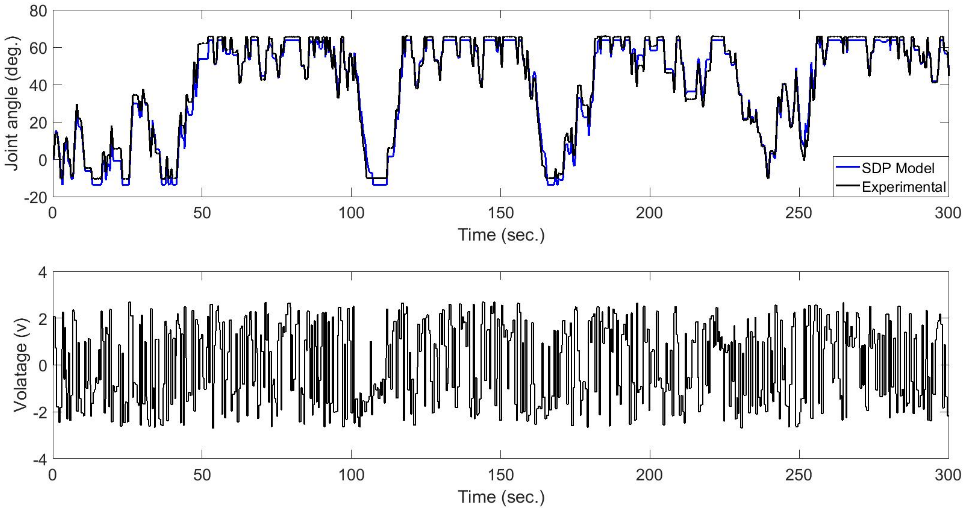


Figure 6.4: *(Top)Comparison of SDP model and experimental output for joint 2 (bottom) pseudo random input voltage.*

removing the dead-zone is similar to the approach in Fortgang et al. [152], in that a fixed parameter inverse dead-zone (IDZ) is applied. The parameters for the IDZ are identified through the open loop tests and the obtained RIV estimates.

A fixed parameter IDZ approach was chosen as it is more straightforward than adaptive inverse approaches, and meant the standard controllers could be used with minimal modification. Adaptive inverse approaches are comprehensively discussed in the thesis by María del Carmen Rodríguez Liñán [154], but left out of this work.

Figure 6.5 [155] shows a schematic representation of the control structure used. Working from left to right, d_k is the desired joint angle, u_k is the controller output, e.g. PI/PIP controller output, v_k is the voltage to the NI-CFP in the range $\pm 10v$, q_k is the joint angular velocity and y_k is the joint angle.

The fixed parameter IDZ is based on a linear estimate of the solid SDP trace shown

Table 6.2: *Mean Squared Error values comparing the SDP model output with the experimentally measured output, each value is the average of 3 iterations of a 5 minute experiment.*

	Joint 1	Joint 2	Joint 3	Joint 4
Limited amplitude	23.828	45.487	6.551	912.122
Limited amplitude and deadband	43.118	10.698	11.921	1358.000

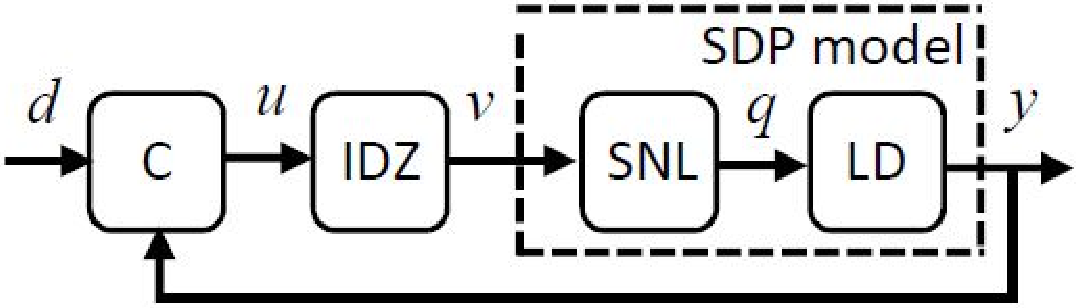


Figure 6.5: *Schematic of control system, in which C, IDZ, SNL and LD represent the linear Controller, the Inverse Dead-Zone control element, the Static NonLinearity and the Linear Dynamics, respectively.*

in Figure 6.3 close to the dead-zone, as shown in Figure 6.6. A control sampling rate of $\Delta t = 0.05s$ is chosen as a compromise between reaction times and keeping the order of the control system reasonably low. Using the same sampling rate as was used for collecting the open loop data, $\Delta t = 0.01s$, would have resulted in a 15th order control system.

Figure 6.6 shows the linearised approximation of the relationship between v_k and

q_k in the range of both v_{min} to $\hat{a}_3$ in the negative direction, and $\hat{a}_6$ to v_{max} in the positive direction. the values of v_{min} and v_{max} are chosen to give a maximum angular velocity of $\approx 20°/s$ (equivalent of 1 degree per sample as shown in Figure 6.6), this was chosen pragmatically as moving any faster is undesirable and could potentially lead to some movement of the platform base due to the weight of the moving manipulator.

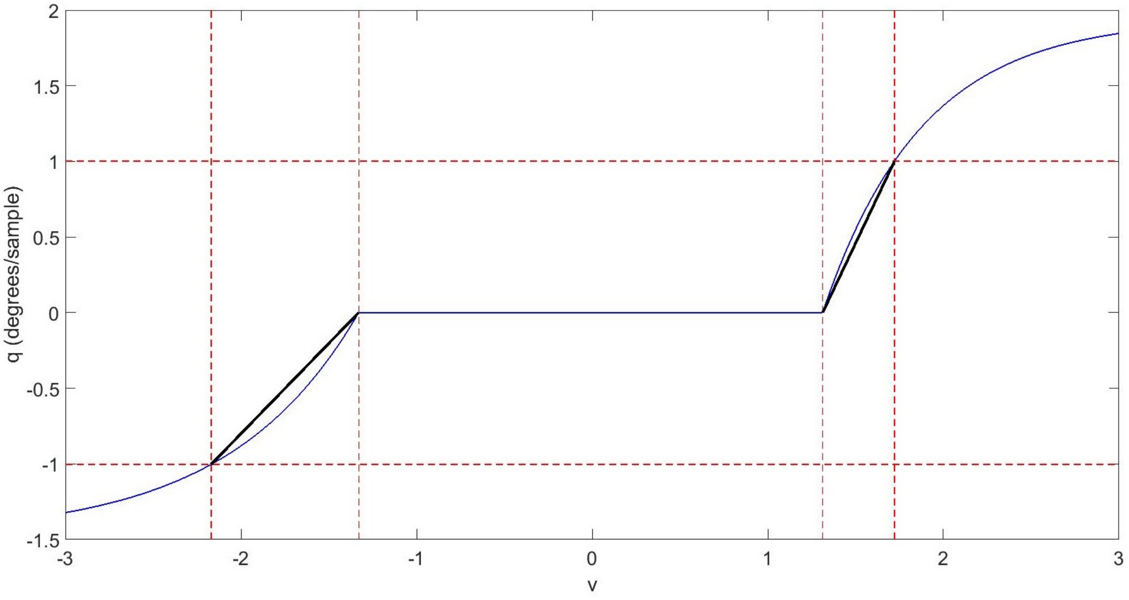


Figure 6.6: *Linear IDZ approximation, showing angular velocity (units of degrees/sample with $\Delta t = 0.05s$) plotted against input voltage to NI-CFP,v. Showing SDP model, thin trace, and linearised model, thick line.*

The Static Non-Linearity, SNL, block of Figure 6.5 is defined by Equation (6.4)

$$q\{v_k\} = \begin{cases} s_n\left(v_k - \hat{a}_3\right) & \text{for} \quad v_k < \hat{a}_3 \\ 0 & \text{for} \quad \hat{a}_3 \leq u_k \leq \hat{a}_6 \\ s_p\left(v_k - \hat{a}_6\right) & \text{for} \quad v_k > \hat{a}_6 \end{cases} \tag{6.4}$$

where the slope coefficients associated with the negative and positive inputs are $s_n = q\{v_{min}\}/\left(v_{min} - \hat{a}_3\right)$ and $s_p = q\{v_{max}\}/\left(v_{max} - \hat{a}_6\right)$ respectively.

The IDZ control element is derived from adapting e.g. Fortgang et al. [152]

$$v_k = \begin{cases} \frac{u_k}{s_n + \hat{a}_3} & \text{for} \quad u_k < -\beta \\ 0 & \text{for} \quad -\beta \leq u_k \leq \beta \\ \frac{u_k}{s_p + \hat{a}_6} & \text{for} \quad u_k > \beta \end{cases} \tag{6.5}$$

where β is a 'chatter' coefficient, set to $\beta = 0.05$, introduced to avoid unnecessary rapid switching of the input between $\hat{a}_3$ and $\hat{a}_6$, as it is known that the inverse dead-zone approach causes chattering in the control signal when small changes occur, (see e.g. [152, 154]). Equation (6.5) aims to cancel the dead-zone, allowing the use of conventional linear controllers.

6.2 Model based controller development

Initially controllers are developed using the SDP model from Equations (6.2) and (6.3) which was implemented in Simulink. Having the SDP model in Simulink allowed controllers to be developed in Matlab/Simulink and controller gains to be tuned based on improving the SDP model output.

This section introduces and develops each controller using the SDP model, the controllers are then compared in Section 6.3.

6.2.1 PI controller

PI controllers are widely used in industry as they combine simplicity with a capability to work effectively in most situations. They are particularly prevalent where the input signal is noisy, as the derivative action of a PID controller is sensitive to measurement noise.

Equation (6.6) shows the standard form of a PI controller

$$U(t) = K_p e(t) + K_I \int e(t).dt \tag{6.6}$$

where $U(t)$ is the controller output, K_p is the proportional gain, K_I the integral gain and $e(t) = d(t) - y(t)$ is the error between the set-point, d and the measured joint angle y.

As well as being the simplest controller that would potentially give a satisfactory response, starting with a PI controller also gives a current industry standard baseline to compare other controllers against.

Figure 6.7 shows the Simulink model used for tuning the controller gains, containing the controller, the input scaling and the SDP model of the joint. The controller block is a PI controller, as shown in Figure 6.8. This controller uses the existing scaled input approach to deal with the dead-zone, as mentioned previously. As such the output of the PI controller, U(t), is in the range of ±100, and is scaled to the range identified for each joint using Equations (6.7) and (6.8), where v_k is the voltage to the NI-CFP, with parameters shown in Table 6.3. This scaled voltage signal is then input to the SDP joint model, to give a joint angle which is fedback as the process variable input to the controller.

Table 6.3: *Values of the dead-zone, saturation points and scaling factors for the existing scaled input in Equations* (6.7) *and* (6.8).

	Deadband		Saturation		Scaling	
	D_n	D_p	S_n	S_p	f_n	f_p
Joint 1	-1.200	1.300	-2.000	3.000	0.0117	0.0119
Joint 2	-1.360	1.300	-2.300	2.300	0.0117	0.0119
Joint 3	-1.200	1.220	-2.000	1.900	0.0117	0.0119
Joint 4	-1.210	1.270	-1.900	2.200	0.0117	0.0119

$$v_k = \begin{cases} D_p + U(t)f_p & \text{for} \quad U(t) > 0 \\ 0 & \text{for} \quad U(t) = 0 \\ D_n - U(t)f_n & \text{for} \quad U(t) < 0 \end{cases} \tag{6.7}$$

$$v_k = \begin{cases} S_n & \text{for} \quad v_k \leq S_n \\ v_k & \text{for} \quad S_n < v_k < S_p \\ S_p & \text{for} \quad v_k \geq S_p \end{cases} \tag{6.8}$$

The PI controller gains were tuned by hand, inspecting the model output with respect to set point following. Values from previously implemented controllers were used as a starting point and the values adjusted from there. The final values of the PI controller gains are shown in Table 6.4.

The results shown in Figures 6.9 and 6.10 show the model output for the four

Table 6.4: *PI controller gain values.*

	Joint 1	Joint 2	Joint 3	Joint 4
K_P	2	9	2	2
K_I	0.15	1	0.2	0.1

investigated joints of the right hand manipulator. The results show that the PI controller can give basic set point following, however performance is better for the sine wave following than the step inputs, where for two of the joints the output oscillates around the set point. This shows the PI controller performs better for smaller movements, i.e. the small changes used to follow the sine curve, as opposed to large step changes.

The oscillation around the set point for two of the joints could be the result of several factors, the most likely is that the scaling used isn't accurate so that the control signal is operating in the deadband or so far from the deadband that small low velocity movement isn't possible. The inaccuracy in the scaling method used is part of the motivation for developing the IDZ scaling approach.

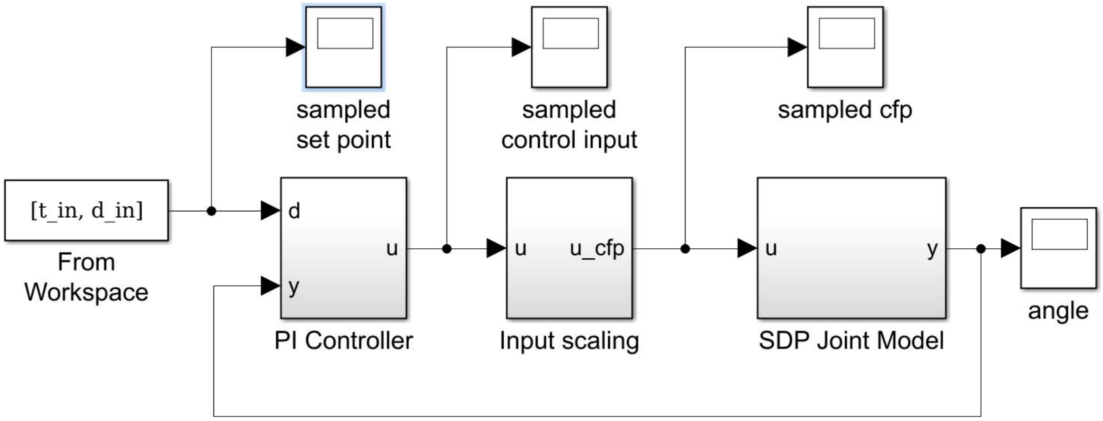


Figure 6.7: *Overview of PI controller in Simulink, showing PI controller block and SDP joint model block with feedback loop.*

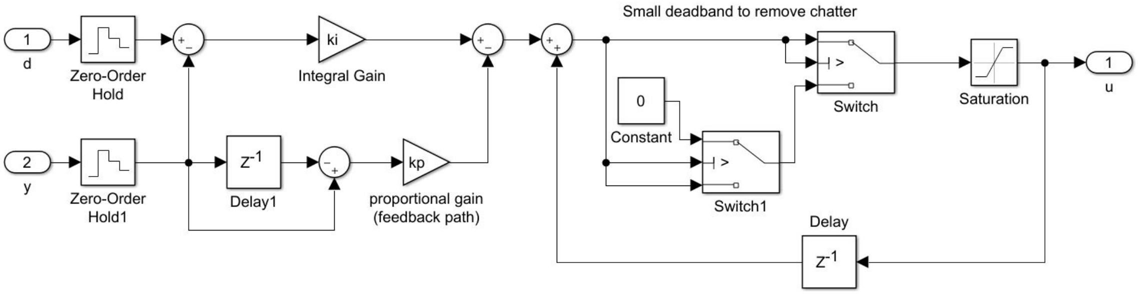


Figure 6.8: *Detailed view of PI controller implementation in Simulink.*

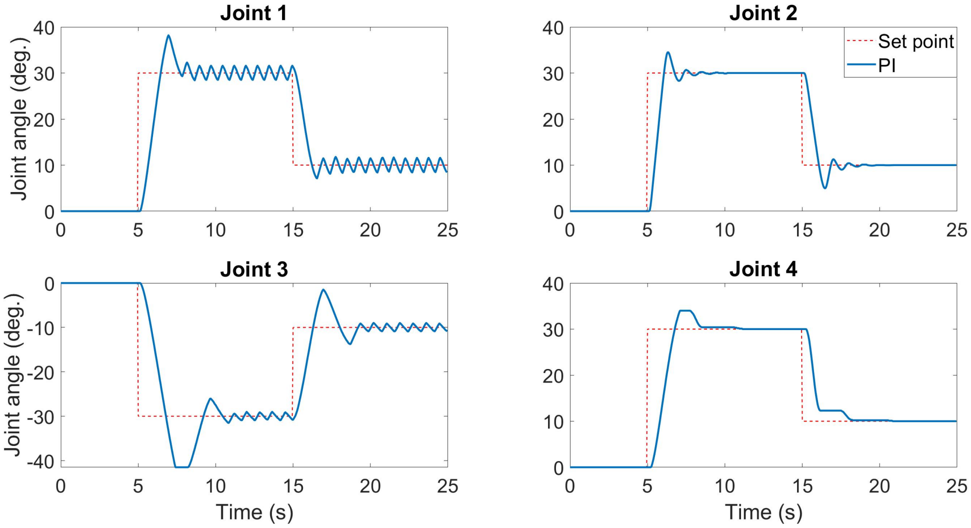


Figure 6.9: *SDP model output for the PI controller set point following, for a step input.*

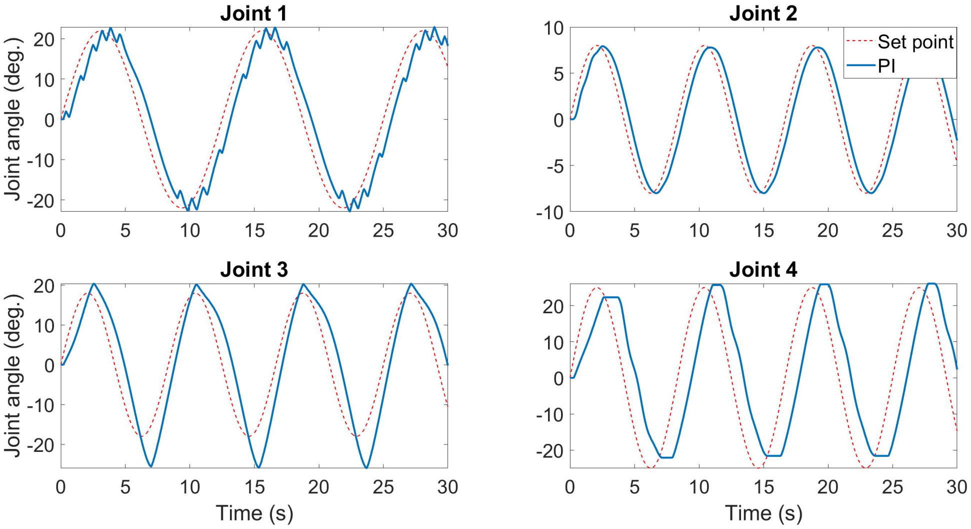


Figure 6.10: *SDP model output for the PI controller set point following, for a sine input.*

6.2.2 PIP controller

Proportional Integral Plus, PIP, control is a logical extension of PI/PID control, with inherent model based predictive control action. It exploits the power of state variable feedback therefore is inherently more flexible and sophisticated than PI or PID controllers.

To develop a PIP controller a linearised version of the model is needed. From the open loop data gathered a linear single input single output discrete time transfer function was identified for each joint.

Details of the general approach can be found in e.g. [11,156], here a worked example is shown for joint 2 of the right hand manipulator, where using a sampling rate of $\Delta = 0.05$ results in a time delay of $\tau = 3$, giving the transfer function:

$$y_k = \frac{b_3 z^{-3}}{1 + a_1 z^{-1}} u_k \tag{6.9}$$

where y_k is the measured joint angle, u_k is the voltage input and z^{-i} is the backward shift operator $z^{-i} y_k = y_{k-i}$. This can be represented in the form of a difference equation

$$y_k = -a_1 y_{k-1} + b_3 u_{k-3} \tag{6.10}$$

Introducing the integral of error, z_k

$$\begin{aligned} z_k =& \frac{1}{1 - z^{-1}} \left(y_{dk} - y_k \right) \\ z_k =& z_{k-1} + y_{dk} - y_k \end{aligned} \tag{6.11}$$

where y_{dk} is the desired joint angle.

Substituting Equation (6.10) into Equation (6.11) gives

$$z_k = z_{k-1} + y_{dk} + a_1 y_{k-1} - b_3 u_{k-3} \tag{6.12}$$

putting this into non-minimal state space (NMSS) form, gives the state equation

$$\begin{bmatrix} y_k \\ u_{k-1} \\ u_{k-2} \\ z_k \end{bmatrix} = \begin{bmatrix} -a_1 & 0 & b_3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a_1 & 0 & -b_3 & 1 \end{bmatrix} \begin{bmatrix} y_{k-1} \\ u_{k-2} \\ u_{k-3} \\ z_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u_{k-1} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} y_{dk} \tag{6.13}$$

and the observation equation

$$y_k = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_k \\ u_{k-1} \\ u_{k-2} \\ z_k \end{bmatrix} \tag{6.14}$$

The state variable feedback (SVF) control law is in the usual format of

$$u_k = -K\boldsymbol{x}_k \tag{6.15}$$

where K is the SVF control gain vector, substituting this in gives

$$u_k = -\begin{bmatrix} f_0 & g_1 & g_2 & -k_I \end{bmatrix} \begin{bmatrix} y_k \\ u_{k-1} \\ u_{k-2} \\ z_k \end{bmatrix} \tag{6.16}$$

which is multiplied out and has Equation (6.11) substituted in for z_k to give

$$u_k = -f_0 y_k - g_1 u_{k-1} - g_2 u_{k-2} + \frac{k_I}{1 - z^{-1}} (y_{dk} - y_k) \tag{6.17}$$

This can be represented in block diagram form as shown in Figure 6.11. Following some block diagram manipulation, this can be rearranged into the standard PIP control structure, as shown in Figure 6.12.

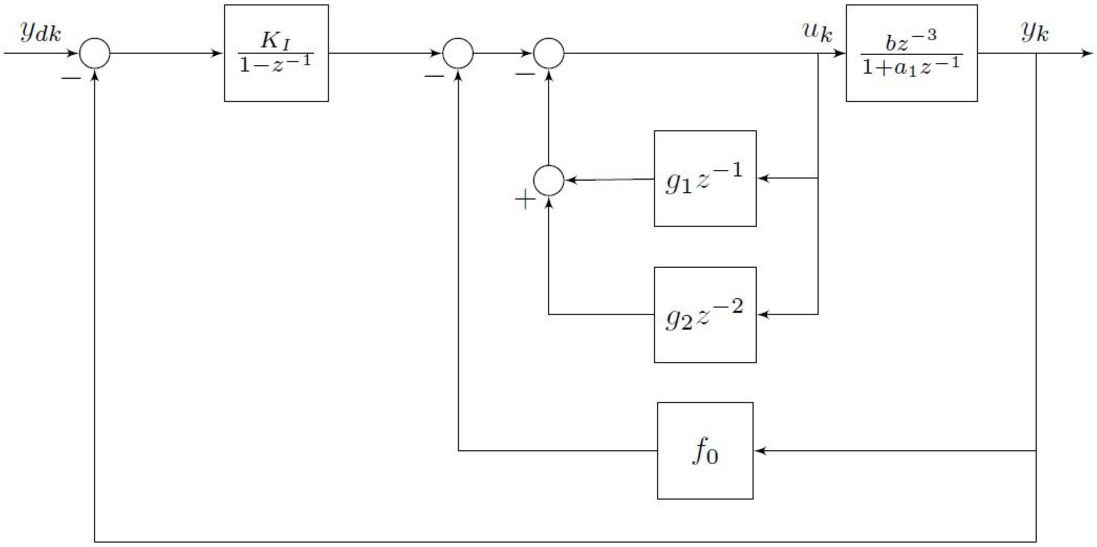


Figure 6.11: *Block diagram representation of Equation* (6.17).

From the block diagram in the PIP form shown in Figure 6.12 the closed loop transfer function can be identified, as shown in Equation (6.18) by following the block diagram simplification rules.

$$\frac{y_k}{y_{dk}} = \frac{Nc}{Dc} \tag{6.18}$$

where

$$Nc = k_I Z^{-3}$$

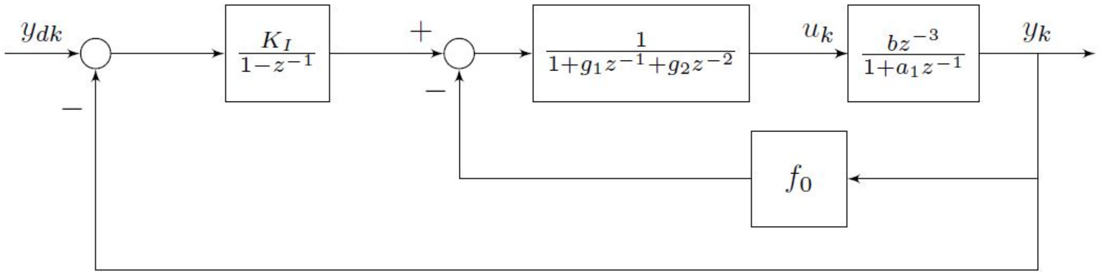

y_{dk}
$\frac{K_I}{1-z^{-1}}$
+
−
−
$\frac{1}{1+g_1z^{-1}+g_2z^{-2}}$
u_k
$\frac{bz^{-3}}{1+a_1z^{-1}}$
y_k
f_0

Figure 6.12: *Block diagram rearranged into standard representation of PIP control structure.*

$$Dc = -\left(a_1g_2 + f_0b\right)z^{-4} + \left(g_1 - 2g_2 + f_0 + k_I\right)z^{-3}+$$

$$\left(a_1g_1 + g_2 - a_1 - G - 1\right)z^{-2} + \left(a_1 + g_1 - 1\right)z^{-1} + 1$$

This closed loop transfer function in PIP form can then be used to find the unknown control gains. There are four unknown control gains, g_1, g_2, f_0 and k_I and four variables in the state equation therefore the problem can be solved and is controllable. The developed PIP controller was implemented in Simulink, as shown in Figure 6.13. The four control gains discussed above are contained in the Simulink model where 'PIP_K' represents K_I, 'PIP_f' represents f_0, and 'PIP_g' is a vector $[1, g_1, g_2]$.

For tuning the controller gains of the PIP controller an optimal control approach is taken, see Chapter 5 of [11], where the control gain vector is chosen to minimise a linear quadratic, LQ, performance criterion. It has been shown in [157] that selection of the LQ weights can in practice always give good, robust, closed loop performance. The approach is implemented using the PIPOPT function from the CAPTAIN toolbox [153] in Matlab. Here three scalar weighting values are chosen, W_y, W_u and W_e, that are used to form the partial-weightings $q_y = W_y/n$, $q_u = W_u/m$ and $q_e = W_e$ respectively, that act on the output, input and integral-of-error variables in the NMSS vector. These three scalar weighting values are chosen to achieve the desired closed-loop performance. Further details are given in [11], specifically in chapter 5.4. The weighting values are chosen by trial and error to achieve satisfactory behaviour of the SDP model response. A pole assignment approach was also trialled but the decision was made to use the LQ approach utilising the CAPTAIN toolbox. As the focus here was on creating a controller that gave satisfactory performance for testing the assisted teleoperation interface developed in Chapter 5 a comparison of different approaches was not carried out and the LQ approach was chosen pragmatically, but due to the SVF formulation any SVF design process could be used.

6.2.3 PIP controller with no scaling

The PIP controller was initially implemented with no scaling to handle the dead-zone, so the controller output was directly used as the CFP voltage. The control gains were found using an LQ design approach, as discussed above, using trial and error to find a satisfactory model response. Table 6.5 shows the controller gain values for the PIP controller with no output scaling, for each joint.

The lack of function to deal with the dead-zone resulted in poor performance with a lot of large rapid oscillations, however despite the oscillations the controller did achieve basic set point following, as can be seen in Figures 6.13 and 6.14 for a step and sine input respectively. This shows that the PIP controller has potential but that the dead-zone does need dealing with.

Table 6.5: *Control gains for PIP controller with no scaling to handle the dead-zone and with the old scaled input approach. Where a = -1, b = [0,0,1] and c = [0,0,0.0583].*

	PIP No scaling	PIP scaled input
Joint 1		
f_0	3.3822	2.3332
$[1, g_1, g_2]$	[1, 0.2998, 0.3382]	[1, 0.1283, 0.1360]
K_I	0.3844	0.1326
PIPOPT weighting	[a, b, 1, 5, 1]	[a, c, 1, 50, 0.1]
Joint 2		
f_0	5.2156	5.8896
$[1, g_1, g_2]$	[1, 0.4417, 0.5216]	[1, 0.1130, 0.1190]
K_I	0.7989	0.2988
PIPOPT weighting	[a, b, 5, 5, 0.01]	[a, c, 10, 100, 1]
Joint 3		
f_0	2.7852	5.0802
$[1, g_1, g_2]$	[1, 0.2506, 0.2785]	[1, 0.1320, 0.1401]
K_I	0.2788	0.2960
PIPOPT weighting	[a, b, 1, 10, 0.1]	[a, c, 10, 100, 1]
Joint 4		
f_0	1.5086	4.2803
$[1, g_1, g_2]$	[1, 0.1382, 0.1406, 0.1415, 0.1509]	[1,0.1844, 0.2000, 0.2155, 0.2311]
K_I	0.0931	0.2883
PIPOPT weighting	[a, b, 1, 100, 1]	[a, c, 10, 100, 0.1]

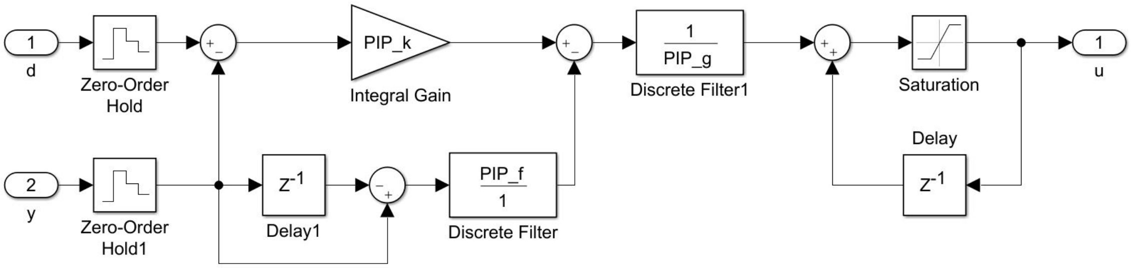


Figure 6.13: *Detailed view of PIP controller implementation in Simulink.*

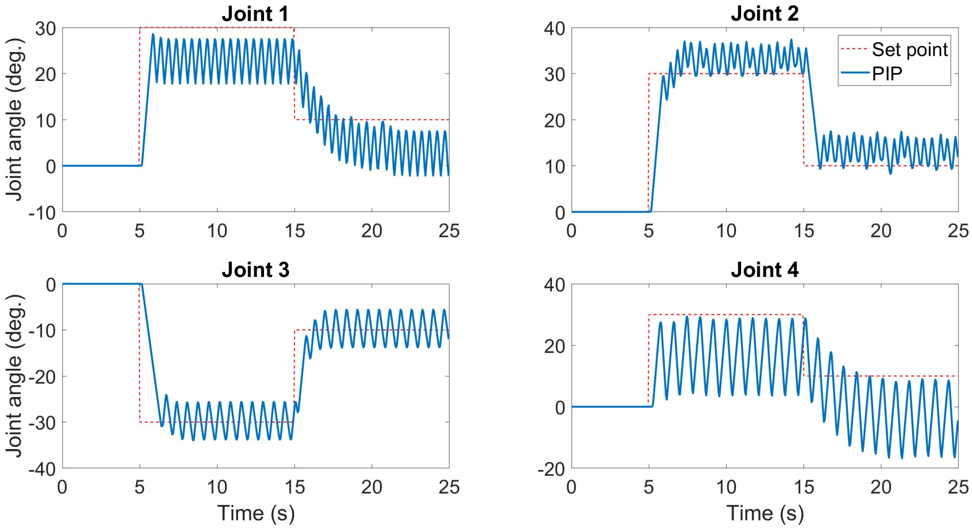


Figure 6.14: *SDP model output of PIP controller set point following, with no scaling to deal with the dead-zone, for a step input.*

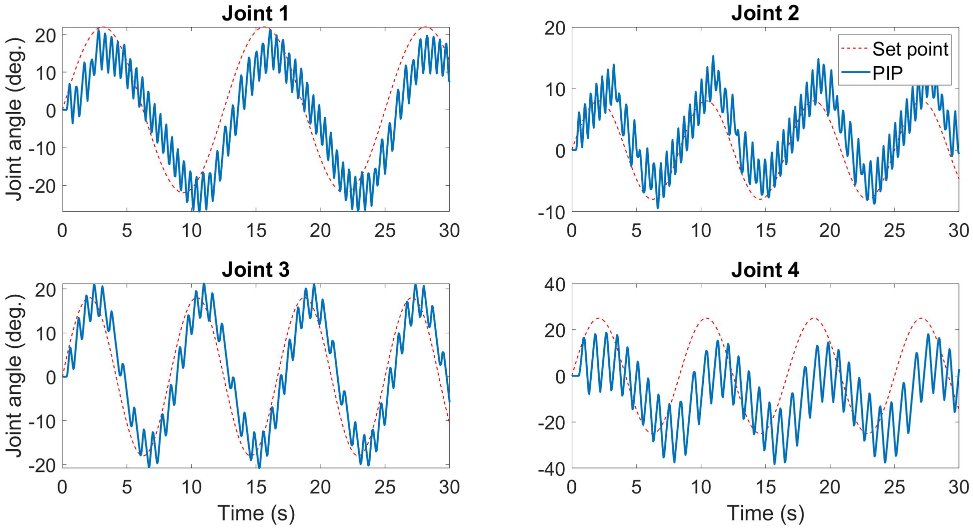


Figure 6.15: *SDP model output of PIP controller set point following, with no scaling to deal with the dead-zone, for a sine input.*

6.2.4 PIP controller with scaled input

Following initial tests with the PIP controller, the same controller (but with different control gains) was then developed with the addition of the previously used scaled input to handle the dead-zone, giving the same structure as shown in Figure 6.7 but with a PIP rather than PI controller. New control gains were needed as the scaling has an input in the range of $\pm 100V$, so the controller output needed to be in this range, whereas without the scaling the controller output was in the range of $\pm 10V$ so it could be directly used as the CFP voltage. The new control gains are shown in Table 6.5.

Figures 6.16 and 6.17 show the results of the PIP controller with the scaled input for step and sine results respectively. As can be seen the inclusion of a method to deal with the dead-zone has led to a large improvement in performance. For both inputs set point tracking has improved and oscillation reduced, although not entirely disappeared. There is still room for improvement, but the importance of including a method to deal with the dead-zone is clearly shown.

These results have a lot of similarity to the PI controller using the same scaling method to deal with the deadzone (figs. 6.9 and 6.10). For both the PI and PIP using this scaling approach joints 1 and 3 show oscillation around the set point in the step tests, the two controllers displaying the same behaviour on these two joints further supports an issue with the scaling approach being the problem. The advantage of the PIP over the PI controller is most clearly seen in the reduced overshoot, in joint 2 for example the overshoot is completely removed. Comparing the PIP with the PI responses, the PIP controller has improved the low frequency system response, but negatively effected the high frequency system response, as can be seen most clearly in the increased delay in the sine wave following of joint 2.

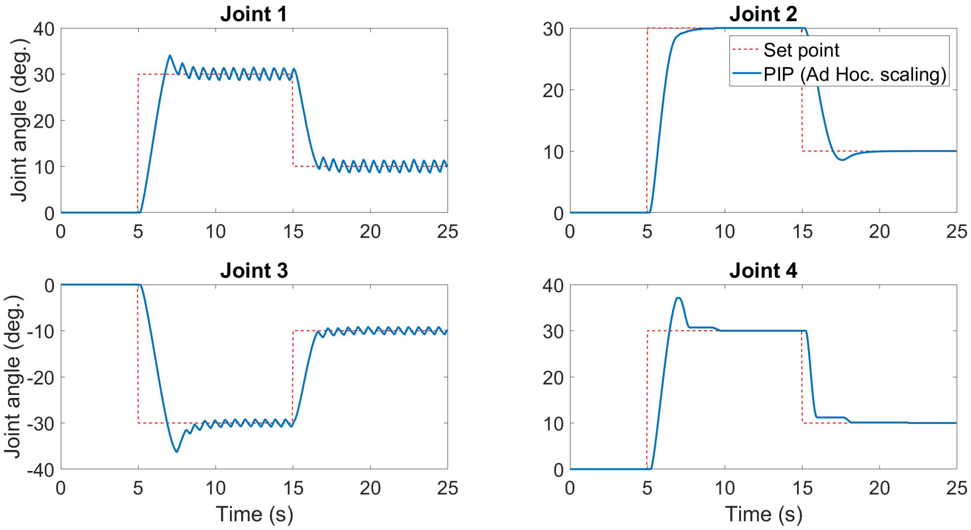


Figure 6.16: *SDP model output of PIP controller set point following, utilising the old scaled input approach, for a step input.*

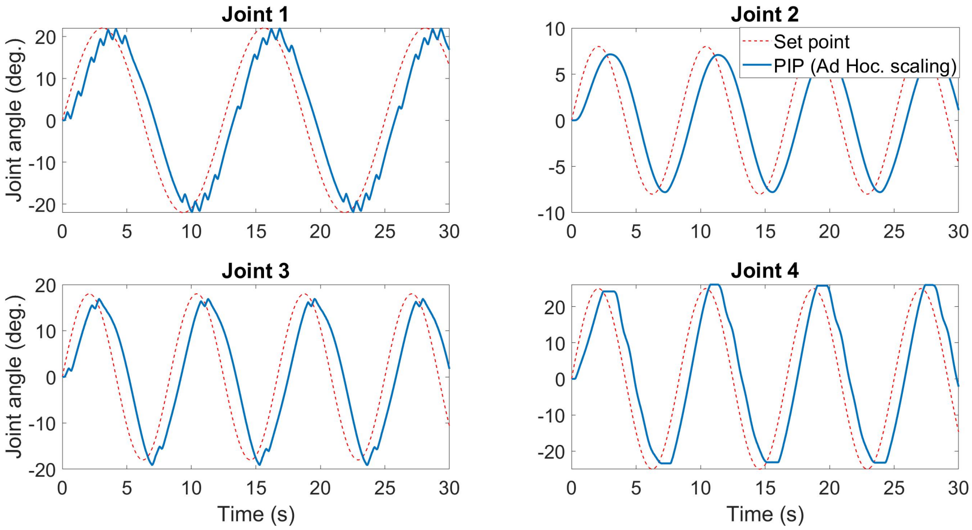


Figure 6.17: *SDP model output of PIP controller set point following, utilising the old scaled input approach, for a sine input.*

6.2.5 PIP IDZ controller

The next controller tested combined the PIP controller with the IDZ function identified during the SDP model development, replacing the previously used scaled input method of dealing with the dead-zone. This controller took the form of the block diagram in Figure 6.5, which when implemented in Simulink is shown Figure 6.18.

Introducing the IDZ function meant that the controller gains again needed adjusting, new gain values were found using the same LQ design approach as previously used.

To demonstrate the advantage of the methodical model based approach to controller design, and the ease of tuning the performance, three different sets of controller gains were found. One that was optimised for step inputs, one for trajectories with a sine wave used as an example, and the final one was a compromise between the

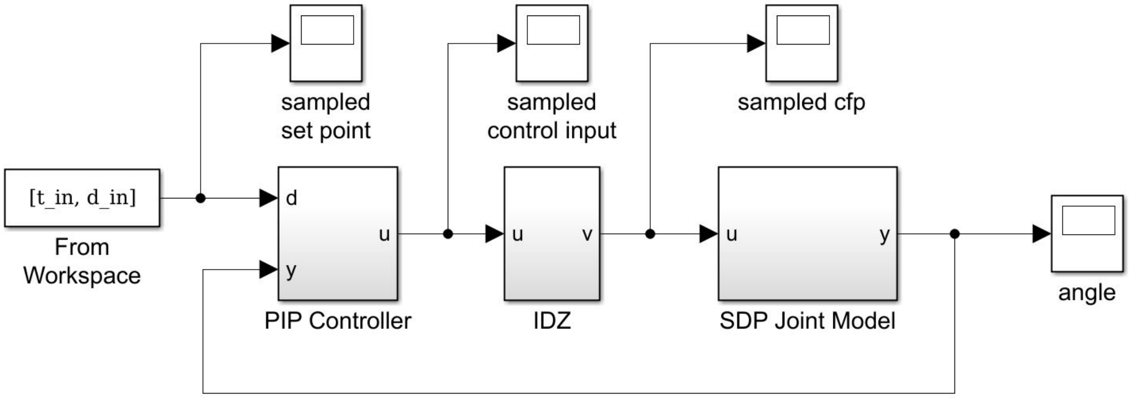


Figure 6.18: *Overview of PIP IDZ controller in Simulink, showing PIP controller, IDZ and SDP joint model blocks with feedback loop.*

two, aiming to give good overall performance. The control gain values for the three optimisations are shown in Table 6.6 along with the LQ (PIPOPT) weights.

Step inputs were chosen as the developed assisted teleoperation system currently provides step inputs to each joint For step inputs the controller was tuned to eliminate overshoot, creating a slower response, but one which reaches the set point smoothly. Overshoot is undesirable as for example it could result in the manipulator hitting objects in the workspace. Trajectory following is chosen as it as an obvious development of the system, providing target trajectories rather than just final target positions, and would be needed in other applications such as welding, or possibly if a collision avoidance algorithm was included. Sine waves were used as an example to give movement in both directions at changing velocities, testing movement in both directions is important as it is known the joint behaviour is asymmetrical. The tuning for the trajectory following was done in such a way to minimise delay, and follow the input as closely as possible.

An example of the performance of the PIP IDZ controller is shown in Figure 6.19 for a step input, and Figure 6.20 for a sine wave input, both showing the SDP model output. These results clearly show the differences in the different tuning of the controller gains, the step tests clearly show a large overshoot for the faster trajectory following tuning, and no overshoot with the step input tuning. The sine input results show that whilst all three controllers follow the input there is a noticeable delay with the slower step input tuning.

These plots show that the high frequency system response of the controller tuned for trajectory following (tuning ‘b’) is good, with joints 1 to 3 showing close and accurate sine wave following, with very low delay in response to the step input and short rise times. However the lower frequency response is less good, with large overshoots on the step inputs and slow settling times. The ‘c’ tuning, tuned for step responses shows the opposite behaviour, with no overshoot on the step tests but slower and less accurate sine wave following.

Table 6.6: *PIP control gains for the three different optimisations chosen, for use with the PIP IDZ controller. Where a = -1, b = [0,0,1].*

	(a)	(b)	(c)
	General Performance	Trajectory Following	Step Input
Joint 1			
f_0	0.1536	0.6351	0.1648
$[1, g_1, g_2]$	[1, 0.1443, 0.1536]	[1, 0.5274, 0.6351]	[1, 0.1555, 0.1648]
K_I	0.0093	0.1077	0.0092
PIPOPT weighting	[a, b, 0.01, 100, 0.1]	[a, b, 20, 1000, 10]	[a, b,1, 1000, 5]
Joint 2			
f_0	0.6925	0.3371	0.1774
$[1, g_1, g_2]$	[1, 0.6000, 0.6925]	[1, 0.2986, 0.3371]	[1, 0.1682, 0.1774]
K_I	0.0925	0.0385	0.0092
PIPOPT weighting	[a, b, 1, 1000, 1]	[a, b, 2, 1000, 1]	[a, b, 0.1, 1000, 10]
Joint 3			
f_0	0.2485	0.5276	0.1774
$[1, g_1, g_2]$	[1, 0.2286, 0.2485]	[1, 0.4481, 0.5276]	[1, 0.1682, 0.1774]
K_I	0.0199	0.0795	0.0092
PIPOPT weighting	[a, b, 0.05, 100, 1]	[a, b, 1, 100, 1]	[a, b, 0.1, 1000, 10]
Joint 4			
f_0	0.7259	0.6865	0.1957
$[1, g_1, g_2, g_3, g_4]$	[1, 0.4978, 0.5738, 0.6499, 0.7259]	[1, 0.4481, 0.5276, 0.6071, 0.6865]	[1, 0.1682, 0.1774, 0.1865, 0.1957]
K_I	0.0760	0.0795	0.0092
PIPOPT weighting	[a, b, 1, 100, 10]	[a, b, 1, 100, 1]	[a, b, 0.1, 1000, 10]

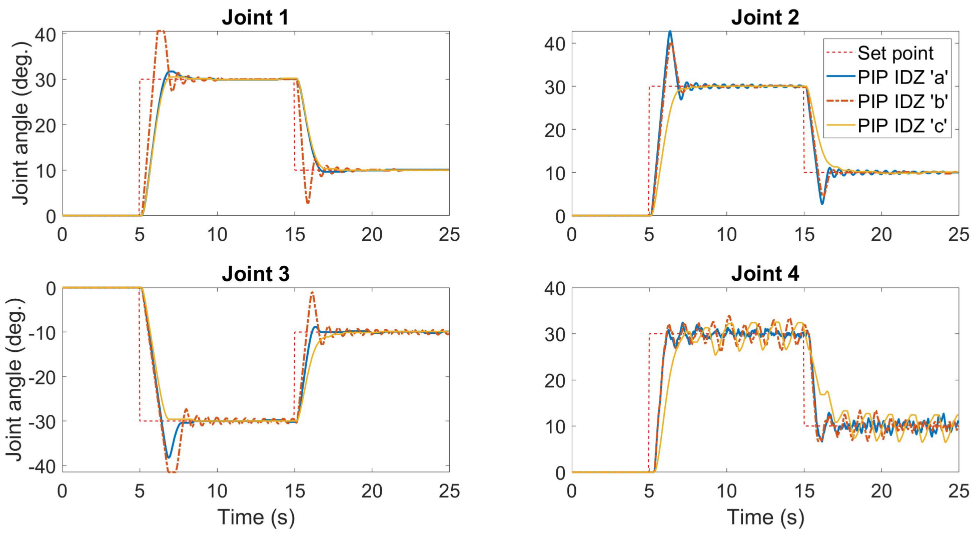


Figure 6.19: *SDP model output of PIP IDZ controller set point following, for a step input, for the three differently tuned controllers, where 'a' is the compromise tuning, 'b' is the trajectory following tuning and 'c' is the step input tuning.*

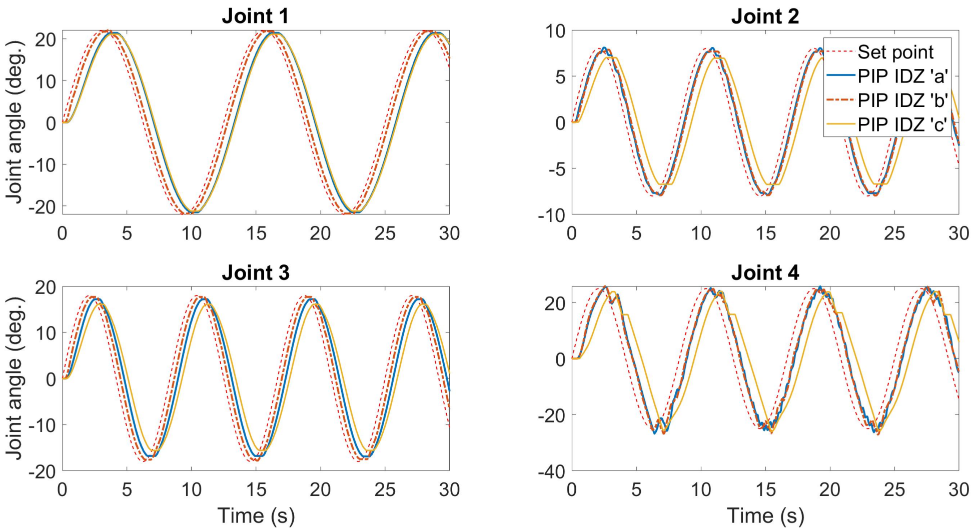


Figure 6.20: *SDP model output of PIP IDZ controller set point following, for a sine input, for the three differently tuned controllers, where 'a' is the compromise tuning, 'b' is the trajectory following tuning and 'c' is the step input tuning.*

6.2.6 Joint 4 special case

Testing showed that whilst the other joints were responding reasonably well, with simulation output closely matching measured experimental output, joint 4 displayed a big disparity between the simulation and measured behaviour, with joint control being poor. This can be seen in Figure 6.21, where for joints 1 to 3 the model and experimental results mostly overlap, but there is a noticeable difference for joint 4. Even just looking at simulation results joint 4 had worse control than the other joints, as can be clearly seen in Figures 6.19 and 6.21. Although the joint does basically follow the setpoint, there is a lot of large rapid oscillation, moving up to approximately ±4° around the setpoint. The exact reason for this is unknown, but it likely linked to the fact that it is the only hydraulic rotary actuator, with the others all being linear actuators. The rotary actuator has currently been modelled in the same way as the linear actuators, however some of the assumptions made in the modelling may not be valid for the rotary actuator, so future work could be to look at a different modelling process for this joint. As a result of this poor behaviour, different approaches were investigated to find a solution to improving the control.

An approach that provided some improvement was to change the SDP model to use linear functions rather than the exponential functions in Equation (6.3). As before, the SDP model can be expressed as

$$y_k = y_{k-1} + q\{v_{k-\tau}\} \tag{6.19}$$

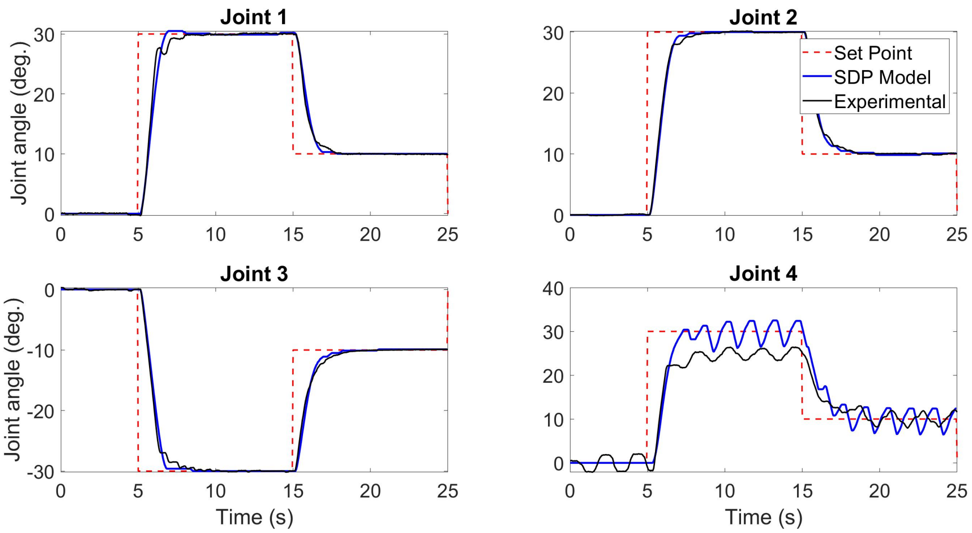


Figure 6.21: *Comparison of SDP model output and experimental measured joint angle for step point following, using tuning 'c' of the PIP controller. The plotted experimental result is the average of 3 iterations.*

where

$$q\{v_k\} = \begin{cases} \beta_1 & \text{for} \quad v_k < \beta_5 \\ \alpha_1 + \alpha_2 v_k & \text{for} \quad \beta_2 \leq v_k \leq \beta_5 \\ 0 & \text{for} \quad \beta_2 < v_k < \beta_4 \\ \alpha_3 + \alpha_4 v_k & \text{for} \quad \beta_4 \leq v_k \leq \beta_6 \\ \beta_1 & \text{for} \quad v_k > \beta_6 \end{cases} \tag{6.20}$$

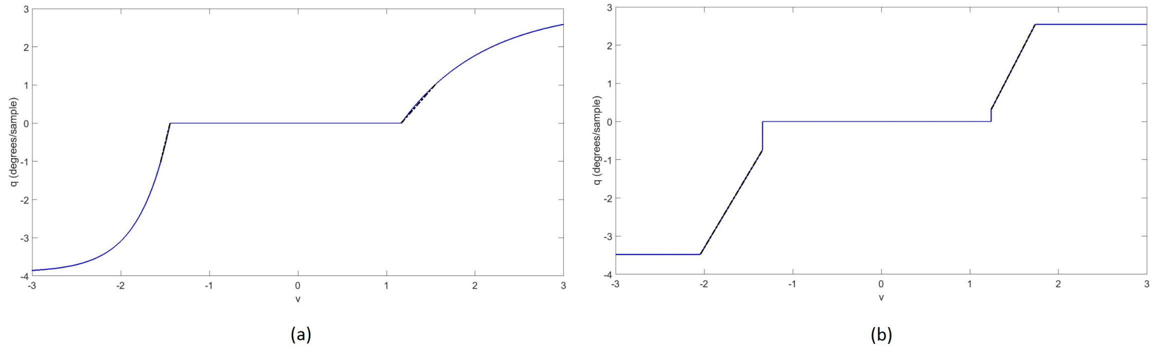
and α_1,α_3 are the intercept points and α_2,α_4 the slope for the negative and positive straight lines respectively, β_1,β_3 are the negative and positive velocity saturation values and β_2,β_4 the deadband voltage values and β_5,β_6 the voltage values at the saturation points. The values used for these parameters are shown in Table 6.7.

Figure 6.22 shows the plot of the CFP voltage, v, against the angular velocity, q,

Table 6.7: *Values of the α and β parameters used in the linear SDP model function for joint 4.*

	1	2	3	4	5	6
α	0.8936	0.7777	-1.0512	0.8983		
β	-0.6955	-1.3400	0.5088	1.2400	-2.0434	1.7366

for both the original SDP model using exponential functions and the new linearised approach, clearly highlighting the difference in the model of the two approaches.



(a)

(b)

Figure 6.22: *Plot of CFP voltage , v against joint angular velocity q in degrees per sample for (a) original SDP model and (b) new linearised approach, showing SDP model, solid trace, and the linearised model, thick dashed trace.*

This linear model approach resulted in improved performance, as shown in Figure 6.23, where it can be seen that using the linear function causes smaller and less frequent oscillations, without having a negative effect on the set point following, for example looking at the step input the rise time remains the same, but the linear model causes less overshoot and less oscillation around the set point. The result is still not ideal however and so further potential improvements were investigated.

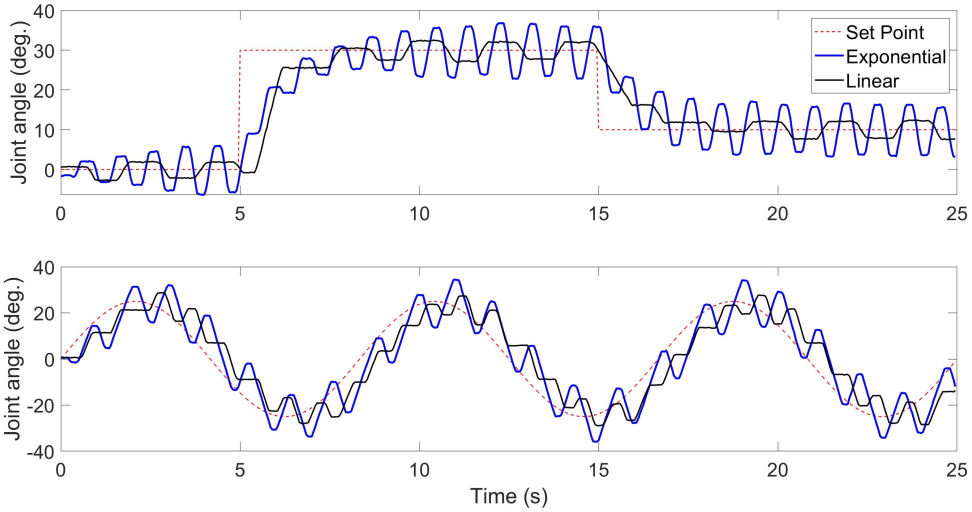


Figure 6.23: *Experimental response of joint 4 for step and sine input following, comparing the use of exponential and linear functions in the SDP model.*

6.2.7 PIP IDZ forward path controller

The next controller developed was a modified version of the PIP IDZ controller, utilising forward path techniques. The forward path controller is a different way of structuring the PIP controller block diagram, which incorporates the estimated transfer function model, see chapter 6 of [11] for a detailed explanation, one way of representing the forward path control structure is shown in Figure 6.24 [156]. The 'pre-compensation' filter is formed from reducing the inner loop of the feedback form PIP controller, using the rules of block diagram analysis. From Figure 6.24 it can be seen that the inner feedback loop is based on the model output rather than the measured output, therefore the estimated model acts as a source of information for the controller.

One of the advantages of a forward path approach like this is that noise on the measured signal does not pass through the feedback filter, instead only passing through the integral component of the controller. As a result the the signal produced by the forward path controller is typically smoother than that of a standard feedback controller. The forward path control structure is also more resilient to input disturbances. A disadvanatge of the forward path approach is that it is often more sensitive to modelling errors, as the model is directly included and cancels the system dynamics.

In theory the control gains used for the feedback and forward path controllers should be the same, as it is just a restructured block diagram. However in practice new control gains needed identifying for the forward path controller as the previous control gains didn't give a satisfactory response. This is due to the developed SDP model not being perfect and there being some mismatch between the model and the real system. The new controller gain values are shown in Table 6.8.

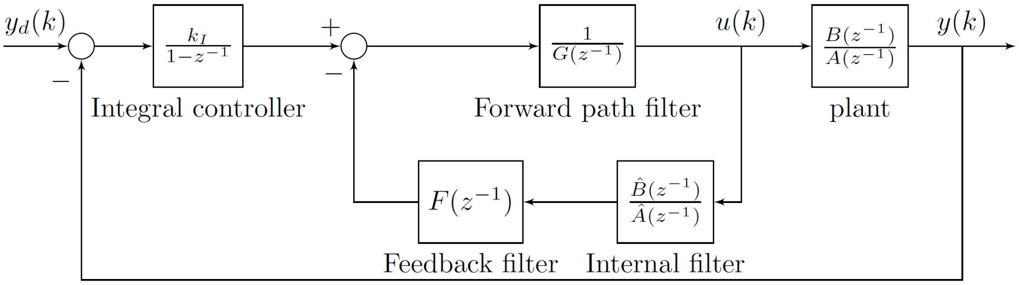


Figure 6.24: *Forward path PIP control structure, with feedback of an internal model.*

The new controller was tested with the same inputs as the other controllers and the results were promising. The bottom right plot of Figures 6.25 and 6.26 show the results for joint 4, as can be seen performance appears improved. The forward path controller structure clearly improves the set point following performance of joint 4,

Table 6.8: *PIP control gains for the three different optimisations chosen, for use with the PIP IDZ FP controller. Where a = -1, b = [0,0,1].*

	(a)	(b)	(c)
	General	Trajectory	Step Input
	Performance	Following	
Joint 1			
PIP_Num	[0.0093, -0.0093]	[0.0795, -0.795]	[0.0030, -0.0030]
PIP_Den	[1, -0.8557, 0.1629, -0.1536]	[1, -0.5519, 0.6071, -0.5276]	[1, -0.8861, 0.1199, -0.1169]
PIPOPT weighting	[a, b, 0.01, 100, 0.1]	[a, b, 1, 100, 1]	[a, b, 0.01, 1000, 7.5]
Joint 2			
PIP_Num	[0.0199, -0.00199]	[0.0279, -0.0279]	[0.0175, -0.0175]
PIP_Den	[1, -0.7714, 0.2683, -0.2485]	[1, -0.7840, 0.3077, -0.2799]	[1, -0.5945, 0.4405, -0.4230]
PIPOPT weighting	[a, b, 0.05, 100, 0.1]	[a, b, 1, 1000, 1]	[a, b, 0.05, 100, 20]
Joint 3			
PIP_Num	[0.0092, -0.0092]	[0.0778, -0.0778]	[0.0029, -0.0029]
PIP_Den	[1, -0.8318, 0.1865, -0.1774]	[1, -0.5281, 0.6276, -0.5498]	[1, -0.8613, 0.1446, -0.1417]
PIPOPT weighting	[a, b, 0.01, 100, 0.1]	[a, b, 1, 100, 5]	[a, b, 0.01, 1000, 15]
Joint 4			
PIP_Num	[0.0065, -0.0065]	[0.0303, -0.0303]	[0.0074, -0.0074]
PIP_Den	[1, -0.8496, 0.0065, 0.0065, 0.0065, 0.0065, 0.1897, -0.1831]	[1, -0.7372, 0.0303, 0.0303, 0.0303, 0.0303, 0.4449, -0.4145]	[1, -0.8055, 0.0074, 0.0074, 0.0074, 0.0074, 0.2387, -0.2313]
PIPOPT weighting	[a, b, 0.05, 1000, 10]	[a, b, 1.2, 1000, 0.5]	[a, b, 0.05, 750, 20]

whilst also providing a small improvement to the performance of the other joints

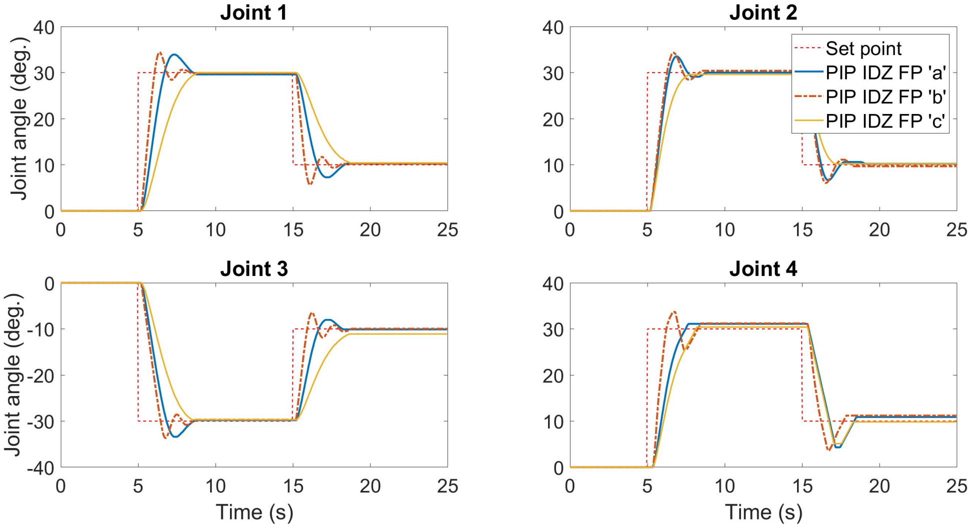


Figure 6.25: *SDP model output of PIP IDZ controller set point following, for a step input, for the three differently tuned controllers, where ‘a’ is the compromise tuning, ‘b’ is the trajectory following tuning and ‘c’ is the step input tuning.*

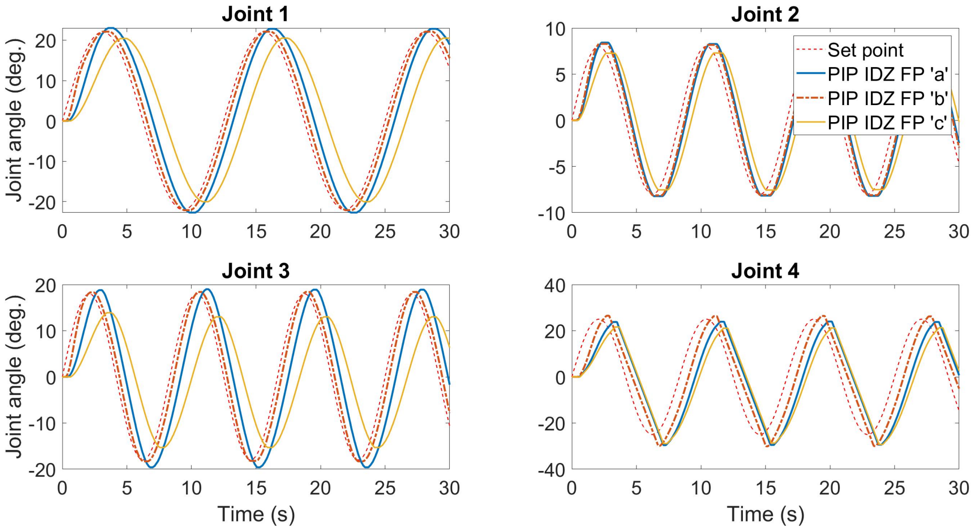


Figure 6.26: *SDP model output of PIP IDZ controller set point following, for a sine input, for the three differently tuned controllers, where 'a' is the compromise tuning, 'b' is the trajectory following tuning and 'c' is the step input tuning.*

6.3 Comparing controller performance

After the controllers had been developed using the SDP model they were each tested on the physical system using the same step and sine inputs. This allows comparison between the different controllers and the between the SDP model and real system performance. All results presented here, and the experimental results in Chapter 7, are the average values from three iterations. This gives a more reliable value and shows repeatability of the system.

Due to the large and rapid oscillations caused by the PIP controller with no scaling, it was decided not to include this in the comparisons as it was clearly the worst so investigating it further would be unproductive. As a result there are eight controllers to compare:

- PI Controller, with scaled input
- PIP Controller, with scaled input
- PIP Controller, with IDZ deadband scaling, and three different sets of gains
- PIP forward path Controller, with IDZ deadband scaling, and three different sets of gains

6.3.1 Step tests

Step inputs were chosen for testing the controllers as the developed assisted teleoperation system currently provides the target joint angles as steps, giving the first angle for the initial position in front of the target, which acts as step from the joints initial position, then giving a second step to go to the final target position.

The step inputs used for testing the controllers were chosen pragmatically so that they cover a good portion of the joints range of motion, and provide a step in both directions, as it is known that the joint behaviour is asymmetrical. To allow comparison the same step input was used consistently for each controller both using the SDP model and later on the real hardware.

Table 6.10 contain the values for comparison of the step input for SDP model results, the best result in each row is shown in bold. Only results useful for a direct comparison between controllers with respect to a step input are included here, with further results shown in Appendix G. Looking at Table 6.10 it can be seen that the PIP IDZ controller in feedback form has the slightly faster rise times for reaching the set point following the step input, whereas the forward path form has the lowest values for percentage overshoot of the set point. The forward path form also has the lowest value for the measure of CFP voltage switching, this is measured by taking the variance of the difference of the CFP voltage output. The difference of the CFP signal gives an array of values measuring the difference between the voltage at each measurement sample, taking the variance of this then gives a measure of how much the CFP signal is switching from positive to negative with a higher value meaning more switching, as such a lower value is desirable.

As well as comparing these controllers against each other they should be checked against the design objectives set out in the introduction of this chapter to ensure the controllers meet the objectives. The first design objective was to have a rise time under two seconds, from Table 6.10 it can be seen that almost all of the controllers meet this objective with some having a much shorter rise time. The only controller that doesn't meet this design objective is the 'c' tuning of the PIP IDZ FP controller, where 3 of the 4 joints have a rise time of over 2 seconds, although they take only slightly longer with the slowest rise time being 2.28 seconds. The second design objective was to have an overshoot of under 10 percent. The different controllers have varied success at meeting this objective with only 2 of them successfully achieving it, the 'c' tuning of both the PIP IDZ and the PIP IDZ FP controllers. The other controllers often have considerably higher overshoots, going up to 43%. As would be expected the controllers with the lowest rise times have the highest overshoots, and the PIP IDZ FP controller with the lowest overshoot has the slowest rise times. The final design objective was to have a steady state error of less than half a degree. Table 6.9 shows the steady state error of each controller after the initial step during the step tests. Most of the controllers didn't reach steady state for all the joints, with the only controller that did being the PIP IDZ FP. For the controllers that did achieve a steady state they were all under the target 0.5° except for joint 4, where both the 'a' and 'b' tuning of the PIP IDZ FP controller reached steady state but with an error of over 1 degree. The 'c' tuning of the FP controller however was the only controller to get all joints to steady state, and with a steady state error below the target 0.5. Overall the 'c' tuning of both the PIP IDZ and the PIP IDZ FP controller come out the best with regards to the design objectives set out in the introduction of this chapter. Taking into account joint 4 which has proven to be troublesome, the forward path controller is overall the best although it doesn't quite meet the objective of having a rise time under two seconds, for the application being considered here accurate movement is more important than speed.

These results highlight the compromise often needed in controller design, i.e. no one controller here is better in every aspect. In this case overshoot is more important than rise time as an overshoot could cause the manipulator to hit something in the workspace. For this example the forward path form of the PIP IDZ controller come out the best as, with the tuning for step inputs, there is 0 overshoot for three of the joints and a very small overshoot for the fourth joint. As a bonus this controller also has the small measure of CFP voltage switching, which was the secondary aim of the controller development, and results in less wear on the hydraulic valves.

An important thing to note is the difference in performance between the PIP IDZ controller's and the PI controller. The PI controller represents an industry standard control approach and the approach that was previously being used on the Brokk-

Table 6.9: *Steady state error of each controller following the initial step movement of the step tests.*

		Joint 1	**Joint 2**	**Joint 3**	**Joint 4**
PI		na	0.00	na	0.00
PIP		na	0	na	0.05
PIP IDZ	**a**	0.10	na	0.20	na
	b	0.00	0.10	na	na
	c	0.10	0.00	0.00	na
PIP IDZ FP	**a**	0.40	0.00	0.20	1.1
	b	0.10	0.40	0.30	1.20
	c	0.00	0.30	0.30	0.4

Hydrolek platform. Therefore improving on the PI controller results suggest the developed controller will give noticeable and useful improvement in performance, therefore being worth the added complexity.

Table 6.10: *SDP model results for PIP controller response to step inputs, comparing rise time, overshoot and a measure of switching of the CFP voltage. The best values are in bold.*

		PIP IDZ			**PIP IDZ FP**			**PIP scaled**	**PI scaled**
		a	**b**	**c**	**a**	**b**	**c**	**input**	**input**
Rise time	**J1**	1.0186	**0.5895**	1.0882	1.061	0.6673	2.2833	1.193	0.9886
	J2	**0.6225**	0.6630	1.2117	0.9055	0.8082	1.5906	1.231	0.7076
	J3	0.9858	**0.9541**	1.1930	1.2046	0.9773	2.1401	1.288	1.2180
	J4 exponential	**0.5014**	0.4763	1.2085				0.890	1.1239
	J4 linear	0.6084	0.5797	1.2395	1.4380	0.6405	2.0375		
Overshoot %	**J1**	6.0201	35.3330	2.0067	14.527	15.0502	**0**	13.667	27.8000
	J2	43.6242	33.8870	0.0000	11.6667	14.3333	**0**	0.000	15.0000
	J3	27.6667	40.6780	0.0000	12.0805	13.4680	0	21.000	38.3333
	J4 exponential	29.0000	37.7926	0.0000				23.667	13.3333
	J4 linear	8.0000	13.0000	8.0000	3.6667	12.3333	**1.333**		
Var of Diff of CFP	**J1**	0.0621	0.2940	0.0464	0.0224	0.0611	**0.0135**	0.6747	0.6109
	J2	0.6569	0.2195	0.0700	0.0507	0.0476	**0.0181**	0.0329	0.2353
	J3	0.0984	0.6206	0.0453	0.0234	0.0797	**0.0136**	0.7124	0.3703
	J4 exponential	0.2545	0.2100	0.0524				0.0184	**0.0179**
	J4 linear	3.1635	1.1910	0.2351	0.0192	0.0237	0.0236		
	Steady State Error								

6.3.2 Sine tests

To test the controllers' trajectory following performance, sine waves were used for each joint. The sine waves were chosen to generate a closed trajectory that covered a large portion of the working area and moved the end effector in 3D space, as shown in Figure 6.27. The complete clsoed loop movement shown in Figure 6.27 is completed in 25 seconds.

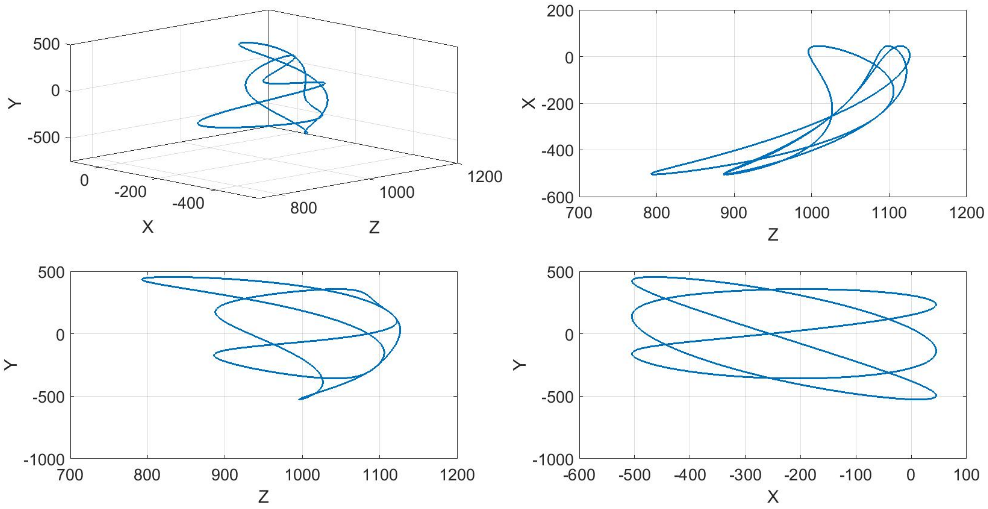


Figure 6.27: *End effector trajectory created by each joint following a sine input, where X, Y and Z are the Horizontal, Vertical and Depth axis respectively.*

The controllers are compared with the sine input in a similar way as they were with the step input. Table 6.11 contains values for comparison of the controllers using the SDP model. The parameters included in these tables are different than the ones used to compare the step tests, as these parameters are more relevant to trajectory following. As before Table 6.11 contains only results useful for a direct comparison between the controllers, with further results presented in Appendix G.

The measures used for comparison are the average euclidean norm error of the end effector position, the mean squared error (MSE) between set point and measured angle for each joint, and the variance of the difference of the CFP voltage, again to measure voltage switching.

Looking at the results in Table 6.11, it can be seen that the PIP IDZ forward path controller produces the majority of the best results, having the smallest mean squared error for 3 of the 4 joints and the lowest CFP voltage switching values. Similarly to with the step results two different controllers come out best depending which metric you look at. In this case the feedback form of the PIP IDZ controller is best when looking at the average euclidean norm of the end effector position, but when looking at the MSE of the individual joints the forward path controller comes out best. The forward path controller also comes out best when looking at the CFP voltage switching. Overall for trajectory following minimising the MSE error between individual joint angles and the target set points is important, therefore again the forward path form of the controller comes out best.

For the trajectory following the improvement over the PI controller is less noticeable than for the step inputs but is an important improvement. Figure 7.10 shows a plot comparing the PI and the forward path PIP IDZ controller, it shows that although the PI controller gives fairly good set point following the PIP IDZ FP controller follows the set point more closely and has a smaller response time to changes in the set point, as well as just generally giving a cleaner response.

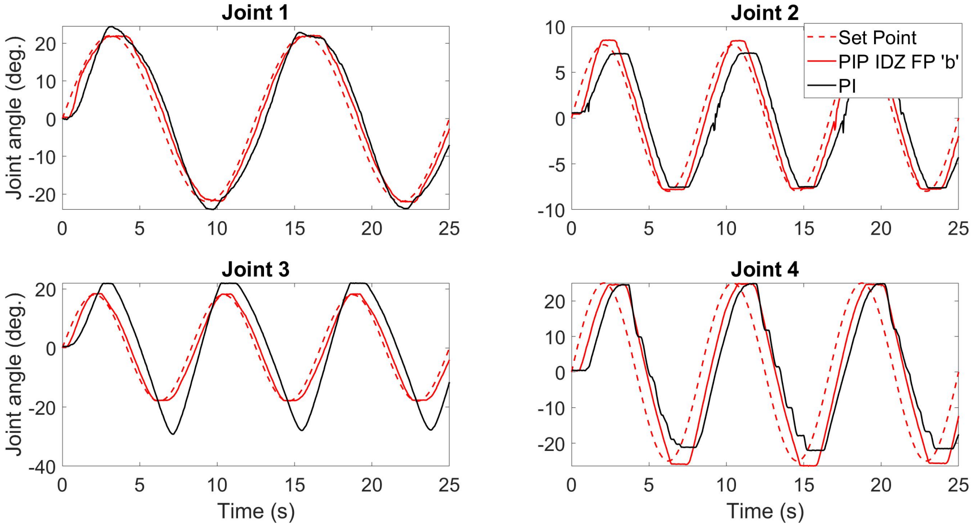


Figure 6.28: *Comparison of PI and PIP IDZ FP controller for a sine input.*

Table 6.11: *SDP model results for PIP controller response to sine inputs, showing euclidean norm error for end effector, MSE between set point and measured angle for each joint and a measure of CFP voltage switching. The best values are in bold.*

		PIP IDZ (J4 Exponential)			**PIP IDZ FP**			**PIP scaled**	**PI scaled**
		a	**b**	**c**	**a**	**b**	**c**	**input**	**input**
Avg. Euclidian Norm									
	J4 exponential	161.133	**77.242**	189.242				181.0945	142.0093
	J4 linear				163.553	78.054	276.28		
MSE set point following	**J1**	37.903	5.044	42.84	31.848	**4.921**	133.436	39.1107	24.5504
	J2	6.018	3.362	13.136	5.495	**2.751**	12.141	15.1334	3.573
	J3	65.992	9.929	65.992	54.567	**7.057**	158.522	62.4317	47.08
	J4 exponential	39.209	**32.698**	166.593				97.8049	161.408
	J4 linear	57.320	49.866	193.897	190.825	68.960	246.93		
Var of Diff of CFP	**J1**	**0.0276**	0.0322	**0.0276**	**0.0276**	0.0277	**0.0276**	0.560	0.5144
	J2	0.1091	0.0658	0.4490	0.0409	**0.0410**	0.4090	0.073	0.0729
	J3	**0.0361**	0.0444	**0.0361**	0.0362	**0.0361**	0.0394	0.262	0.0858
	J4 exponential	0.0393	0.0391	0.0392				0.074	0.0853
	J4 linear	1.0818	0.4363	0.0611	**0.0371**	**0.0371**	**0.0371**		

6.4 Concluding remarks

Following a comprehensive series of open-loop tests, an SDP model of each joint of the right hand manipulator was identified and used to develop several different model-based controllers. The controllers range from standard PI control to a novel PIP IDZ controller, where the IDZ control element is used to deal with the dead-zone in the hydraulic actuator system. The IDZ approach facilitates use of standard linear controllers for the dynamic response. These controllers were developed and controller parameters tuned based on the SDP model output.

The various controllers were compared, first using a step command input then with a sine wave trajectory. The results show that whilst the newly developed controllers all improve upon simple PI control, no single controller performs the “best” for all the metrics used for comparison. However, when considering the specific requirements of the application investigated here, namely improving control performance for the assisted teleoperation interface developed in Chapter 5, and the control objectives set out in the chapter introduction, the forward path form of the PIP IDZ controller emerges as the most suitable candidate design for further development. For step input tests, this gave the smallest overshoots, achieved steady state for each joint with either a small or no steady state error and had the least amount of CFP voltage switching. For the sine wave inputs it yields the lowest MSE for joint angle set point following, and again the lowest CFP voltage switching.

In more generic terms, the key novelty and contribution of the chapter was the use of SDP system identification methods to quantitatively identify the IDZ control element, the development of various model-based control systems using this control element, and the simulation based investigations into the most suitable implementation form. The following Chapter 7 demonstrates application of the selected controller to the real robotic system in the laboratory.

Chapter 7

Testing Controllers on hardware

Alongside the model-based controllers developed in Chapter 6, a very different type of controller to support the assisted teleoperation system is considered and tested on the Brokk- Hydrolek platform, namely a fuzzy logic based controller. The general fuzzy logic approach and the two controllers developed in this thesis, namely single-input and two-input controllers, are all discussed in Section 7.1.

The SDP based controllers developed in Chapter 6, as well as the new fuzzy controller, are subsequently tested on the real system and their performance compared. Initial tests using the same step and sine wave sequences as used for testing with the SDP model, are presented in Section 7.2. This allows for a comparison between the simulated results and the experimental results, which gives an indication as to how good the new SDP joint models are when operating in closed-loop mode. From these results, the best performing controller was identified and tested with the assisted teleoperation system developed in Chapter 5, with the results discussed in section Section 7.3.

7.1 Fuzzy logic approach

Fuzzy controllers are non linear and do not require a mathematical model of the system, instead relying on a set of linguistic rules set by the developer based on their experience. In this way they are unlike the model based controllers discussed earlier, this has both potential advantages and disadvantages, but it makes the fuzzy approach an interesting one to investigate for the control of the hydraulic manipulators. The advantage of this approach is that it creates a non linear controller so can potentially provide better control of the non linear hydraulic actuator, also not needing a model means it can be quick to develop and implement. The downside of this approach is that because it does not use a model it relies on experience and an amount of guesswork and trial and error to develop the controller, so an optimal controller may not be developed.

Fuzzy logic can be incorporated into a control system in a number of ways, such as directly as is done here, or another common method is using fuzzy logic to tune PID gains in real-time, so that a fuzzy PID system can be used to control a non linear system. However it is used the basic principles are the same as discussed in the following sections.

In the same way as other control systems, a fuzzy controller has 3 main stages, an input, a processing stage and an output. The processing stage consists of four parts, as shown in Figure 7.1 [158]. First the input, or inputs, go through a fuzzification process which modifies the inputs into a form compatible with the rules stored in the rule-base. Next the inference mechanism applies the relevant rules to the input. The rules are what determine the behaviour of the controller. The output of the inference mechanism goes through defuzzification to convert it back to a signal that can be used as the input to the plant. Figure 7.1 shows an overview of a single input fuzzy logic controller, which looks very similar to any other feedback controller diagram, but with some added detail inside the controller block.

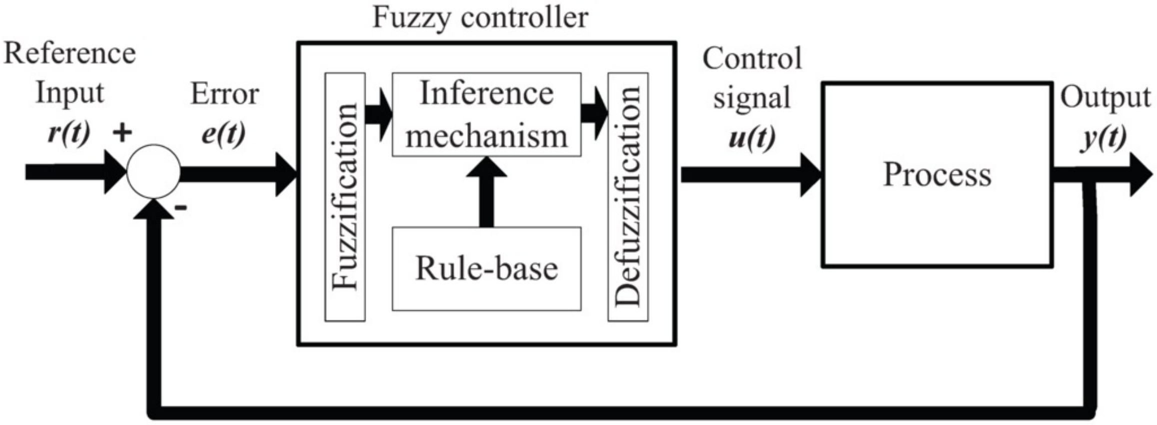


Figure 7.1: *Block diagram showing structure of a fuzzy controller.*

Within the inference mechanism all the inputs to the system are mapped to membership functions generating a value for each rule that applies to the inputs. Membership functions match the input to one or more fuzzy sets, giving a degree of membership between 0 and 1. For example if room temperature was an input, the temperature at which the room becomes hot is subjective and varies between person, in traditional logic a set value would need to be decided where above that value is hot and below is not hot. In fuzzy logic the room could be ‘quite hot’, so the room could be hot to a degree of 0.7 for example. Linguistic terms such as very, quite and slightly are used to define membership functions. The membership functions are often triangular, as shown in the example in Figure 7.2, but bell curves and trapezoids are also common shapes.

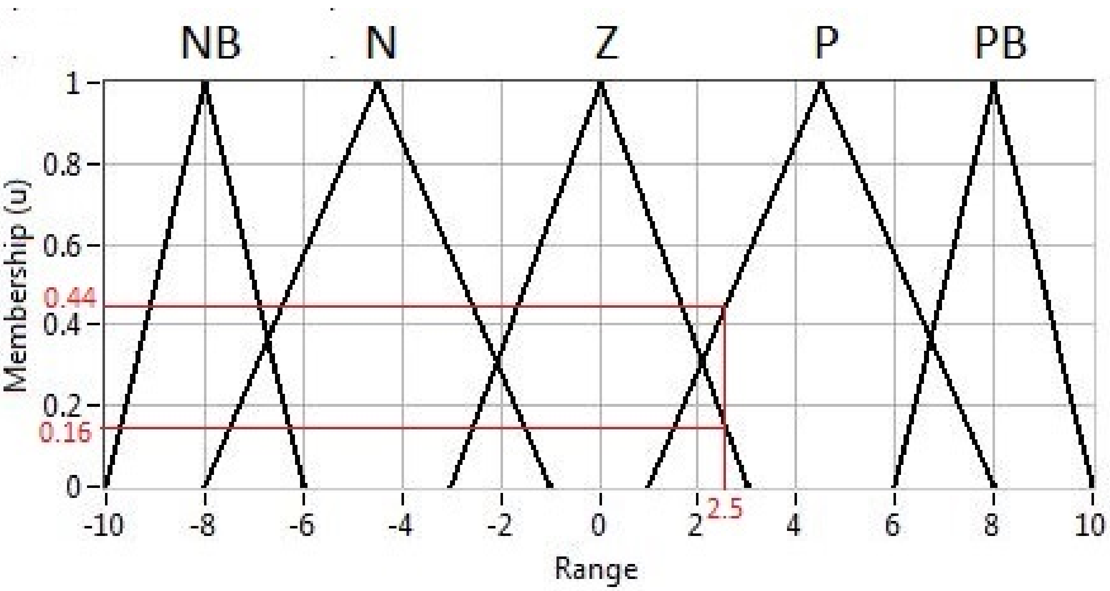


Figure 7.2: *Diagram showing example of 5 membership functions, from left to right, Negative Big, Negative, Zero, Positive and Positive Big. An example of a single input being in two fuzzy sets is shown.*

The inference mechanism then uses a series of linguistic rules to generate the output values. The rules are typically of the form IF...THEN..., for example IF room temperature is 'hot' THEN fan equals 'high'. All rules that apply are processed potentially giving multiple outputs of the inference mechanism. The rules being in this simple linguistic format makes it easy for people to develop the controller rule base, as rules can be based on experience and are simple to understand.

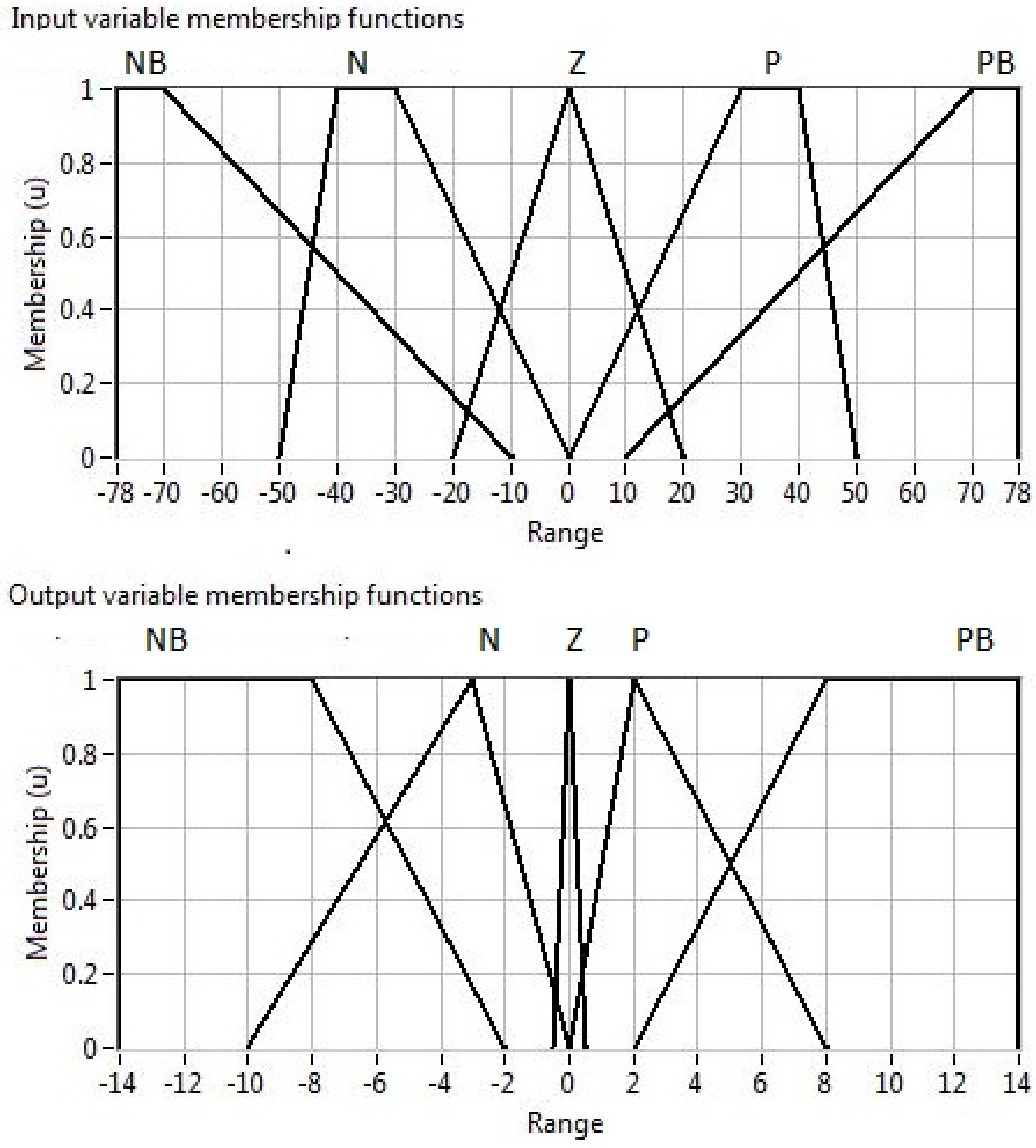
As an example, in Figure 7.2, it is shown how an input of 2.5 belongs to both the Zero and Positive membership functions, due to the membership functions overlapping. This would trigger at least two rules, depending on the size of the rule base and complexity of the designed controller, and generate multiple outputs to go into the defuzzification stage.

7.1.1 Single input fuzzy controller

For the initial investigation to test the viability of a fuzzy controller a simple single input single output system is investigated, with the error in joint position being the input and voltage to the CFP the output. This is the simplest possible fuzzy controller. Five membership functions were used with angle error being categorised as ‘Negative Big’, ‘Negative’, ‘Zero’, ‘Positive’ and ‘Positive Big’. Due to being a simple single input system the rule set is small, so the whole system is very simple.

The fuzzy logic controller is developed and implemented directly in NI Labview and as such there are no model or simulation results, only experimental results. A trial and error based approach is used to create and modify the membership functions. For the initial development and tuning of the membership functions, a text file containing set point and measured joint angle data was provided and the controller output observed to ensure it was in the correct range and behaved in the manner expected. This reduced the likelihood of damaging the manipulators by using an untested controller on them.

Figure 7.3 shows an example of the input and output membership functions for the single input controller, as can be seen these are a mix of trapezoids and triangles. The selection of membership function shape and position was decided through trial and error, using knowledge gained from using the system over time, making tweaks until satisfactory performance was achieved. Figure 7.4 shows the control surface of the single input controller corresponding to the membership functions in Figure 7.3, highlighting the non-linear relationship between the input and output.


Input variable membership functions

Output variable membership functions

Figure 7.3: *(Top) Input membership functions and (bottom) Output membership functions, for J2 of right hand manipulator.*

Figure 7.5 shows the set point following performance of the single input controller, following a series of steps of different amplitudes that cover the range of motion of

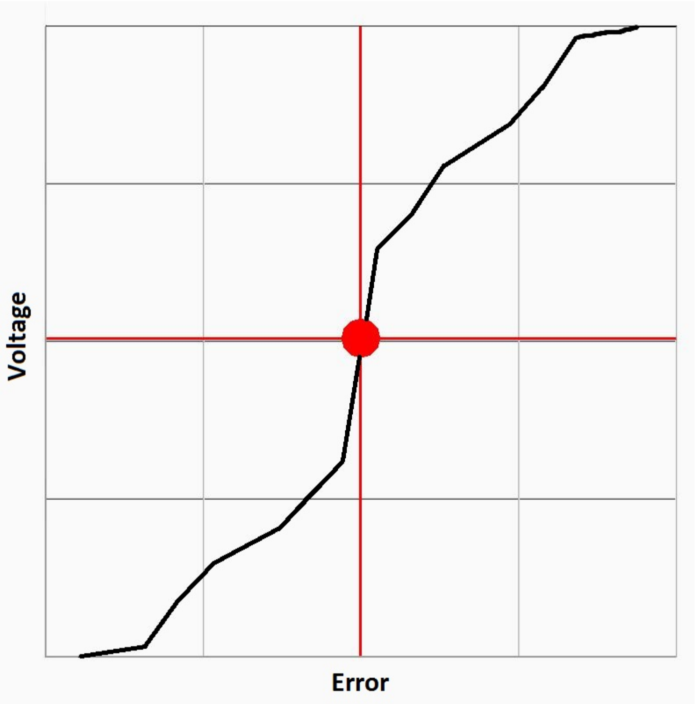


Figure 7.4: *Control surface of single input fuzzy controller, showing non linear relationship between the input, position error, and output, voltage.*

the joint. Set point following performance is good across the range of the joint, with different sized steps all being followed successfully, but with not reaching the set point at the upper limit. There is some small overshoot, particularly when moving to increase joint angle in a positive direction, which is not as noticeable when moving in a negative direction, this is likely caused by the asymmetry of the actuator.

7.1.2 Two input fuzzy controller

Once the single input fuzzy controller had shown that fuzzy logic was a viable approach to the control problem, a more complex 2 input controller was investigated, using position error and change of error as the inputs. A 2 input fuzzy controller is more

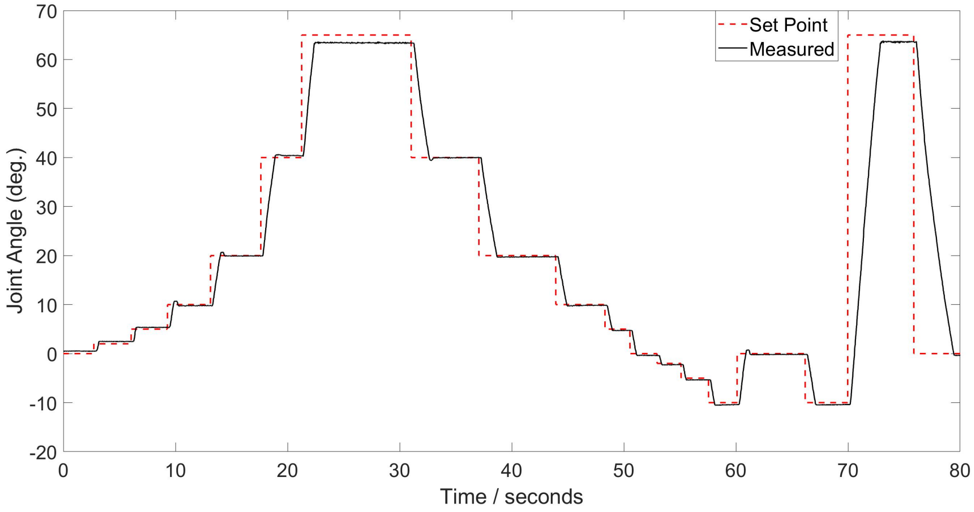


Figure 7.5: *Single input fuzzy controller for Joint 2 set point following a series of step inputs.*

conventional, with change of error or rate of change of error the most common second inputs. Change of error was used here, and is simply found by subtracting the current error from the error at the previous sample. This is potentially useful to include so that control action can depend on both position error and joint velocity, for example if position error is small and change of error is small, a different output may be beneficial than if the position error is small and change of error is large. In the first case the output is close to the setpoint and moving slowly, in the second case the output is again close to the setpoint but moving fast, so is likely to overshoot if the controller can not deal with this.

Adding a second input increases the number of rules from 5 to 25, increasing complexity but potentially improving performance. Table 7.1 shows the rules used for the new fuzzy controller, as before NB represents Negative Big, N Negative, Z Zero, P Positive and PB Positive Big. The input and output membership functions for each joint are shown in Appendix F for brevity. Figure 7.6 shows the control surface of the 2 input controller, comparing this with the flat ramp you would get with a PID controller, and the benefit of a non linear fuzzy controller is clear for the non linear hydraulic system.

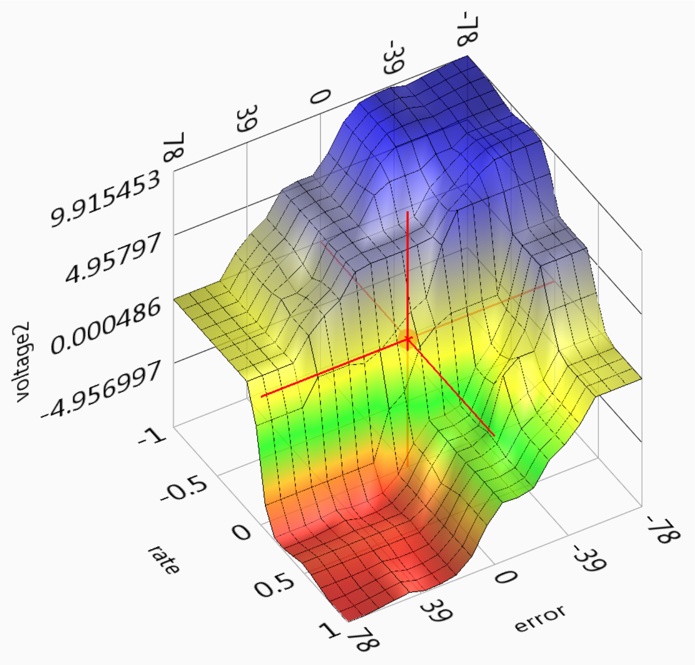


Figure 7.6: *Control surface for initial 2 input fuzzy controller.*

7.1.3 Fuzzy controller results

The two input fuzzy controller was tested with the same step and sine tests as the other controllers, to give a fair comparison, example outputs are shown in Figures 7.7 and 7.8 for step and sine inputs respectively. As can be seen set point following performance is generally good, with the exception of the joint 1 step input, usually with just a small overshoot in the step test and very close following of the sine input. The results

Table 7.1: *Initial rule table for 2 input fuzzy controller.*

	Change of error				
Error	NB	N	Z	P	PB
NB	PB	PB	P	P	P
N	PB	P	P	P	P
Z	Z	Z	Z	Z	Z
P	N	N	N	NB	NB
PB	N	N	N	NB	NB

are discussed and compared with the model based controllers in Section 7.2. There is however room for improvement, particularly with joint 1 where there is sizeable oscillation at the top of the step movement, and small oscillation along the sine input. This improvement could be achieved in future development by modifying the rule table or tweaking the membership functions. However these results do prove the viability of using a fuzzy logic controller on the hydraulic manipulator, showing good control is achievable.

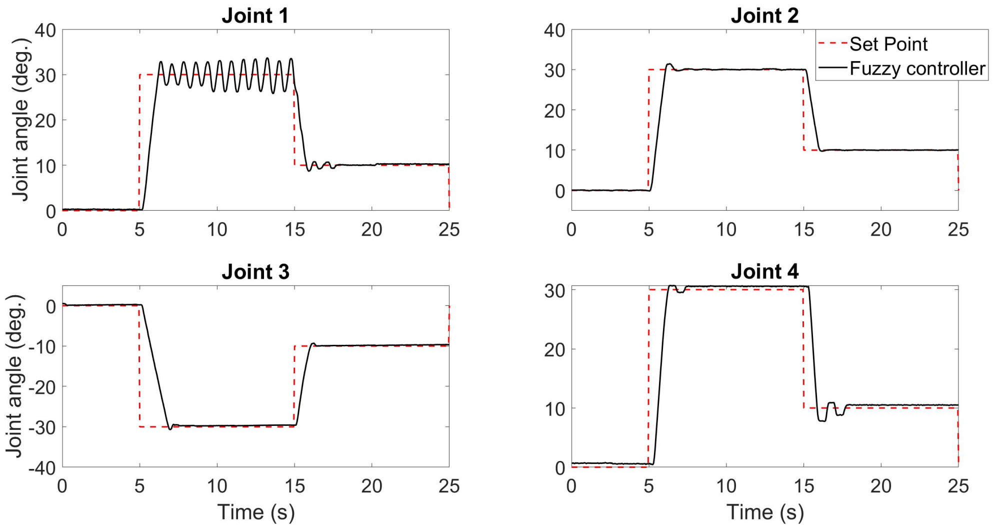


Figure 7.7: *Two input fuzzy controller set point following, for a step input, showing average of 3 tests.*

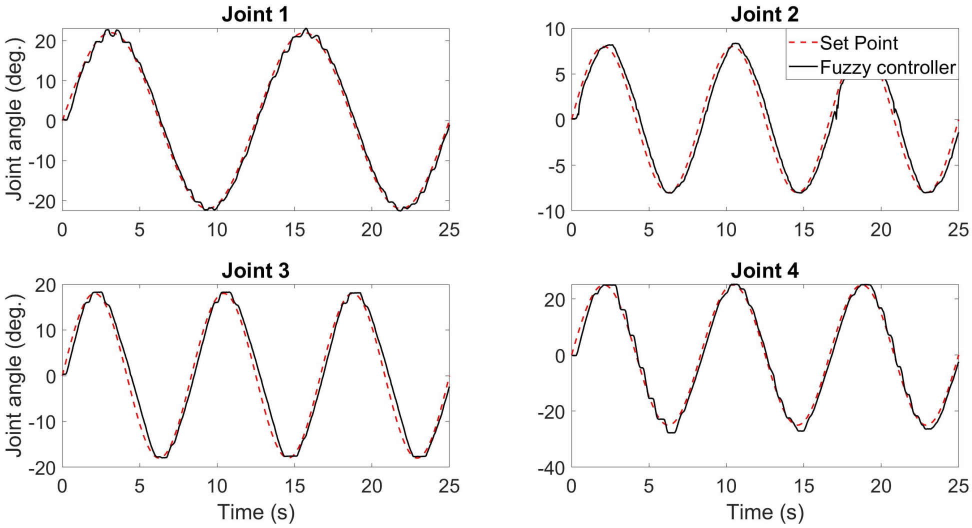


Figure 7.8: *Two input fuzzy controller set point following, following a sine wave input, showing average of 3 tests.*

7.2 Comparing controller performance

The controllers were tested on the Brokk-Hydrolek platform using the same step and sine inputs as used for the model based results and the performance of the different controllers is compared here. For all the experimental results presented here tests were repeated at least three times to confirm repeatability, and the average values of the 3 iterations are used for the comparison. For each test the system was operated beforehand to ensure fully working, and to warm up the system so that controllers were not being tested on a cold system, as this may impact on performance.

Before comparing the controllers it should be made clear that the fuzzy logic controller was developed directly using the real system, and not based on a model like the PIP controllers were. As such this will have some impact on the results, as the model based controllers can only be as good as the model and any errors or inaccuracy in the model will weaken the performance of the controller when tested on the real system. However approaches such as the forward path controller are not possible without a model. Also as the purpose of investigating the fuzzy approach was to determine whether it is a viable alternative to established model based controllers, including one of the main benefits of the approach namely not requiring a model is justified.

7.2.1 Step tests

Figure 7.9 shows a plot comparing the experimental step response of the different model based PIP controllers, showing the ‘c’ tuning of the controllers using the IDZ method. The plot corresponds to the data in Table 7.3. From visual inspection of the plots it can be seen that for most of the joints the response of the different controllers is very similar, with the scaled input approach being slightly worse overall than the IDZ controller, although noticeably not for joint 4 which has proven to be problematic. From Figure 7.9, the forward path PIP controller gives the best performance overall, although there is still room for improvement, for example there is still sizeable overshoot of joint 1. The benefit of the forward path controller is most clearly shown in the response of joint 4, where the feedback IDZ controller has constant oscillation which is removed by using the forward path approach, for the other three joints the benefit of the forward path approach is less noticeable from the plots.

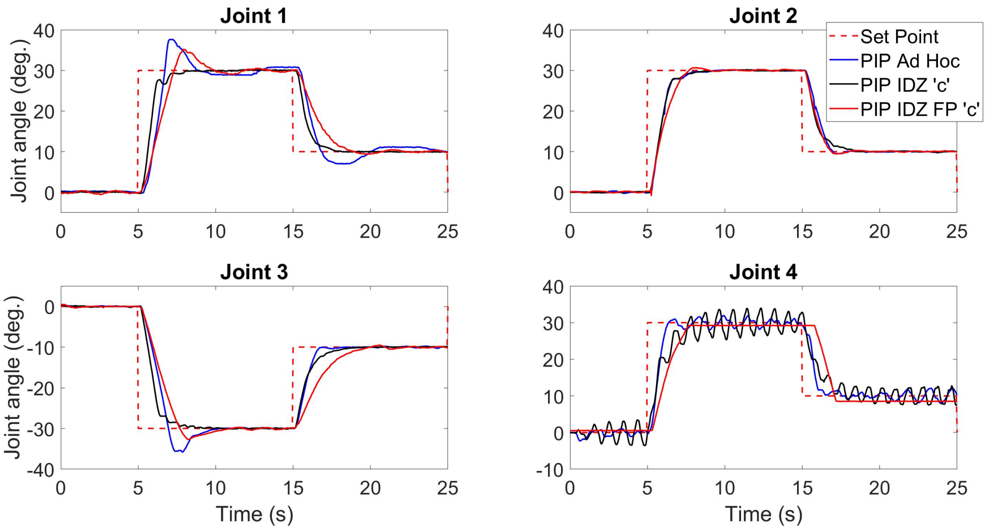


Figure 7.9: *Comparison of PIP controllers for step input set point following, experimental results.*

Looking at Table 7.3 for the experimental results shows a similar story to the model results in Section 6.3 with the PIP IDZ feedback controller having the majority of the lowest values for rise time, which can be seen in Figure 7.9. However the experimental results for the forward path controller are not as good as the SDP model results, with the feedback form of the controller having the majority of lower values for percentage overshoot, and the PIP controller with scaled input actually having the lowest measure of CFP voltage switching for joints 2 and 3. Unlike the SDP model results the experimental results don't show one controller as being clearly better than another, with the best performance for each metric being fairly spread across different controllers and the fuzzy controller providing similar levels of performance to the model based controllers. However they do show that the forward path approach was able to handle the problematic joint 4, and so that approach has promise but may need further tweaking or tuning of the gains using the output of the real system rather than the SDP model.

Table 7.2 shows a typical example of the subtracting the experimental result values from the SDP model results, showing the difference between the model and experimental data. It shows that in most cases the model response is better than the experimental result, in that the modelled overshoot is typically smaller as is the CFP voltage switching (indicated by a negative value in the table). These results show that the rise time and CFP voltage switching are very similar between the SDP model and real system, but that typically the real system has a noticabely larger overshoot showing there is room for improvement in the SDP model. These differences between the SDP model and the real system likely explain the overshoot in the measured step test results for the forward path controller, and suggest that improving the SDP model may improve the forward path controller performance.

Table 7.2: *Example of Values of Model output minus Experimental data.*

		PIP IDZ FP		
		a	**b**	**c**
Rise time	**J1**	0.1953	0.0377	0.6694
	J2	-0.1118	0.0167	-0.1127
	J3	-0.0015	-0.0275	0.2083
	J4	0.0354	N/A	0.0995
Overshoot %	**J1**	-14.1808	-7.3987	-17.36
	J2	4.84	1.7177	-2.2067
	J3	-1.7317	2.4513	-8.9978
	J4	2.8467	-29.4823	1.333
Var of Diff of CFP	**J1**	-0.0625	-0.2018	-0.0481
	J2	-0.0649	-0.1246	-0.1414
	J3	-0.0804	-0.2832	-0.0614
	J4	0.0103	-0.123	0.0132

Table 7.3: *Experimental results for PIP controller response to step inputs, comparing rise time, overshoot and a measure of switching of the CFP voltage. The best values are in bold and N/A means data was oscillating too much to calculate value.*

		PIP IDZ			**PIP IDZ FP**			**PIP scaled**	**PI scaled**	**Fuzzy**
		a	**b**	**c**	**a**	**b**	**c**	**input**	**input**	
Rise time	**J1**	0.9291	0.7681	0.8212	0.8657	**0.6296**	1.6139	0.9876	1.0054	0.7808
	J2	0.7850	**0.7338**	1.1889	1.0173	0.7915	1.7033	1.1408	0.8397	0.7400
	J3	1.1155	**1.0210**	1.1439	1.2061	1.0048	1.9318	1.3857	1.2456	1.3704
	J4 exponential	N/A	N/A	N/A				0.7896	0.9924	**0.6870**
	J4 linear	N/A	**1.0809**	2.0501	1.4026	N/A	1.9380			
Overshoot %	**J1**	19.7611	27.3311	**0.6356**	28.7078	22.4489	17.3600	25.4067	32.2144	11.3722
	J2	28.1122	44.2044	0.5689	6.8267	12.6156	2.2067	**0.3356**	16.6500	4.8967
	J3	37.2033	38.9011	**0.4711**	13.8122	11.0167	8.9978	19.3167	39.5878	2.2600
	J4 exponential	29.5056	30.9311	13.2311				6.4144	20.1467	2.3456
	J4 linear	0.1011	2.6433	7.5322	0.8200	41.8156	**0.0000**			
Var of Diff of CFP	**J1**	0.0747	0.3133	0.0944	0.0849	0.2629	**0.0616**	0.0301	0.1799	0.1770
	J2	0.1934	0.4041	0.0705	0.1156	0.1722	0.1595	**0.0540**	0.2325	0.3880
	J3	0.1642	0.6214	0.0757	0.1038	0.3629	0.0750	**0.0527**	0.1093	0.0682
	J4 exponential	1.1106	2.3445	0.2228				0.1703	0.0766	0.1220
	J4 linear	0.3775	0.1923	0.2019	0.0089	0.1467	**0.0104**			

7.2.2 Sine tests

Figure 7.10 shows a comparison of the different PIP controllers for each joint, now using the ‘b’ tuning for the IDZ controllers. From these plots it is clear that the forward path controller gives the best response to a trajectory input for the model based controllers; closely following the sine input with a very small delay, unlike the scaled input approach which has a noticeable lag in response, and with no oscillation unlike the feedback form of the IDZ controller which has noticable oscillation for all the joints.

Unlike with the step tests, when looking at the experimental sine wave results, Table 7.4, the forward path controller is clearly the best of the model based controllers too, although it less clear than with the SDP model results. For the experimental results the best value of each metric is more spread across different tunings of the forward path controller whereas for the model response tuning ‘b’ contains most of the best values.

Unlike with the step tests, here the fuzzy controller shows noticeable improvement over the other controllers, with the lowest values of MSE for joint set point following, showing that it most closely follows the target. This can also be seen in Figure 7.8, where the set point is followed with almost no delay and only slight oscillation. Although the fuzzy controller doesn’t have the lowest values for the measure of CFP voltage switching, the results are still very low and the improvement in trajectory following is a bigger priority, so overall the fuzzy controller still gives the best results.

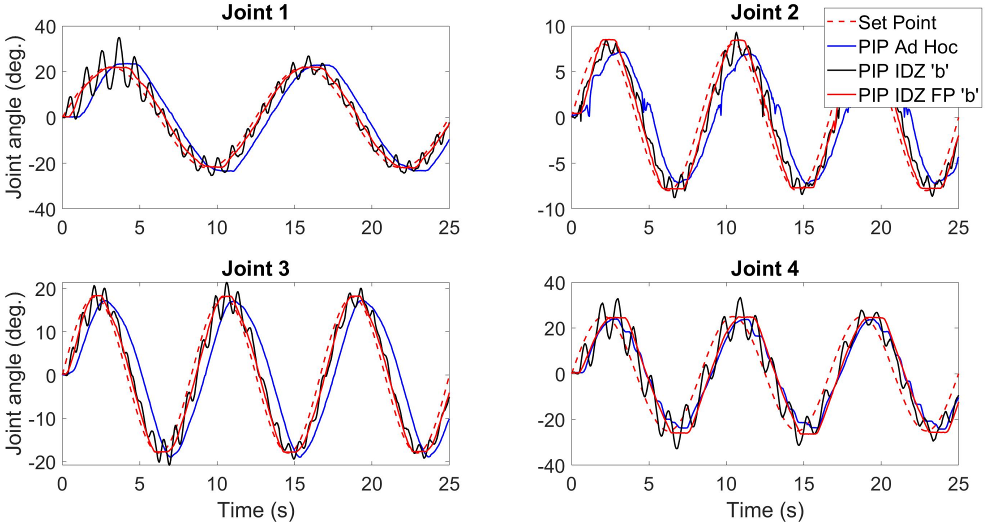


Figure 7.10: *Comparison of PIP controllers for sine input set point following, experimental results.*

Table 7.4: *Experimental results for controller response to sine inputs, showing euclidean norm error for end effector, MSE between set point and measured angle for each joint and a measure of CFP voltage switching. The best values are in bold.*

		PIP IDZ			**PIP IDZ FP**			**PIP scaled input**	**PI scaled input**	**Fuzzy Logic**
		a	**b**	**c**	**a**	**b**	**c**			
Avg. Euclidian Norm										
	J4 exponential	164.79	131.7265	170.8485				189.9105	159.9839	**40.0108**
	J4 linear	165.051	132.936	170.5319	170.8278	**77.2116**	276.4220			
MSE set point following	**J1**	34.3284	17.8304	38.5135	40.2781	5.6797	147.0428	56.8033	20.5543	**1.2944**
	J2	4.5741	3.4513	1.1633	3.9828	1.934	10.5186	12.7770	8.0446	**0.9787**
	J3	29.4723	12.0346	58.2137	44.4468	5.75	157.5334	64.0240	81.9050	**4.1550**
	J4 exponential	58.5378	59.0855	126.4875				77.0698	144.7568	**4.2416**
	J4 linear	68.634	**56.7307**	122.816	311.1693	85.443	287.5734			
Var of Diff of CFP	**J1**	0.0277	0.384	0.0555	0.0148	0.0213	**0.0131**	0.0263	0.0301	0.0996
	J2	0.255	0.2709	0.1187	0.0255	0.0263	**0.0249**	0.1110	0.0540	0.0933
	J3	0.0747	0.4914	0.0534	0.202	0.2660	**0.0308**	0.0448	0.0527	0.0833
	J4 exponential	0.7721	1.0368	0.3305				0.1922	0.1703	0.1583
	J4 linear	0.4347	0.3043	0.1113	0.0236	0.2250	**0.0188**			

7.3 Testing with vision system

After developing and testing the controllers in controlled step and sine tests they were tested on the robotic platform, getting the set point input from the vision system developed in Chapter 5. As the forward path form of the PIP IDZ controller was shown to perform best out of the model based controllers on the step and sine inputs, and the fuzzy controller showed to be equal to or better than the model based controllers, only the results for these two methods are discussed here.

The user interface discussed in Chapter 5 was modified to use the new controller, by simply changing the Labview program. Following this tests were carried out placing a vertical plastic pipe in front of the platform, and then using the interface to generate target positions for the grasp, (a cut position was also identified but as the controller isn't yet implemented on the left hand manipulator this wasn't tested). Due to there being no collision avoidance algorithm in place, the manipulator was started at a set starting position away from the pipe and the other manipulator to reduce the likelihood of collisions. The coordinates of the grasp position were recorded so the same test could be repeated, as this wouldn't be possible by using the interface as it would require the pipe and the user click to be in exactly the same position.

7.3.1 PIP IDZ forward path controller

Figure 7.11 shows the end effector path during an illustrative grasp movement, showing the end effector moving to the initial position in front of the target then moving forward to the final target. Note that the path shows the wrist position, so the offset between the final position and the pipe is to allow for the gripper. From Figure 7.11 it can be seen that the end effector follows a smooth path with only some slight oscillation around each of the target positions, this is a noticable improvement over the original trajectory shown in Figure 5.9.

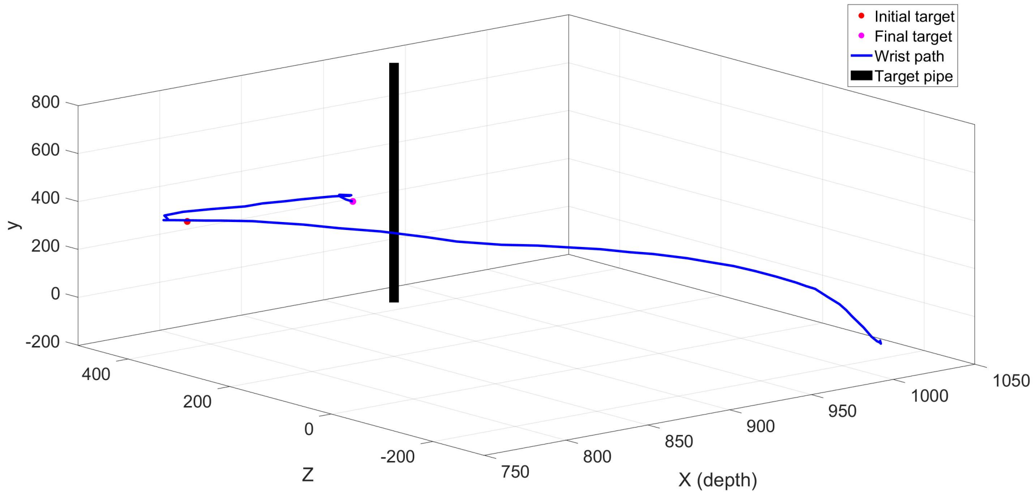


Figure 7.11: *End effector path during an illustrative pipe grasp, using the PIP IDZ Forward path controller.*

Figure 7.12 show the joint angles during the illustrative grasp motion in Figure 7.11, using the ‘c’ tuning of the PIP IDZ FP controller. Figure 7.13, shows the same but using the ‘b’ tuning of the controller, to illustrate the differences between different tuning of the controller parameters. Each joint starts at either 20 or -20 degrees, so as to position the arm in a safe starting position as discussed, and the initial step occurs at time $t = 0$, the second step occurs when each of the joints have reached a set error threshold to the target. The results show that with the ‘c’ tuning the response is slower than with the ‘b’ tuning, as expected. However what is interesting is that with the ‘c’ tuning the target set point is not reached and the joint settles with a small offset. This is not present in the SDP model results and shows an area that needs further investigation as it could be due to some model mismatch. Tuning ‘b’, optimised for trajectory following, does a good job of reaching the set point, but due to the rapid movement there is overshoot of the target before settling at the set point. Table 7.5 shows that the overshoot is quite significant and higher than desirable, so from the current results tuning ‘c’ would be more desirable, even though it is less accurate.

Figures 7.11 and 7.12 show that the PIP IDZ controller provides good control for the real grasp motion, but further improvement could be achieved, possibly by tuning the controller parameters using the experimental data rather than the SDP model output. The secondary aim of improving the controller was to produce a clean voltage signal to the CFP to reduce oscillation and therefore wear on the valves. Figure 7.14 shows an example of the CFP voltage signal for each joint during the grasp motion, it shows the signal has little oscillation, importantly no excessive switching from positive to negative voltage and so the controller fulfills the secondary aim as well.

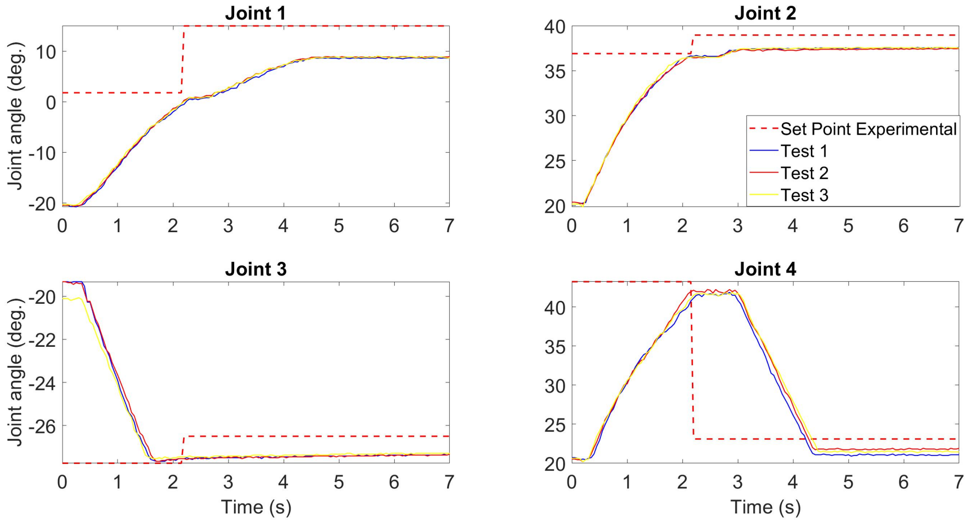


Figure 7.12: *Joint angles during grasp motion, controller tuning 'c', initial step at time 0, average of 3 iterations.*

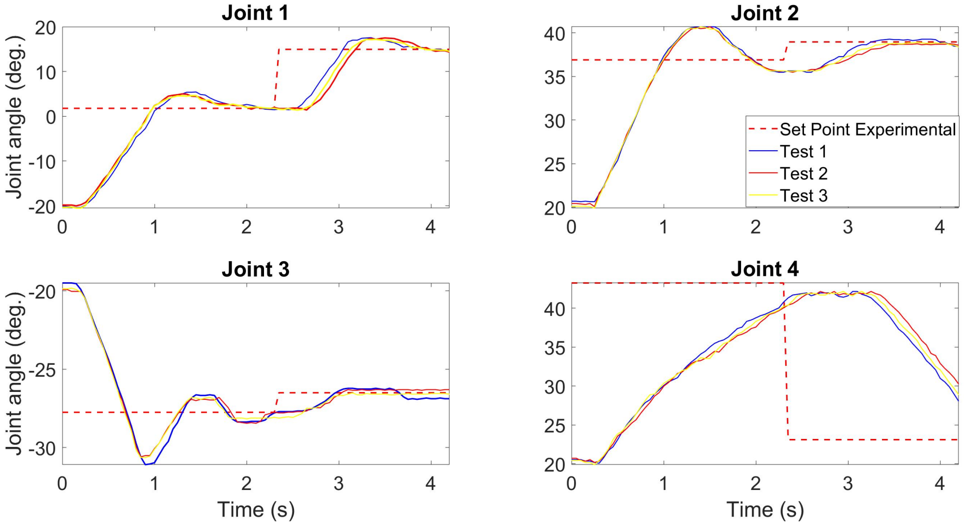


Figure 7.13: *Joint angles during grasp motion, controller tuning 'b', initial step at time 0, average of 3 iterations.*

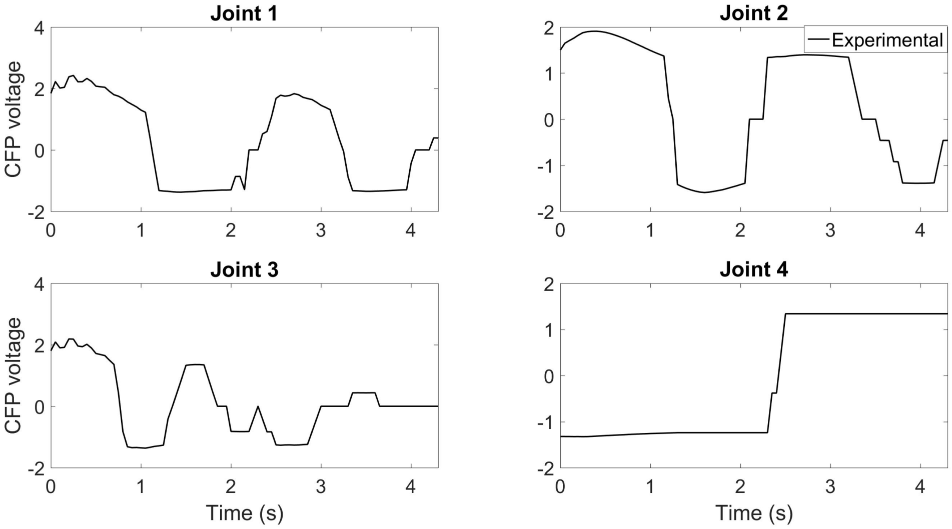


Figure 7.14: *Example of voltage sent to CFP during grasp motion, average of 3 iterations.*

Table 7.5: *PIP IDZ FP controller comparison with fuzzy controller.*

		PIP IDZ FP			**Fuzzy**
		a	**b**	**c**	
Rise time	**J1**	0.8251	0.5994	1.5805	0.6634
	J2	0.7634	0.5962	1.4384	0.5796
	J3	0.5345	0.3801	0.9677	0.3954
	J4	1.9217	1.9479	1.6803	0.5252
Overshoot %	**J1**	16.8979	14.9089	0.0000	0.0000
	J2	11.4615	22.4852	3.1578	0.0000
	J3	30.8559	39.0151	0.0000	10.2667
	J4	0.0000	0.0000	0.0000	0.1406
Var of Diff of CFP	**J1**	0.0402	0.1923	0.0262	0.0869
	J2	0.0263	0.1493	0.0559	0.2370
	J3	0.0226	0.1813	0.0216	0.3121
	J4	0.0147	0.0745	0.0457	0.1458

7.3.2 Fuzzy controller

From Table 7.5 it can be seen that the fuzzy controller generally gave similar rise times to the fast 'b' tuning of the PIP IDZ FP controller, with the exception of joint 4 in which the fuzzy controller has a much smaller rise time. Where the fuzzy controller really stands out is with the mostly small or zero overshoots whilst having fast rise times. As a result of the lack of overshoot and the fast rise times the fuzzy controller completes the grasp motion very quickly, this can be seen in Figure 7.15 which shows the joint angles during the grasp motion. The area the fuzzy controller is less good is in the CFP voltage switching, Table 7.5 shows that it has the highest voltage switching value for each of the joints, although the value is still low so there is not a large amount of switching.

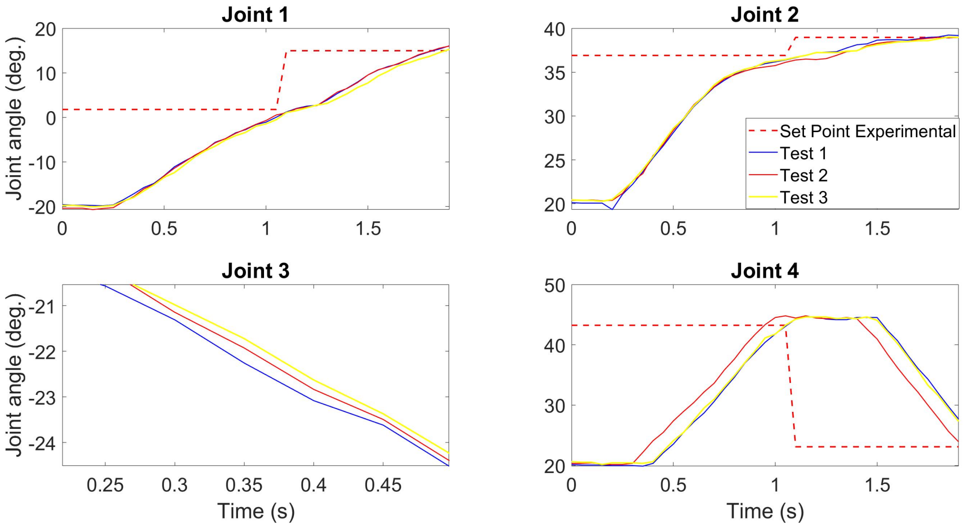


Figure 7.15: *Joint angles during grasp motion, with Fuzzy controller initial step at time 0, average of 3 iterations.*

7.4 Concluding remarks

A fuzzy logic approach to the control problem was investigated as a potential alternative to the model-based methods of Chapter 6. The performance of the new fuzzy logic controller, and the model-based controllers developed in Chapter 6, were evaluated using the Brokk-Hydrolek hardware platform. Initial comparisons used the same step and sine inputs as used for the SDP simulation study in Chapter 6, allowing for comparison between the SDP model output and the real hardware using the same controllers. Further experiments tested these controllers within the context of the assisted teleoperation system developed in Chapter 5. The results showed that both the model-based and fuzzy logic control approaches have benefits, but that overall for use with the assisted teleoperation interface, the fuzzy controller appears to yield slightly improved performance. This is particularly the case in respect to reduced rise time and overshoots during testing with the assisted teleoperation system, in which the fuzzy logic approach yields faster and more accurate movement (in comparison to the specific SDP-based designs used here).

However the fuzzy controller was developed directly on the real system, and not using a SDP model, so some of the improvement may come from inaccuracy in the model, to investigate this further and draw stronger conclusions further work could be done either developing PIP controllers from the real data not the SDP model, or tuning the fuzzy controller using the same SDP model. This would allow a more direct comparison between the two approaches. However these results do successfully show that a fuzzy logic controller is a very viable alternative to traditional model based controllers for this application.

Another thing worth pointing out based on these results is that the reason for the noticeably poorer performance of joint 4 may be due to it being a completely different actuator to the other three joints, joint 4 is a vane type rotary hydraulic actuator, whereas the others are linear hydraulic actuators. Because of this it exhibits different physical behaviours, and the SDP model approach may need adjustment to improve results. For example assumptions such as no oil leakage may not be valid, and a more complex model may be required. Although joint 4 also showed some more minor issues with the fuzzy approach, for example when following the sine trajectory the response is very jerky, see Figure 7.8. As the fuzzy approach is not model based, this may indicate some hardware fault with the particular actuator.

Chapter 8

Conclusions

This thesis has made research contributions in three main areas, all in the context of nuclear decommissioning and with laboratory evaluation using a ground-based, hydraulically actuated, dual-manipulator robot. The three areas are: the development of a detailed mechanistic model of the manipulator platform, the development of a new assisted-teleoperation user interface, and the design and testing of nonlinear and fuzzy logic control systems to improve the controlled performance of the hydraulic manipulators.

Alluding to the formal research objectives set out in Section 1.2, the first two objectives concerned the mechanistic modelling equations and the use of evolutionary algorithms to estimate the unknown parameters of the model. The results were discussed in Chapter 4. The model, which contains all the mechanical, hydraulic and electrical elements of the system, as well as providing a visual representation of the manipulators, builds on previous work at Lancaster University, but with significant modifications and improvements introduced in this thesis. The hydraulic system model component was recreated from the ground up to more accurately reflect the real hardware. Once the model was finalised, a genetic algorithm was developed and tuned to this particular problem, and subsequently used to identify the unknown parameters of the system i.e. to determine realistic values that matched the model performance to that of the hardware. The developed model can be used for future research into advanced controllers that are beyond the scope of the present thesis (e.g. research on algorithms that combine inverse kinematics solvers for trajectory tracking with control system design). Another future use for the mechanistic model is to investigate the impact certain parameters can have, for example, what effect a drop in oil pressure or an increase in friction on a particular actuator or joint would have on the overall system performance. In the latter regard, the new model can be used for further research into condition monitoring. Testing against laboratory data showed that overall the model yields a good indication of the real manipulator behaviour. The model performance is best for the joints closest to the base, but for the further out joints, the model needs more work to improve its accuracy. Nonetheless, one of the major benefits of the model is that it yields a graphical representation of both manipulators hence their movements can be visualised in simulation before being tested on the real hardware. This allows new task scenarios (e.g. pipe cutting) to be investigated off-line, and so allows the operator to set up the task in a way that reduces the likelihood of the arms colliding with each other or items in the workspace.

The third research objective was to develop an assisted-teleoperation interface to replace the current industry standard joystick control. This was described in Chapter 5. The original method of using joysticks to control the manipulators is highly skilled and very slow, hence speeding up this process and reducing the length of training needed would help reduce decommissioning costs and time-scales. The developed interface requires just four mouse clicks to perform a pipe grasp and cut movement, replacing the old system of moving each joint independently with joysticks. This approach allows all joints to be moved simultaneously, which speeds up the action considerably. Initial testing, with experienced and inexperienced operators, showed that the new interface is much faster and easier to use than the old highly skilled method. The assisted- teleoperation system works by using a MS Kinect RGB-D camera to gather depth data about the workspace, then transforming the user selected points into 3D coordinates that are used as inputs to an inverse kinematics algorithm, which calculates the joint angles needed to reach the target position. The user selects the target points from the image on screen, which shows both a live colour video of the workspace and an edge detected image of reachable objects. It is on the edge image that the operator selects target objects and positions. This semi-autonomous system reduces operator workload and is shown to work effectively, with one manipulator successfully grasping a target pipe and the other moving across to simulate a cutting operation. It should be pointed out that the cutting operation itself is not investigated in this thesis, in part due to health and safety issues for use of the machine in its current laboratory setting. Nonetheless, the output of this part of the project is a novel user interface and assisted-teleoperation system for e.g. pipe-cutting. This reduces operator workload and task completion time, whilst keeping the operator in control of the system and avoids using a fully autonomous system which is undesirable in the nuclear industry.

The fourth research objective was to develop, implement and evaluate various feedback control algorithms. The development of linear and novel nonlinear model-based PIP based controllers was discussed in Chapter 6, followed by their implementation and testing on the robotic platform in Chapter 7. A series of controllers of differing levels of complexity were designed, based on the PIP control structure, and tested using both a step input (which is similar to the inputs from the assisted-teleoperation interface) and a sine wave input i.e. to investigate trajectory following performance. To develop the nonlinear PIP controllers, an SDP model was identified from experimental data. In contrast to prior research also using the Brokk-Hydrolek platform and SDP models, this thesis investigated the angular velocity of each joint via the introduction of a state- dependent gain. This facilitated the use of an IDZ control design approach as an improved way of handling the dead-zone in the hydraulic actuators.

Testing of the controllers showed that joint 4 was particularly problematic, with excessive oscillation and relatively poor responses. The cause of this was not fully identified, but it is suspected that it is connected to joint 4 being the only rotary hydraulic actuator, with the others all being linear actuators. However, a forward path PIP controller approach, using the model output in the feedback of the control loop, yields improved results. This approach substantially reduced the excessive oscillations associated with the original design, reduced the overshoot and yields good set point tracking. The forward path approach also yields improved performance for the other joints, although much less noticeably. The PIP based controllers were tested on the physical system. The results of the model output and real measured results were compared, to evaluate the accuracy of the SDP model and to identify which controller gave the best performance. Once the best performing controller had been identified, it was directly tested with the new assisted tele-operation system, to see if it resulted in an improvement in performance. These results were good, with successful pipe grasps and smooth motion to reach the target position. The switching of the voltage sent to the hydraulic valves was also reduced. This means the valves were opening and closing less frequently, reducing wear on them and other components.

As well as the model based PIP controller development, a separate completely different approach was also taken. A fuzzy logic controller was developed in Chapter 7, which used the error between the set point and measured angle, as well as the rate of change of error, as inputs. The fuzzy controller is not model based and instead relies on a series of linguistic rules that are set up based on the developers experience. Testing showed that the tuned fuzzy controllers gave very good set point following performance, a little better than the current PIP designs, demonstrating that a fuzzy logic controller is a viable approach for control of the non-linear hydraulic manipulators. The disadvantage of the fuzzy logic approach is that it lacks the systematic and mathematical processes for updating the controllers over time and for new manipulators, and is only available for testing on the real hardware so cannot be used for modelling and simulation work. By contrast, the new PIP inverse dead zone controllers, which were shown to yield improved performance over the industry standard PI designs, are relatively straightforward to update in the future if the parameters change over time, or the hardware is modified. Finally, the new controllers were combined with the assisted-teleoperation system. Hence, the output of the research reported in Chapters 5 through to 7 is a complete system, one that allows the user to select targets to be accurately reached, with smooth movement between target positions.

In this manner, each of the four main research objectives described in Section 1.2 have been achieved, albeit with scope for further research.

8.1 Future work

There are a number of areas of the project that would benefit from continued future work. Recommendations for future work based on this project can be split into short and long term suggestions and also some suggestions for work that is more tangential but still connected.

Short term suggestions are mainly to improve the control system performance or to expand the implementation. For example, the control systems developed in Chapters 6 and 7 could only be developed for four of the joints due to hardware failure of joint 5. Ideally the controller would have been developed and tested for the full arm, but due to the realities of a practical based project this was not achievable so is an obvious extension for future work when the hardware limitations have been addressed. Fortunately, due to the systematic approach taken to the model-based controller development, extending these results to include joint 5 would be a relatively straightforward task. Extending the fuzzy controller would be a case of creating an extra set of membership functions and tweaking the settings based on experimental performance.

Short term suggestions for future work are:

- Develop the mechanistic model further, so all joints provide accurate reflections of the real system, currently joints 3 and 4 do not provide good results when tested with real grasp data. This would allow the model to be used for investigation to the system and for developing the latest control systems.
- Extend the controllers to include joint 5 following the hardware being repaired.
- Carry out a full user study of the assisted teleoperation system, using for example the NASA-TLX (NASA Task Load Index), to give a measurable comparison between the joystick teleoperation and the new system. This is common in HRI work and would validate the system.
- Further work to develop the assisted teleoperation system to improve it, for example expanding to other tasks such as pick and place of objects.
- Refine the PIP controller, for example tuning the controller using the experimental data rather than the SDP model output. The results of the fuzzy controller for trajectory following show that improved control is achievable.

Long term suggestions are more focussed on developing the system for testing in a real nuclear environment. Robotic systems used in the nuclear industry have to be proven to work due to the very strict safety legislation in the industry, so testing in a more realistic environment would be beneficial. With this in mind long term suggestions are:

- Investigate radiation hardening and protection for the system. Electronics and some materials can be damaged by radiation so knowing what effect radiation would have on the particular hardware would be very beneficial, as would having strategies to protect sensitive components.

- Investigate suitable cutting tools for attachment to the system. A study is needed to investigate what the most suitable cutting tool for the particular application, and how this can be integrated into the platform.

- Carry out practical field tests in a real world test scenario. This would prove the usefulness of a dual manipulator mobile platform and potentially a first step in moving from a research platform towards starting work on a commercial platform that could be used in the real world.

Other related work that could be used to extend the system include, for example, adding cooperative support robots, such as discussed in the PhD project of Tom Burrell [159], where it is proposed to use a UAV to gather image data that is fed into the system rather than just a fixed camera. Another option may be to add eye in hand cameras to expand the visual capabilities of the system. Although these would increase the complexity of the system, the improved capability may be worth this additional complexity.

Bibliography

[1] M. J. Bakari, K. M. Zied, and D. W. Seward, “Development of a multi-arm mobile robot for nuclear decommissioning tasks,” *International Journal of Advanced Robotic Systems*, vol. 4, no. 4, pp. 502–524, 2007.

[2] M. J. Bakari, *The development of a a multi-arm mobile robot system for nuclear decommissioning applications.* PhD thesis, Lancaster University, Engineering Department, 2008.

[3] M. J. Bakari, D. W. Seward, and C. J. Taylor, “The development of a prototype of a multi–arm robotic system for decontamination and decommissioning applications within the nuclear industry,” in *12th International Conference on Environmental Remediation and Radioactive Waste Management*, (Liverpool, UK), October 2009.

[4] C. J. Taylor and D. W. Seward, “Control of a dual–arm robotic manipulator,” *Nuclear Engineering International*, vol. 55, pp. 24–26, August 2010.

[5] H. Kinoshita, R. Tayama, Y. Kometani, T. Asano, and Y. Kani, “Development of new technology for fukushima daiichi nuclear power station reconstruction,”

Hitachi Hyoron, vol. 95, pp. 41–46, 2013.

[6] N. E. Hitachi-GE *et al.*, “Double-arm heavy machinery-type robot’astaco-sora’,” *E-Journal of Advanced Maintenance*, vol. 6, no. 1, 2014.

[7] C. J. Taylor and D. Robertson, “State-dependent control of a hydraulically actuated nuclear decommissioning robot,” *Control Engineering Practice*, vol. 21, no. 12, pp. 1716–1725, 2013.

[8] D. Robertson, *State–dependent control of a hydraulically–actuated nuclear decommissioning robot.* PhD thesis, Lancaster University, Engineering Department, 2015.

[9] A. Montazeri and J. Ekotuyo, “Development of dynamic model of a 7dof hydraulically actuated tele-operated robot for decommissioning applications,” in *American Control Conference (ACC), 2016*, pp. 1209–1214, IEEE, 2016.

[10] M. Antoine, *Dynamic modelling and parameter estimation of a hydraulic actuated manipulator.* Msc thesis, Lancaster University, 2014.

[11] C. J. Taylor, P. C. Young, and A. Chotai, *True digital control: statistical modelling and non-minimal state space design.* John Wiley & Sons, 2013.

[12] E. Guzzo, “What is a robot? - robots: Your guide to the world of robotics.” https://robots.ieee.org/learn/, accessed 7 April 2019.

[13] M. Simon, “What is a robot?,” Aug 2017. https://www.wired.com/story/what-is-a-robot/, accessed 7 April 2019.

[14] World Nuclear News, “Nuclear power in the united kingdom.” Website, october 2014 2014. http://www.world-nuclear.org/info/Country-Profiles/Countries-T-Z/United-Kingdom/ , Accessed 6/11/2014.

[15] L. Cragg and H. Hu, “Application of mobile agents to robust teleoperation of internet robots in nuclear decommissioning,” in *Industrial Technology, 2003 IEEE International Conference on*, vol. 2, pp. 1214–1219, IEEE, 2003.

[16] Nuclear Energy Institure, “Decommissioning of nuclear power plants,” 2014. https://www.nei.org/CorporateSite/media/filefolder/Backgrounders/Fact-Sheets/Decommissioning-of-Nuclear-Power-Plants-August-2014.pdf?ext=.pdf.

[17] Sustainable development commission, “The role of nuclear power in a low carbon economy- paper 5: Waste and decommissioning,” 2006. http://www.sd-commission.org.uk/data/files/publications/SDC-NuclearPosition-2006.pdf.

[18] C. R. Bayliss, *Nuclear decommissioning, waste management, and environmental site remediation*. Boston, MA: Butterworth-Heinemann, 2003.

[19] Nuclear Regulatory Commission, “Subpart c–occupational dose limits,” 2014. http://www.nrc.gov/reading-rm/doc-collections/cfr/part020/part020-1201.html.

[20] World Nuclear Assosciation, “Decommissioning nuclear facilities,” 2014. http://www.world-nuclear.org/info/nuclear-fuel-cycle/nuclear-wastes/decommissioning-nuclear-facilities/.

[21] Magnox, "Berkeley-preparing for care and maintenance." http://www.magnoxsites.co.uk/site/berkeley/.

[22] OECD/NEA, "Radioactive waste management programmes in oecd/new member countries," 2011. https://www.oecd-nea.org/rwm/pubs/2005/5248-rwm-programmes-member-countries.pdf.

[23] I. Tsitsimpelis, C. J. Taylor, B. Lennox, and M. J. Joyce, "A review of ground-based robotic systems for the characterization of nuclear environments," *Progress in Nuclear Energy*, vol. 111, pp. 109–124, 2019.

[24] Edited by Michele Laraia, *Nuclear Decommissioning-Planning, execution and international experience.* Woodhead publishing series in energy, 2012.

[25] "Limits for intakes of radionuclides by workers.," 1979. ICRP Publication 30 (Part1), Ann. ICRP 2 (3-4).

[26] R. Bogue, "Robots in the nuclear industry: a review of technologies and applications," *Industrial Robot: An International Journal*, vol. 38, no. 2, pp. 113–118, 2011.

[27] B. Baker, "News: Robot removal," *The Engineer*, p. 10, 18 may 2009. https://www.theengineer.co.uk/issues/18-may-2009/robot-removal/, Accessed 20 Nov. 2014.

[28] S. Watson, B. Lennox, P. Green, A. Griffiths, A. Dikarev, J. E. Mendoza, T. Wright, M. Nancekievill, and B. Bird, "Robotic systems for remote charac-

terization and decommissioning," in *Waste Management Symposium Journal*, vol. 1, 2016.

[29] A. Griffiths, A. Dikarev, P. R. Green, B. Lennox, X. Poteau, and S. Watson, "Avexis aqua vehicle explorer for in-situ sensing," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 282–287, 2016.

[30] A. R. Jones, A. Griffiths, M. J. Joyce, B. Lennox, S. Watson, J.-i. Katakura, K. Okumura, K. Kim, M. Katoh, K. Nishimura, *et al.*, "On the design of a remotely-deployed detection system for reactor assessment at fukushima daiichi," in *Nuclear Science Symposium, Medical Imaging Conference and Room-Temperature Semiconductor Detector Workshop (NSS/MIC/RTSD), 2016*, pp. 1–4, IEEE, 2016.

[31] R. Buckingham and A. Graham, "Dextorous manipulators for nuclear inspection and maintenance-case study," *IEEE: Applied robotics for the power industry, 2010 1st international conference, Montreal*, pp. 1–6, 5-7 Oct 2010.

[32] OC Robotics, "Laser Snake 2." online, 2017. http://www.ocrobotics.com/lasersnake2/, accessed 24 Nov 2017.

[33] J. Ford, "Robot snake lasers offer safe decommissioning solution," *The Engineer (Online)*, 2013 Apr 17. https://www.theengineer.co.uk/issues/april-2013-online/robot-snake-lasers-offer-safe-decommissioning-solution/ Accessed 20 Nov. 2014.

[34] P. Hilton and A. Khan, “New developments in laser cutting for nuclear decommissioning,” in *WM2014 conference, March*, 2014.

[35] Areva, “Decommissioning of nuclear installations: world first for the robot charli,” 2013. http://www.areva.com/EN/news-10083/decommissioning-of-nuclear-installations-world-first-for-the-robot-charli.html?xtmc=charli&xtcr=3, accessed 20 Nov. 2014.

[36] World nuclear News, “Charli cleans up at creys-malville,” 2013. http://www.world-nuclear-news.org/C-Charli-cleans-up-at-Creys-Malville-0612137.html Accessed 20 Nov. 2014.

[37] Cybernetix, “Nuclear products.” http://www.cybernetix.fr/en/nucleaire/produits, accessed 07/01/2015.

[38] “Industrial dual-armed robot for mining, welding & more - guardian GT.” Sarcos, https://www.sarcos.com/products/guardian-gt/, accessed 30-10-2017.

[39] B. Wolff, “Why human-controlled, force-multiplying robots are the future of work on earth.” IEEE Spectrum, https://spectrum.ieee.org/automaton/robotics/industrial-robots/why-human-controlled-force-multiplying-robots-are-the-future-of-work-on-earth, Accessed 30-10-2017.

[40] Epson, “See, sense, think, and react ”autonomous dual-arm robot”.” http://global.epson.com/products/robots/dualarmrobot/, accessed 24/11/2014.

[41] ABB, “Yumi-creating an automated future together.” http://new.abb.com/products/robotics/yumi, accessed 24/11/2014.

[42] E. Guizzo and E. Ackerman, “The rise of the robot worker,” *IEEE Spectrum*, vol. 49, no. 10, 2012.

[43] J. Montague, “Semi-autonomous welding robot will repair new york’s steam pipes,” *Control Engineering*, vol. 48, no. 9, p. 8, 2001.

[44] E. M. Shaban, S. Ako, C. J. Taylor, and D. W. Seward, “Development of an automated verticality alignment system for a vibro-lance,” *Automation in Construction*, vol. 17, no. 5, pp. 645–655, 2008.

[45] R. Dixon, A. Chotai, P. C. Young, and J. N. Scott, “The automation of piling rig positioning utilising multivariable proportional-integral-plus (PIP) control,” in *Proceedings 12th International Conference on Systems Engineering (ICSE 1997)*, (9-11 September, Coventry University, UK), 1997.

[46] C. Marques, J. Cristovao, P. Lima, J. Frazao, I. Ribeiro, and R. Ventura, “Raposa: Semi-autonomous robot for rescue operations,” in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 3988–3993, 2006.

[47] Z. Mughal, S. K. Garan, and R. Kamoua, “Eagle o: A semi-autonomous robot,” in *Systems, Applications and Technology Conference (LISAT), 2011 IEEE Long Island*, pp. 1–5, 2011.

[48] N. Marturi, A. Rastegarpanah, C. Takahashi, M. Adjigble, R. Stolkin, S. Zurek, M. Kopicki, M. Talha, J. A. Kuo, and Y. Bekiroglu, “Towards advanced robotic

manipulation for nuclear decommissioning: A pilot study on tele-operation and autonomy," in *2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA)*, pp. 1–8, Dec 2016.

[49] A. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 1–8, 2012.

[50] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pp. 763–770, IEEE, 2013.

[51] R. L. A. Shauri, K. Saiki, S. Toritani, and K. Nonami, "Sensor integration and fusion for autonomous screwing task by dual-manipulator hand robot," *Procedia Engineering*, vol. 41, no. 0, pp. 1412–1420, 2012.

[52] G. Du and P. Zhang, "Markerless humanrobot interface for dual robot manipulators using kinect sensor," *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 2, pp. 150–159, 2014.

[53] P. Neto, J. N. Pires, and A. P. Moreira, "High-level programming and control for industrial robotics: using a hand-held accelerometer-based input device for gesture and posture recognition," *The Industrial Robot*, vol. 37, no. 2, pp. 137–147, 2010.

[54] N. Checka, S. Schaffert, D. Demirdjian, J. Falkowski, and D. H. Grollman, "Handheld operator control unit," in *7th ACM/IEEE International conference*

on human-robot interaction. Boston, Massachusetts, USA, vol. LBR Highlights, 2012.

[55] G. Randelli, M. Venanzi, and D. Nardi, "Tangible interfaces for robot teleoperation," in *Human-Robot Interaction (HRI), 2011 6th ACM/IEEE International Conference on*, pp. 231–232, 2011.

[56] A. Eliav, T. Lavie, Y. Parmet, H. Stern, and Y. Edan, "Advanced methods for displays and remote control of robots," *Appl Ergon*, vol. 42, no. 6, pp. 820–9, 2011.

[57] "Kinect 2 with oculus rift gets nasa robotic arm workout," 2013. http://phys.org/news/2013-12-kinect-oculus-rift-nasa-robotic.html, accessed 6/11/2014.

[58] C. Loughlin, A. AlbuSchffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimbck, and G. Hirzinger, "The dlr lightweight robot: design and control concepts for robots in human environments," *Industrial Robot: An International Journal*, vol. 34, no. 5, pp. 376–385, 2007.

[59] Honda Robotics, "Asimo-the worlds most advanced humanoid robot." http://asimo.honda.com/ , accessed 11/12/2014.

[60] E and T, The IET, "Honda unveils e2-dr, its first humanoid disaster-relief robot." Website, Oct 2017. https://eandt.theiet.org/content/articles/2017/10/honda-unveils-e2-dr-its-first-humanoid-disaster-relief-robot/ Accessed 30 jan 2018.

[61] Boston Dynamics, “Atlas - the agile anthropomorphic robot.” http://www.bostondynamics.com/robot-Atlas.html, accessed 11/12,2014.

[62] G. M. Atmeh, I. Ranatunga, D. O. Popa, K. Subbarao, F. Lewis, and P. Rowe, “Implementation of an adaptive, model free, learning controller on the atlas robot,” in *American Control Conference (ACC), 2014*, pp. 2887–2892, 2014.

[63] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, “Optimization based full body control for the atlas robot,” in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pp. 120–127, IEEE, 2014.

[64] G. Nelson, A. Saunders, N. Neville, B. Swilling, J. Bondaryk, D. Billings, C. Lee, R. Playter, and M. Raibert, “Petman: A humanoid robot for testing chemical protective clothing,” *Journal of the Robotics Society of Japan*, vol. 30, no. 4, pp. 372–377, 2012.

[65] S. Sangok, A. Wang, C. Meng Yee, D. Otten, J. Lang, and K. Sangbae, “Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 3307–3312, 2013.

[66] MIT, “Biomimetic robotics lab.” http://biomimetics.mit.edu/, accessed 11/12/2014.

[67] Boston Dynamics, “Bigdog - the most advanced rough-terrain robot on earth.” http://www.bostondynamics.com/robot-bigdog.html. accessed 11/12/2014.

[68] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert, "Autonomous navigation for bigdog," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 4736–4741, 2010.

[69] M. Raibert, K. Blankespoor, G. Nelson, R. Playter, and t. B. Team, "Bigdog, the rough-terrain quadraped robot," in *17th World Congress The International Federation of Automatic Control. 2008: Seoul, korea.*, July 6-11 2008.

[70] M. Li, Z. Jiang, P. Wang, L. Sun, and S. S. Ge, "Control of a quadruped robot with bionic springy legs in trotting gait," *Journal of Bionic Engineering*, vol. 11, no. 2, pp. 188–198, 2014.

[71] University of Manchester, "UOM Robotics." Website, 2017. http://uomrobotics.com/research/index.html, Accessed 22 Nov. 2017.

[72] B. L. Luk, D. S. Cooke, S. Galt, A. A. Collie, and S. Chen, "Intelligent legged climbing service robot for remote maintenance applications in hazardous environments," *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 142–152, 2005.

[73] B. L. Luk, K. Liu, A. A. Collie, D. S. Cooke, and S. Chen, "Tele-operated climbing and mobile service robots for remote inspection and maintenance in nuclear industry," *Industrial Robot: An International Journal*, vol. 33, no. 3, pp. 194–204, 2006.

[74] H. Marvi, C. Gong, N. Gravish, H. Astley, M. Travers, R. L. Hatton, J. R. Mendelson, H. Choset, D. L. Hu, and D. I. Goldman, "Sidewinding with minimal

slip: Snake and robot ascent of sandy slopes," *Science*, vol. 346, no. 6206, pp. 224–229, 2014.

[75] A. Johnson, C. Wright, M. Tesch, K. Lipkin, and H. Choset, "A novel architecture for modular snake robots," *Citeseer, Tech. Rep.*, 2011.

[76] N. Daye, "Snake arm takes robotics inside aerospace," *Plant*, vol. 66, no. 1, pp. 17–18, 2007.

[77] The Engineer, "Snake-arm robot set to go through its paces at sellafield," *The Engineer (Online)*, 2011 Jul 05 2011. Accessed 21 November 2017.

[78] OCRobotics, "Jetsnake: Snake arms for tunnel boring machines." http://www.ocrobotics.com/applications–solutions/construction/construction-case-study-jetsnake/ accessed 7/1/2015.

[79] E. Garcia, M. A. Jimenez, P. G. De Santos, and M. Armada, "The evolution of robotics research," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 1, pp. 90–103, 2007.

[80] S. Xie, E. Haemmerle, Y. Cheng, and P. Gamage, "Vision-guided robot control for 3d object recognition and manipulation," *Robot Manipulators, InTech, Rijeka/Shanghai*, pp. 521–546, 2008.

[81] M. Prabakar and J.-H. Kim, "Telebot: Design concept of telepresence robot for law enforcement," in *Advances in Nano, Biomechanics, Robotics and Energy resarch (ANBRE13) 2013: Seoul, Korea*, 2013.

[82] C. Evans-Pughe, “I think, therefore,” *Engineering & Technology*, vol. 4, no. 2, pp. 24–26, 2009.

[83] P. ChanHun and P. KyoungTaik, “Design and kinematics analysis of dual arm robot manipulator for precision assembly,” in *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on*, pp. 430–435.

[84] L. Sukhan, “Dual redundant arm configuration optimization with task-oriented dual arm manipulability,” *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 1, pp. 78–97, 1989.

[85] J. Zhou, X. Ding, and Y. Q. Yu, “Automatic planning and coordinated control for redundant dual-arm space robot system,” *The Industrial Robot*, vol. 38, no. 1, pp. 27–37, 2011.

[86] C. Taeyong, L. Joon-Woo, K. Doohyung, D. Hyunmin, P. Dongil, and P. Chanhun, “Development of redundant shoulder complex of human-like robot driven by flexible wire tendons,” *Electronics Letters*, vol. 50, no. 20, pp. 1417–1419, 2014.

[87] D. Robertson, C. J. Taylor, and C. Lokuciewski, “State–dependent system identification for control of a hydraulically–actuated nuclear decommissioning robot,” *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 1233–1238, 2012.

[88] D. Robertson and C. J. Taylor, “Low–level joint control of a robotic manipulator using state–dependent parameter models,” in *22nd International Conference on Systems Engineering (ICSE)*, (Coventry, UK), September 2012.

[89] D. Robertson and C. J. Taylor, "State–dependent control of a hydraulically–actuated nuclear decommissioning robot," in *9th UKACC International Control Conference*, (Cardiff, UK), September 2012.

[90] S. Yogeswaran, "Integration of force feedback into nuclear decommissioning robots," Master of Science, Engineering Department, Lancaster University, 2014.

[91] P. Besset and C. J. Taylor, "Inverse kinematics for a redundant robotic manipulator used for nuclear decommissioning," in *10th UKACC International Control Conference*, (Loughborough, UK), July 2014.

[92] T. Burrell, A. Montazeri, S. Monk, and C. J. Taylor, "Feedback control–based inverse kinematics solvers for a nuclear decommissioning robot," in *7th IFAC Symposium on Mechatronic Systems*, (Loughborough, UK), September 2016.

[93] J. U. Ekot-Uyo, *Dynamic modelling and parameter estimation of an articulated manipulator*. Msc thesis, Lancaster University, 2015.

[94] J. Swevers, C. Ganseman, J. De Schutter, and H. Van Brussel, "Experimental robot identification using optimised periodic trajectories," *Mechanical Systems and Signal Processing*, vol. 10, no. 5, pp. 561–577, 1996.

[95] C. J. Taylor, E. Shaban, M. Stables, and S. Ako, "Proportional-integral-plus control applications of state-dependent parameter models," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 221, no. 7, pp. 1019–1031, 2007.

[96] P. C. Young, *Recursive estimation and time-series analysis: An introduction for the student and practitioner.* Springer Science & Business Media, 2011.

[97] M. Gautier and P. Poignet, "Extended kalman filtering and weighted least squares dynamic identification of robot," *Control Engineering Practice*, vol. 9, no. 12, pp. 1361–1372, 2001.

[98] W. Khalil and E. Dombre, *Modeling, identification and control of robots.* Butterworth-Heinemann, 2004.

[99] J. Swevers, W. Verdonck, and J. De Schutter, "Dynamic model identification for industrial robots," *IEEE Control Systems*, vol. 27, no. 5, pp. 58–71, 2007.

[100] M. Gautier, A. Janot, and P.-O. Vandanjon, "A new closed-loop output error method for parameter identification of robot dynamics," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 2, pp. 428–444, 2013.

[101] A. Janot, P.-O. Vandanjon, and M. Gautier, "A generic instrumental variable approach for industrial robot identification," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 1, pp. 132–145, 2014.

[102] M. Gautier, A. Jubien, and A. Janot, "New closed-loop output error method for robot joint stiffness identification with motor force/torque data," in *Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on*, pp. 592–597, IEEE, 2012.

[103] E. Tohme, R. Ouvrard, A. Abche, J.-C. Trigeassou, T. Poinot, and G. Mercère, "A methodology to enhance the convergence of output error identification algo-

rithms," in *Control Conference (ECC), 2007 European*, pp. 5721–5728, IEEE, 2007.

[104] M. Affenzeller, S. Wagner, S. Winkler, and A. Beham, *Genetic algorithms and genetic programming: modern concepts and practical applications.* Crc Press, 2009.

[105] C. C. Coello, "Evolutionary multi-objective optimization: a historical view of the field," *IEEE computational intelligence magazine*, vol. 1, no. 1, pp. 28–36, 2006.

[106] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[107] K. E. Parsopoulos, *Particle swarm optimization and intelligence: advances and applications: advances and applications.* IGI global, 2010.

[108] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE transactions on evolutionary computation*, vol. 15, no. 1, pp. 4–31, 2011.

[109] A. Gotmare, S. S. Bhattacharjee, R. Patidar, and N. V. George, "Swarm and evolutionary computing algorithms for system identification and filter design: A comprehensive review," *Swarm and Evolutionary Computation*, 2016.

[110] L. Yao and W. A. Sethares, “Nonlinear parameter estimation via the genetic algorithm,” *IEEE Transactions on signal processing*, vol. 42, no. 4, pp. 927–935, 1994.

[111] E. K. Nyarko and R. Scitovski, “Solving the parameter identification problem of mathematical models using genetic algorithms,” *Applied mathematics and Computation*, vol. 153, no. 3, pp. 651–658, 2004.

[112] A. Montazeri, J. Poshtan, and A. Yousefi-Koma, “The use of particle swarmto optimize the control system in a pzt laminated plate,” *Smart Materials and Structures*, vol. 17, no. 4, p. 045027, 2008.

[113] M. Ye, “Parameter identification of dynamical systems based on improved particle swarm optimization,” *Intelligent Control and Automation*, pp. 351–360, 2006.

[114] J. Sun and X. Liu, “A novel apso-aided maximum likelihood identification method for hammerstein systems,” *Nonlinear Dynamics*, vol. 73, no. 1-2, pp. 449–462, 2013.

[115] D. Krusienski and W. Jenkins, “Design and performance of adaptive systems based on structured stochastic optimization strategies,” *IEEE Circuits and Systems Magazine*, vol. 5, no. 1, pp. 8–20, 2005.

[116] C. Mavroidis, J. Flanz, S. Dubowsky, P. Drouet, and M. Goitein, “High performance medical robot requirements and accuracy analysis,” *Robotics and Computer-Integrated Manufacturing*, vol. 14, no. 5-6, pp. 329–338, 1998.

[117] G. Calafiore and M. Indri, "Robust calibration and control of robotic manipulators," in *American Control Conference, 2000. Proceedings of the 2000*, vol. 3, pp. 2003–2007, IEEE, 2000.

[118] A. M. Hosseini, S. Arzanpour, F. Golnaraghi, and A. M. Parameswaran, "Solenoid actuator design and modeling with application in engine vibration isolators," *Journal of Vibration and Control*, vol. 19, no. 7, pp. 1015–1023, 2013.

[119] J. Watton, *Fundamentals of fluid power control*, vol. 10. Cambridge University Press, 2009.

[120] M. T. Jensen, "Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation," *Journal of Mathematical Modelling and Algorithms*, vol. 3, no. 4, pp. 323–347, 2004.

[121] D. F. Lochtefeld and F. W. Ciarallo, "Multi-objectivization via decomposition: An analysis of helper-objectives and complete decomposition," *European Journal of Operational Research*, vol. 243, no. 2, pp. 395–404, 2015.

[122] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[123] A. Montazeri and J. Poshtan, "Ga-based optimization of a mimo anc system considering coupling of secondary sources in a telephone kiosk," *Applied Acoustics*, vol. 70, no. 7, pp. 945–953, 2009.

[124] V. Grecu, L. Grecu, G. Demian, M. Demian, N. Mastorakis, and O. Martin, "A genetic algorithm for optimization in conceptual robot manipulator," in *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*, WSEAS, 2009.

[125] A. Jafari, M. Safavi, and A. Fadaei, "A genetic algorithm to optimum dynamic performance of industrial robots in the conceptual design phase," in *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on*, pp. 1129–1135, 2009.

[126] H. V. Hultmann Ayala and L. dos Santos Coelho, "Tuning of pid controller based on a multiobjective genetic algorithm applied to a robotic manipulator," *Expert Systems with Applications*, vol. 39, no. 10, pp. 8968–8974, 2012.

[127] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-based robot grasping with the google object recognition engine," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 4263–4270, IEEE, 2013.

[128] M. Talha, E. Ghalamzan, C. Takahashi, J. Kuo, W. Ingamells, and R. Stolkin, "Towards robotic decommissioning of legacy nuclear plant: Results of human-factors experiments with tele-robotic manipulation, and a discussion of challenges and approaches for decommissioning," in *Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on*, pp. 166–173, IEEE, 2016.

[129] S. Charoenseang, A. Srikaew, D. M. Wilkes, and K. Kawamura, “Integrating visual feedback and force feedback in 3-d collision avoidance for a dual-arm humanoid robot,” in *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, vol. 4, pp. 3406–3411 vol.4, 1998.

[130] F. Reme, “Advanced remote control of industrial robots,” Master’s thesis, Institutt for produksjons-og kvalitetsteknikk, 2012.

[131] A. Hossain and T. Zaman, “HMI design: An analysis of a good display for seamless integration between user understanding and automatic controls,” in *American Society for Engineering Education*, American Society for Engineering Education, 2012.

[132] D. G. Caldwell, A. Wardle, and M. Goodwin, “Tele-presence: visual, audio and tactile feedback and control of a twin armed mobile robot,” in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 244–249 vol.1, 1994.

[133] D. A. Kontarinis, J. S. Son, W. Peine, and R. D. Howe, “A tactile shape sensing and display system for teleoperated manipulation,” in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 1, pp. 641–646 vol.1, 1995.

[134] G. Borenstein, *Making things see-3D vision with Kinect, Processing, Arduino, and MakerBot.* O’Reilly Media, 2012.

[135] M. Wasielica, M. Wasik, A. Kasinski, and P. Skrzypczynski, "Interactive programming of a mechatronic system: A small humanoid robot example," in *Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on*, pp. 459–464, 2013.

[136] M. Biao, X. Wensheng, and W. Songlin, "A robot control system based on gesture recognition using kinect," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 11, no. 5, pp. 2605–2611, 2013.

[137] K. Qian, J. Niu, and H. Yang, "Developing a gesture based remote human-robot interaction system using kinect," *International Journal of Smart Home*, vol. 7, no. 4, 2013.

[138] P. Benavidez and M. Jamshidi, "Mobile robot navigation and target tracking system," in *System of Systems Engineering (SoSE), 2011 6th International Conference on*, pp. 299–304, 2011.

[139] D. S. O. Correa, D. F. Sciotti, M. G. Prado, D. O. Sales, D. F. Wolf, and F. S. Osorio, "Mobile robots navigation in indoor environments using kinect sensor," in *Critical Embedded Systems (CBSEC), 2012 Second Brazilian Conference on*, pp. 36–41, 2012.

[140] I. Siradjuddin, L. Behera, T. M. McGinnity, and S. Coleman, "A position based visual tracking system for a 7 dof robot manipulator using a kinect camera," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pp. 1–7, 2012.

[141] F. Husain, A. Colom, B. Dellen, G. Alenya, and C. Torras, "Realtime tracking and grasping of a moving object from range video," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 2617–2622, IEEE, 2014.

[142] D. Kent, C. Saldanha, and S. Chernova, "A comparison of remote robot teleoperation interfaces for general object manipulation," 2017.

[143] D. J. Bruemmer, J. L. Marble, D. D. Dudenhoeffer, M. Anderson, and M. D. McKay, "Mixed-initiative control for remote characterization of hazardous environments," in *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pp. 9–pp, IEEE, 2003.

[144] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[145] Y. Xu, G. Fang, S. Chen, J. J. Zou, and Z. Ye, "Real-time image processing for vision-based weld seam tracking in robotic gmaw," *The International Journal of Advanced Manufacturing Technology*, vol. 73, no. 9, pp. 1413–1425, 2014.

[146] C. Y. Hu, C. R. Chen, C. H. Tseng, A. P. Yudha, and C. H. Kuo, "Visual servoing spanner picking and placement with a scara manipulator," in *2016 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1632–1637, 2016.

[147] R. Maini and H. Aggarwal, "Study and comparison of various image edge detection techniques," *International journal of image processing (IJIP)*, vol. 3, no. 1, pp. 1–11, 2009.

[148] S. R. Buss, “Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods,” *IEEE Journal of Robotics and Automation*, vol. 17, no. 1-19, p. 16, 2004.

[149] P. Besset, *Inverse Kinematics for a redundant hydraulic robotic manipulator.* Msc, Lancaster university, 2013.

[150] P. C. Young, P. McKenna, and J. Bruun, “Identification of non-linear stochastic systems by state dependent parameter estimation,” *International journal of control*, vol. 74, no. 18, pp. 1837–1857, 2001.

[151] A. Janot, P. C. Young, and M. Gautier, “Identification and control of electro-mechanical systems using state-dependent parameter estimation,” *International Journal of Control*, vol. 90, no. 4, pp. 643–660, 2017.

[152] J. D. Fortgang, L. E. George, and W. J. Book, “Practical implementation of a dead zone inverse on a hydraulic wrist,” in *ASME 2002 International Mechanical Engineering Congress and Exposition*, pp. 149–155, American Society of Mechanical Engineers, 2002.

[153] P. C. Young, W. Tych, and C. J. Taylor, “The captain toolbox for matlab,” *IFAC Proceedings Volumes*, vol. 42, no. 10, pp. 758–763, 2009.

[154] M. d. C. R. Liñán, *New control design and analysis techniques for plants with actuator nonlinearities.* PhD thesis, University of Manchester, 2013.

[155] C. West, E. D. Wilson, Q. Clairon, S. Monk, A. Montazeri, and C. J. Taylor, “State-dependent parameter model identification for inverse dead-zone control

of a hydraulic manipulator - submitted," in *18th IFAC Symposium on System Identification, (SYSID 2018)*, 2018.

[156] C. J. Taylor, A. Chotai, and P. C. Young, "Design and application of pip controllers: robust control of the ifac93 benchmark," *Transactions of the Institute of Measurement and Control*, vol. 23, no. 3, pp. 183–200, 2001.

[157] C. Taylor, P. Young, and A. Chotai, "PIP optimal control with a risk sensitive criterion," in *UKACC international conference on control, Control '96 p. 959-964*, IET, 1996.

[158] N. Pitala-Daz, E. J. Herrera-Lpez, G. Valencia-Palomo, A. Gonzlez-Angeles, R. A. Rodrguez-Carvajal, and N. R. Cazarez-Castro, "Comparative analysis between conventional pi and fuzzy logic pi controllers for indoor benzene concentrations," *Sustainability*, vol. 7, no. 5, pp. 5398–5412, 2015.

[159] T. Burrell, C. West, S. D. Monk, A. Montezeri, and C. J. Taylor, "Towards a co-operative robotic system for autonomous pipe cutting in nuclear decommissioning," in *2018 UKACC 12th International Conference on Control (CONTROL)*, pp. 283–288, IEEE, 2018.

Appendix A

Labview Programs

Table A.1: *List of the key Labview programs created during this project*

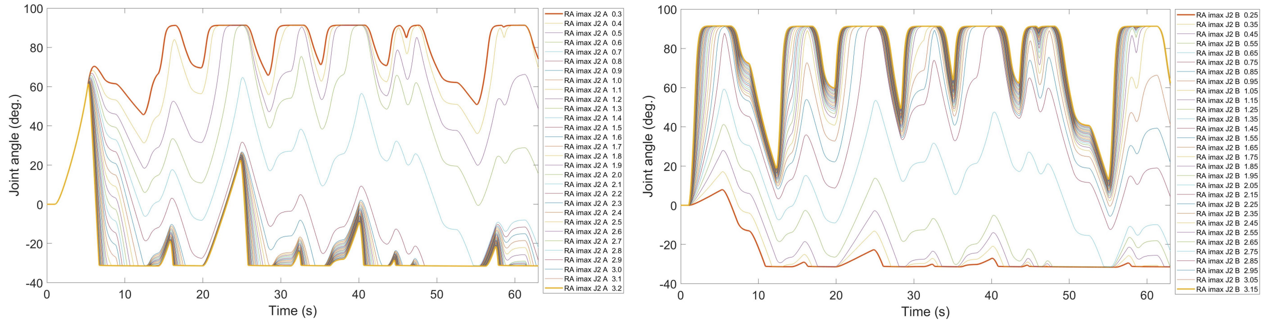
Filename	Description
Joystick final model_CW	Provides joystick control of both manipulators
Slider_interface_control	Control both arms using sliders or text boxes on GUI
Deg2Pot_left Deg2Pot_right	Converts degrees to potentiometer reading
Pot2Deg_left Pot2Deg_right	Converts potentiometer reading to degrees
Monitor	Display all potentiometer readings and angles in degrees

PIP_AdHoc_control_J1	
PIP_AdHoc_control_J2	Control of joint using PIP control with old style
PIP_AdHoc_control_J3	scaling, reading set point from text file
PIP_AdHoc_control_J4	
PIP_AdHoc_control_ALL	Control of 4 joint simultaneously
PIP_IDZ_control_J1	
PIP_IDZ_control_J2	Control of joint using PIP control with IDZ
PIP_IDZ_control_J3	scaling, reading set point from text file
PIP_IDZ_control_J4	
PIP_IDZ_control_ALL	Control of 4 joint simultaneously
PIP_IDZ_control_J1_FP	
PIP_IDZ_control_J2_FP	Control of joint using PIP control with IDZ
PIP_IDZ_control_J3_FP	scaling, reading set point from text file
PIP_IDZ_control_J4_FP	
PIP_IDZ_FP_ALL	Control of 4 joints for implementation in other programs
PIP_IDZ_control1_All_FP	Control 4 joints for resolved motion, input from text file
txtFile_Input_Fuzzy	Controls individual joint using Fuzzy controller, set point from text file

txtFile_Input_Fuzzy_All	Controls all joints using Fuzzy controller, set point from text file
FuzzyControllerBlock	Fuzzy controller for implementation in other programs such as vision control
Vision_Control_Fuzzy	Receives set points from Matlab vision interface, using Fuzzy controller
Vision_Control_PI	Receives set points from Matlab vision interface, using PI controller
Vision_Control_PIP_AdHoc	Receives set points from Matlab vision interface, using PIP controller with old scaling
Vision_Control_PIP_IDZ	Receives set points from Matlab vision interface, using PIP controller with IDZ scaling
Vision_Control_PIP_IDZ_FP	Receives set points from Matlab vision interface, using PIP forward path controller with IDZ scaling

Appendix B

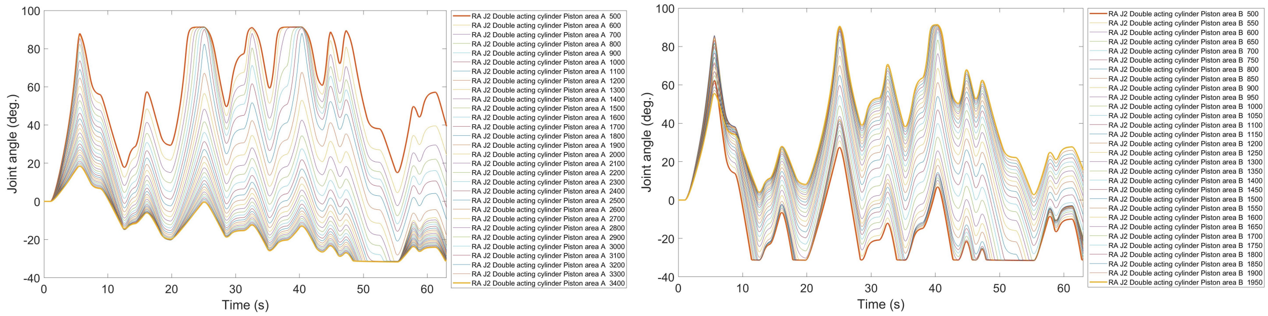
Sensitivity analysis



(a) Max proportional amplifier current A, range from 0.3 to 3.2 A

(b) Max proportional amplifier current B, range from 0.25 to 3.15 A

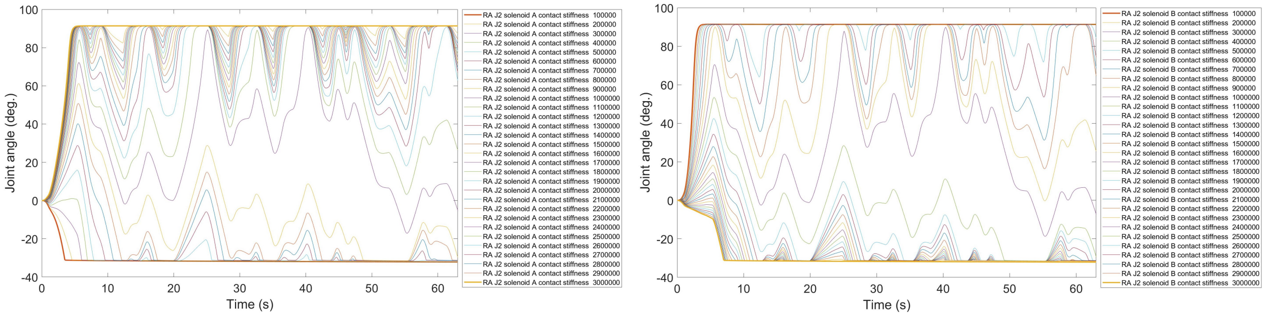
Figure B.1: Joint angle response to varying the max proportional amplifier current for both proportional amplifiers



(a) Piston area A, range from 500 to 3500 mm^2

(b) Piston area B, range from 500 to 2000 mm^2

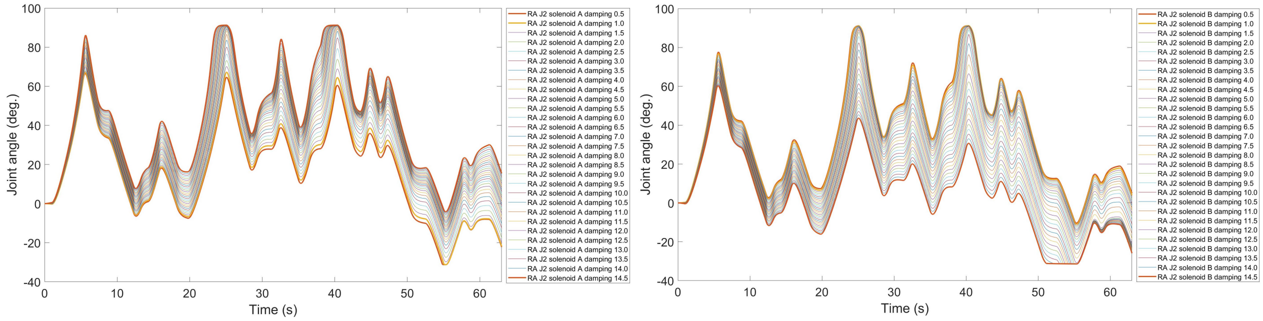
Figure B.2: Joint angle response to varying the piston areas A and B



(a) Solenoid A contact stiffness, range from 100000 to 3000000 N/m

(b) Solenoid B contact stiffness, range from 100000 to 3000000 N/m

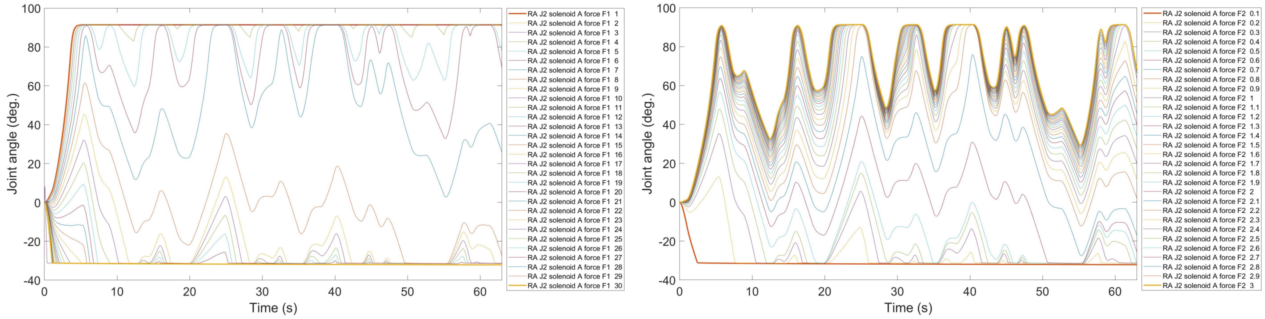
Figure B.3: Joint angle response to varying the solenoid A and B contact stiffness



(a) Solenoid A damping, range from 0 to 15 $N/(m/s)$

(b) Solenoid B damping, range from 0 to 15 $N/(m/s)$

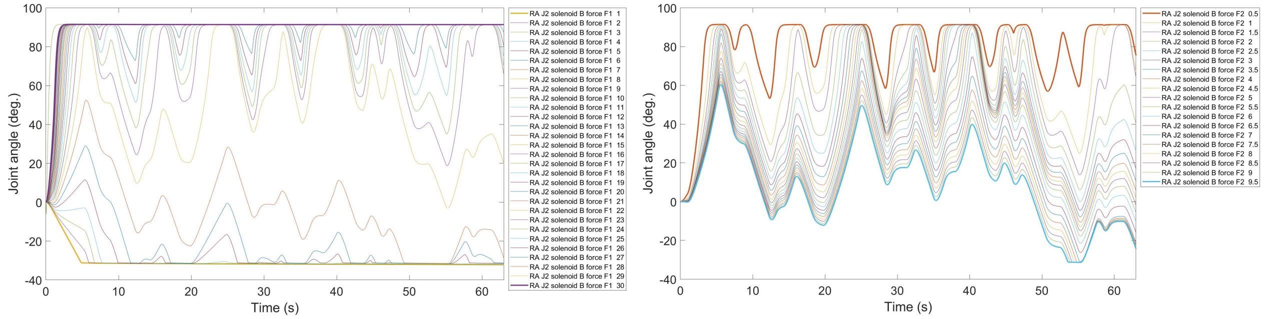
Figure B.4: Joint angle response to varying the solenoid A and B damping



(a) Solenoid A force F1 parameter, range from 1 to 30 N

(b) Solenoid A force F2 parameter, range from 0.1 to 3 N

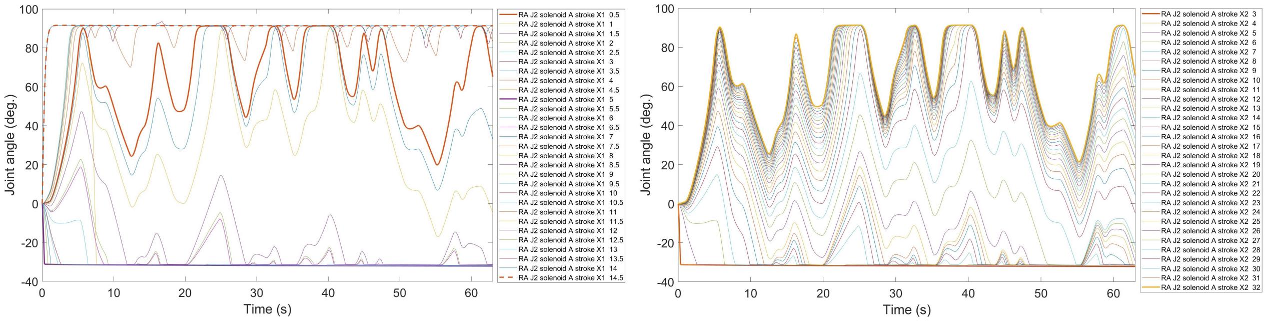
Figure B.5: Joint angle response to varying the solenoid A force F1 and F2 parameters



(a) Solenoid B force F1 parameter, range from 1 to 30 N

(b) Solenoid B force F2 parameter, range from 0.1 to 9.5 N

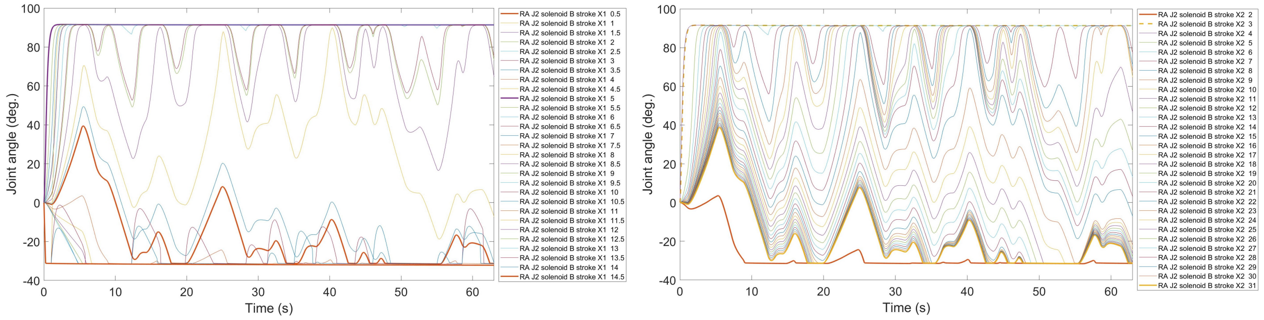
Figure B.6: Joint angle response to varying the solenoid B force F1 and F2 parameters



(a) Solenoid A stroke X1 parameter, range from 0.5 to 15 *mm*

(b) Solenoid A stroke X2 parameter, range from 3 to 32 *mm*

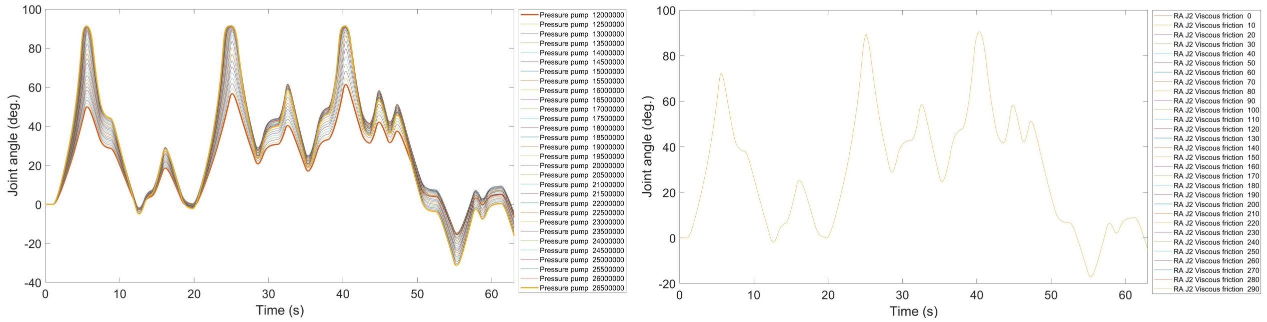
Figure B.7: Joint angle response to varying the solenoid A stroke X1 and X2 parameters



(a) Solenoid B stroke X1 parameter, range from 0.5 to 15 *mm*

(b) Solenoid B stroke X2 parameter, range from 2 to 31 *mm*

Figure B.8: Joint angle response to varying the solenoid B stroke X1 and X2 parameters



(a) Pump pressure, range from 12000000 to 26500000

(b) viscous friction, range from 0 to 290 $N/(m/s)$

Figure B.9: Example of parameters not included in GA, Pump pressure does have an effect on the joint response but is less pronounced than other parameters, and oil pressure of the real system can be measured to give a value for use in the model

Appendix C

Forward Kinematics Equations

Rotation matrices

$$\text{Joint 1} = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) \\ 0 & 1 & 0 \\ -\sin(\theta_1) & 0 & \cos(\theta_1) \end{bmatrix} \text{Joint 2} = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{Joint 3} = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Joint 4} = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) \\ 0 & 1 & 0 \\ -\sin(\theta_4) & 0 & \cos(\theta_4) \end{bmatrix} \text{Joint 5} = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{Joint 6} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Link geometry vector

$$Link_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} Link_1 = \begin{bmatrix} 70 \\ 0 \\ 0 \end{bmatrix} Link_2 = \begin{bmatrix} 523.48 \\ 0 \\ 0 \end{bmatrix} Link_3 = \begin{bmatrix} 165 \\ -212.60 \\ 0 \end{bmatrix} Link_4 = \begin{bmatrix} 44.45 \\ 0 \\ 0 \end{bmatrix} Link_5 = \begin{bmatrix} 284.80 \\ -18.47 \\ 0 \end{bmatrix} Link_6 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

End effector position

$$\begin{aligned}P = \begin{bmatrix}0\\0\\0\end{bmatrix} + \begin{bmatrix}\cos(\theta_1) & 0 & \sin(\theta_1)\\ 0 & 1 & 0\\ -\sin(\theta_1) & 0 & \cos(\theta_1)\end{bmatrix} \times \left(\begin{bmatrix}70\\0\\0\end{bmatrix} + \begin{bmatrix}\cos(\theta_2) & -\sin(\theta_2) & 0\\ \sin(\theta_2) & \cos(\theta_2) & 0\\ 0 & 0 & 1\end{bmatrix} \times \ldots\right.\\
\left(\begin{bmatrix}523.48\\0\\0\end{bmatrix} + \begin{bmatrix}\cos(\theta_3) & -\sin(\theta_3) & 0\\ \sin(\theta_3) & \cos(\theta_3) & 0\\ 0 & 0 & 1\end{bmatrix} \times \left(\begin{bmatrix}165\\-212.60\\0\end{bmatrix} + \begin{bmatrix}\cos(\theta_4) & 0 & \sin(\theta_4)\\ 0 & 1 & 0\\ -\sin(\theta_4) & 0 & \cos(\theta_4)\end{bmatrix} \times \ldots\right.\right.\\
\left.\left.\left.\left(\begin{bmatrix}44.45\\0\\0\end{bmatrix} + \begin{bmatrix}\cos(\theta_5) & -\sin(\theta_5) & 0\\ \sin(\theta_5) & \cos(\theta_5) & 0\\ 0 & 0 & 1\end{bmatrix} \times \left(\begin{bmatrix}284.80\\-18.47\\0\end{bmatrix} + \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} \times \begin{bmatrix}0\\0\\0\end{bmatrix}\right)\right)\right)\right)\right)\end{aligned} \tag{C.1}$$

Appendix D

Inverse Kinematics Equations

$$J = \begin{bmatrix} \frac{\delta f_x}{\delta \theta_1} & \frac{\delta f_x}{\delta \theta_2} & \frac{\delta f_x}{\delta \theta_3} & \frac{\delta f_x}{\delta \theta_4} & \frac{\delta f_x}{\delta \theta_5} \\ \\ \frac{\delta f_y}{\delta \theta_1} & \frac{\delta f_y}{\delta \theta_2} & \frac{\delta f_y}{\delta \theta_3} & \frac{\delta f_y}{\delta \theta_4} & \frac{\delta f_y}{\delta \theta_5} \\ \\ \frac{\delta f_z}{\delta \theta_1} & \frac{\delta f_z}{\delta \theta_2} & \frac{\delta f_z}{\delta \theta_3} & \frac{\delta f_z}{\delta \theta_4} & \frac{\delta f_z}{\delta \theta_5} \end{bmatrix} \tag{D.1}$$

$S1 = \sin(\theta_1)$ and $C1 = \cos(\theta_1)$ etc.

$$\begin{aligned} \frac{\delta f_x}{\delta \theta_1} = \quad & -S1 \times (C2 \times (S3 \times ((1847 \times C5)/100 - (1424 \times S5)/5 + 1063/5) + \dots \\ & C3 \times (C4 \times ((1424 \times C5)/5 + (1847 \times S5)/100 + 889/20) + 165) + \dots \\ & 13087/25) + S2 \times (C3 \times ((1847 \times C5)/100 - (1424 \times S5)/5 + \dots \\ & 1063/5) - S3 \times (C4 \times ((1424 \times C5)/5 + (1847 \times S5)/100 + 889/20) \dots \\ & + 165)) + 70) - C1 \times S4 \times ((1424 \times C5)/5 + (1847 \times S5)/100 + 889/20) \end{aligned}$$

$$\begin{aligned}\frac{\delta f_x}{\delta\theta_2} = \quad & C1 \times \big(C2 \times \big(C3 \times \big(\big(1847 \times C5\big)/100 - \big(1424 \times S5\big)/5 + 1063/5\big) - \ldots \\ & S3 \times \big(C4 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + 889/20\big) + 165\big)\big) - \ldots \\ & S2 \times \big(S3 \times \big(\big(1847 \times C5\big)/100 - \big(1424 \times S5\big)/5 + 1063/5\big) + \ldots \\ & C3 \times \big(C4 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + 889/20\big) + 165\big) + \ldots \\ & 13087/25\big)\big)\end{aligned}$$

$$\begin{aligned}\frac{\delta f_x}{\delta\theta_3} = \quad & C1 \times \big(C2 \times \big(C3 \times \big(\big(1847 \times C5\big)/100 - \big(1424 \times S5\big)/5 + 1063/5\big) - \ldots \\ & S3 \times \big(C4 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + 889/20\big) + 165\big)\big) - \ldots \\ & S2 \times \big(S3 \times \big(\big(1847 \times C5\big)/100 - \big(1424 \times S5\big)/5 + 1063/5\big) + \ldots \\ & C3 \times \big(C4 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + 889/20\big) + 165\big)\big)\big)\end{aligned}$$

$$\begin{aligned}\frac{\delta f_x}{\delta\theta_4} = \quad & -C1 \times \big(C2 \times C3 \times S4 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + \ldots \\ & 889/20\big) - S2 \times S3 \times S4 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + \ldots \\ & 889/20\big)\big) - C4 \times S1 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + 889/20\big)\end{aligned}$$

$$\begin{aligned}\frac{\delta f_x}{\delta\theta_5} = \quad & -C1 \times \big(C2 \times \big(S3 \times \big(\big(1424 \times C5\big)/5 + \ldots \\ & \big(1847 \times S5\big)/100\big) - C3 \times C4 \times \big(\big(1847 \times C5\big)/100 - \ldots \\ & \big(1424 \times S5\big)/5\big)\big) + S2 \times \big(C3 \times \big(\big(1424 \times C5\big)/5 + \ldots \\ & \big(1847 \times S5\big)/100\big) + C4 \times S3 \times \big(\big(1847 \times C5\big)/100 - \ldots \\ & \big(1424 \times S5\big)/5\big)\big)\big) - S1 \times S4 \times \big(\big(1847 \times C5\big)/100 - \big(1424 \times S55\big)/5\big)\end{aligned}$$

$$\frac{\delta f_y}{\delta \theta_1} = \quad 0$$

$$\begin{aligned}\frac{\delta f_y}{\delta \theta_2} = \quad & C2 \times \big(S3 \times \big(\big(1847 \times C5\big)/100 - \big(1424 \times S5\big)/5 + 1063/5\big) + \ldots \\ & C3 \times \big(C4 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + 889/20\big) + \ldots \\ & 165\big) + 13087/25\big) + S2 \times \big(C3 \times \big(\big(1847 \times C5\big)/100 - \ldots \\ & \big(1424 \times S5\big)/5 + 1063/5\big) - S3 \times \big(C4 \times \big(\big(1424 \times C5\big)/5 + \ldots \\ & \big(1847 \times S5\big)/100 + 889/20\big) + 165\big)\big)\end{aligned}$$

$$\begin{aligned}\frac{\delta f_y}{\delta \theta_3} = \quad & C2 \times \big(S3 \times \big(\big(1847 \times C5\big)/100 - \big(1424 \times S5\big)/5 + 1063/5\big) + \ldots \\ & C3 {\times} \big(C4 {\times} \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + 889/20\big) + \ldots \\ & 165\big)\big) + S2 \times \big(C3 \times \big(\big(1847 \times C5\big)/100 - \big(1424 \times S5\big)/5 + \ldots \\ & 1063/5\big) - S3 \times \big(C4 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + \ldots \\ & 889/20\big) + 165\big)\big)\end{aligned}$$

$$\begin{aligned}\frac{\delta f_y}{\delta \theta_4} = \quad & -C2 \times S3 \times S4 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + \ldots \\ & 889/20\big) - C3 \times S2 \times S4 \times \big(\big(1424 \times C5\big)/5 + \ldots \\ & \big(1847 \times S5\big)/100 + 889/20\big)\end{aligned}$$

$$\begin{aligned}\frac{\delta f_y}{\delta\theta_5} = \quad & C2 \times \big(C3 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100\big) + \ldots \\ & C4 \times S3 \times \big(\big(1847 \times C5\big)/100 - \big(1424 \times S5\big)/5\big)\big) - \ldots \\ & S2 \times \big(S3 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100\big) - \ldots \\ & C3 \times C4 \times \big(\big(1847 \times C5\big)/100 - \big(1424 \times S5\big)/5\big)\big)\end{aligned}$$

$$\begin{aligned}\frac{\delta f_z}{\delta\theta_1} = \quad & S1 \times S4 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + 889/20\big) - \ldots \\ & C1 \times \big(C2 \times \big(S3 \times \big(\big(1847 \times C5\big)/100 - \big(1424 \times S5\big)/5 + \ldots \\ & 1063/5\big) + C3 \times \big(C4 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + \ldots \\ & 889/20\big) + 165\big) + 13087/25\big) + S2 \times \big(C3 \times \big(\big(1847 \times C5\big)/100 - \ldots \\ & \big(1424 \times S5\big)/5 + 1063/5\big) - S3 \times \big(C4 \times \big(\big(1424 \times C5\big)/5 + \ldots \\ & \big(1847 \times S5\big)/100 + 889/20\big) + 165\big)\big) + 70\big)\end{aligned}$$

$$\begin{aligned}\frac{\delta f_z}{\delta\theta_2} = \quad & -S1 \times \big(C2 \times \big(C3 \times \big(\big(1847 \times C5\big)/100 - \big(1424 \times S5\big)/5 + 1063/5\big) - \ldots \\ & S3 \times \big(C4 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + 889/20\big) + \ldots \\ & 165\big)\big) - S2 \times \big(S3 \times \big(\big(1847 \times C5\big)/100 - \big(1424 \times S5\big)/5 + \ldots \\ & 1063/5\big) + C3 \times \big(C4 \times \big(\big(1424 \times C5\big)/5 + \big(1847 \times S5\big)/100 + \ldots \\ & 889/20\big) + 165\big) + 13087/25\big)\big)\end{aligned}$$

$$\begin{aligned}\frac{\delta f_z}{\delta \theta_3} = \quad & -S1 \times \bigl(C2 \times \bigl(C3 \times \bigl(\bigl(1847 \times C5\bigr)/100 - \bigl(1424 \times S5\bigr)/5 + \dots \\ & 1063/5\bigr) - S3 \times \bigl(C4 \times \bigl(\bigl(1424 \times C5\bigr)/5 + \bigl(1847 \times S5\bigr)/100 + \dots \\ & 889/20\bigr) + 165\bigr)\bigr) - S2 \times \bigl(S3 \times \bigl(\bigl(1847 \times C5\bigr)/100 - \dots \\ & \bigl(1424 \times S5\bigr)/5 + 1063/5\bigr) + C3 \times \bigl(C4 \times \bigl(\bigl(1424 \times C5\bigr)/5 + \dots \\ & \bigl(1847 \times S5\bigr)/100 + 889/20\bigr) + 165\bigr)\bigr)\bigr)\end{aligned}$$

$$\begin{aligned}\frac{\delta f_z}{\delta \theta_4} = \quad & S1 \times \bigl(C2 \times C3 \times S4 \times \bigl(\bigl(1424 \times C5\bigr)/5 + \bigl(1847 \times S5\bigr)/100 + \dots \\ & 889/20\bigr) - S2 \times S3 \times S4 \times \bigl(\bigl(1424 \times C5\bigr)/5 + \dots \\ & \bigl(1847 \times S5\bigr)/100 + 889/20\bigr)\bigr) - C1 \times C4 \times \dots \\ & \bigl(\bigl(1424 \times C5\bigr)/5 + \bigl(1847 \times S5\bigr)/100 + 889/20\bigr)\end{aligned}$$

$$\begin{aligned}\frac{\delta f_z}{\delta \theta_5} = \quad & S1 \times \bigl(C2 \times \bigl(S3 \times \bigl(\bigl(1424 \times C5\bigr)/5 + \bigl(1847 \times S5\bigr)/100 - \dots \\ & C3 \times C4 \times \bigl(\bigl(1847 \times C5\bigr)/100 - \bigl(1424 \times S5\bigr)/5\bigr)\bigr) + \dots \\ & S2 \times \bigl(C3 \times \bigl(\bigl(1424 \times C5\bigr)/5 + \bigl(1847 \times S5\bigr)/100\bigr) + \dots \\ & C4 \times S3 \times \bigl(\bigl(1847 \times C5\bigr)/100 - \bigl(1424 \times S5\bigr)/5\bigr)\bigr)\bigr) - \dots \\ & C1 \times S4 \times \bigl(\bigl(1847 \times C5\bigr)/100 - \bigl(1424 \times S5\bigr)/5\bigr)\end{aligned}$$

Appendix E

SDP Model Validation

Validation results for SDP model, 5 minute long pseudo random voltage inputs, 2 examples for each joints.

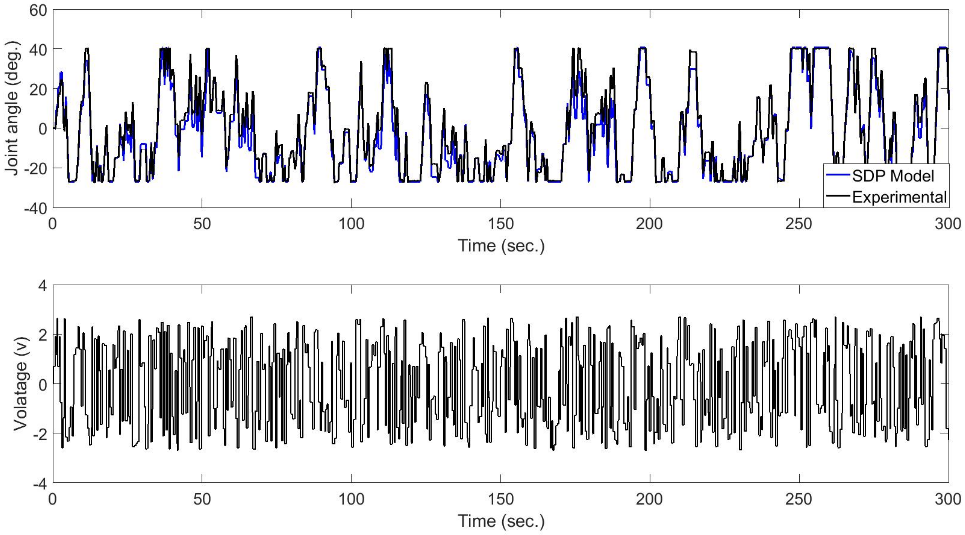


Figure E.1: *Joint 1 SDP model validation example 1, (Top)Comparison of SDP model and experimental output (bottom) pseudo random input voltage.*

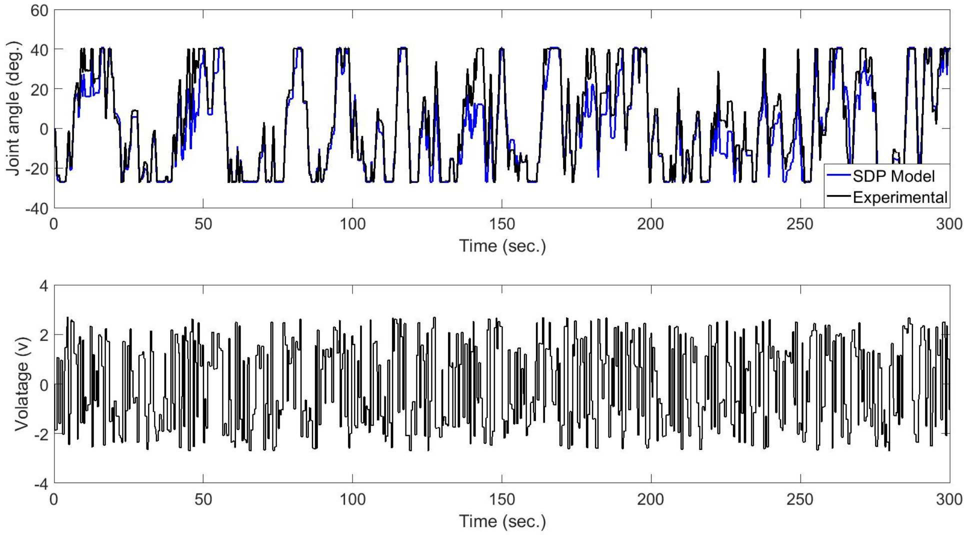


Figure E.2: *Joint 1 SDP model validation example 2, (Top)Comparison of SDP model and experimental output (bottom) pseudo random input voltage.*

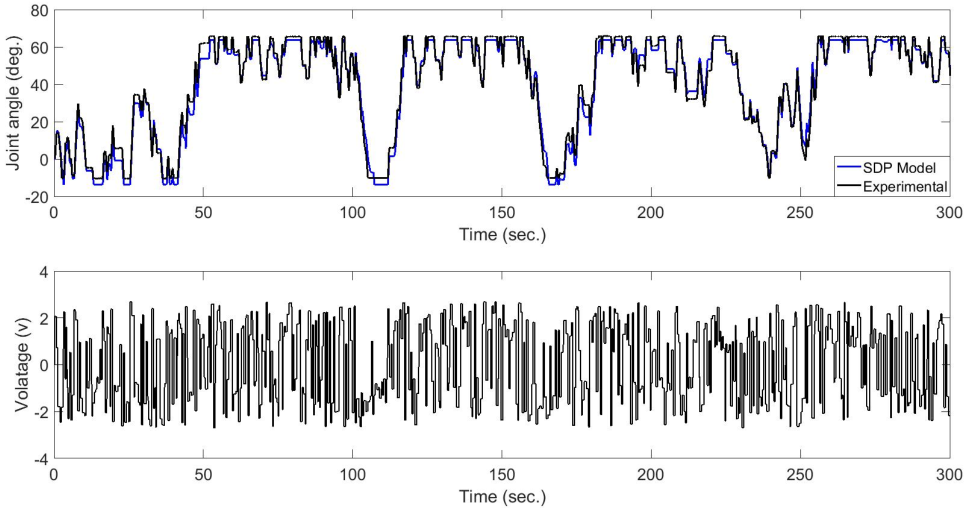


Figure E.3: *Joint 2 SDP model validation example 1, (Top)Comparison of SDP model and experimental output (bottom) pseudo random input voltage.*

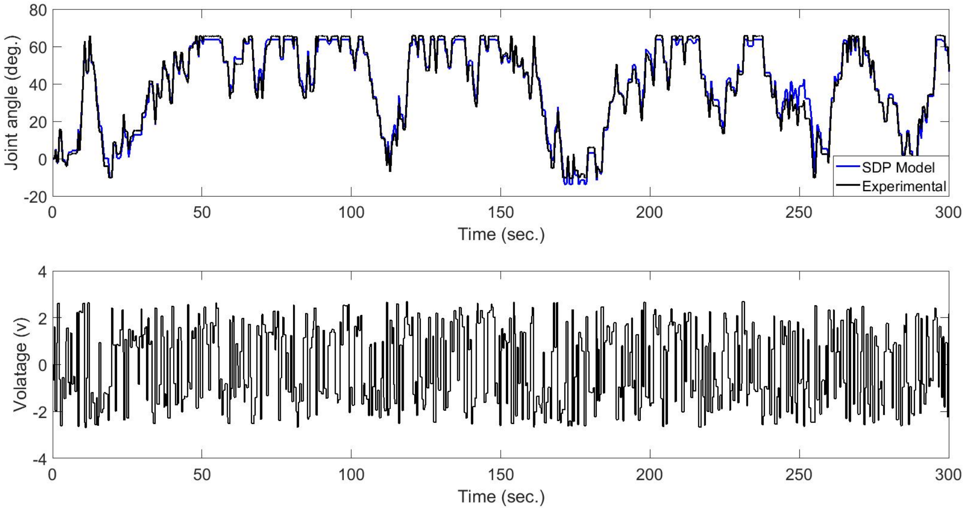


Figure E.4: *Joint 2 SDP model validation example 2, (Top)Comparison of SDP model and experimental output (bottom) pseudo random input voltage.*

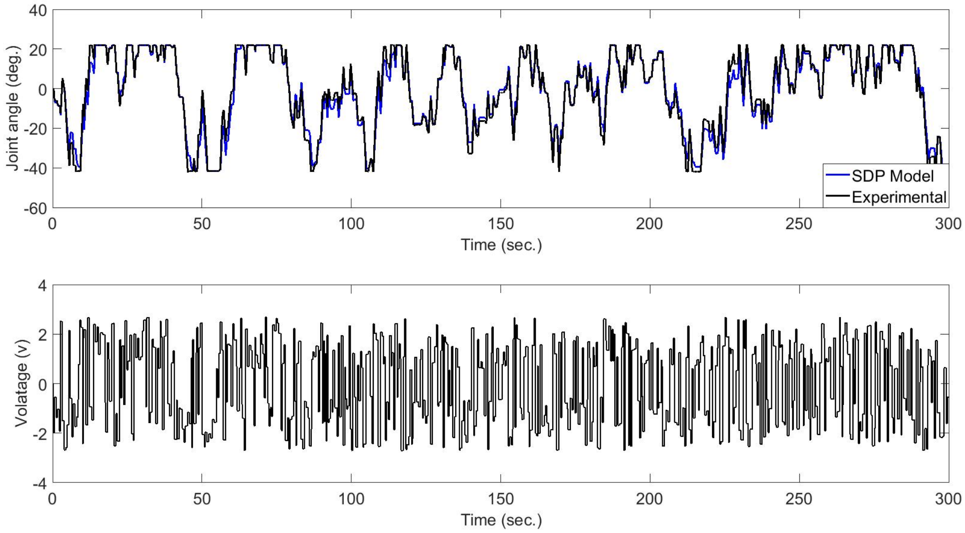


Figure E.5: *Joint 3 SDP model validation example 1, (Top)Comparison of SDP model and experimental output (bottom) pseudo random input voltage.*

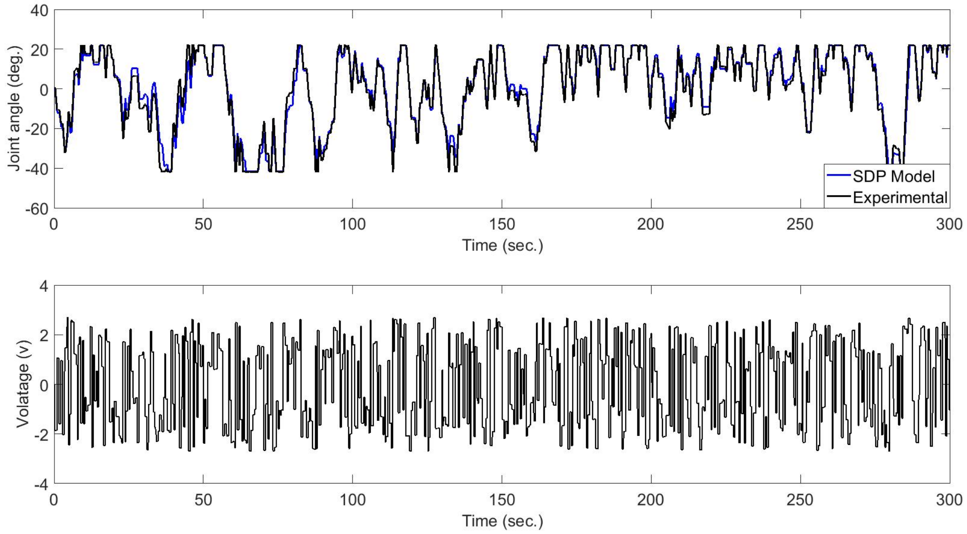


Figure E.6: *Joint 3 SDP model validation example 2, (Top)Comparison of SDP model and experimental output (bottom) pseudo random input voltage.*

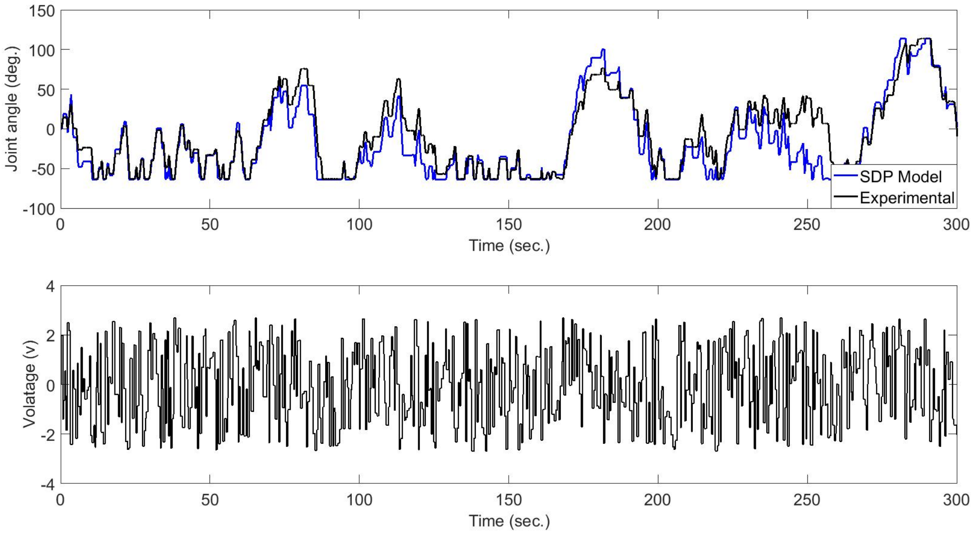


Figure E.7: *Joint 4 SDP model validation example 1, (Top)Comparison of SDP model and experimental output (bottom) pseudo random input voltage.*

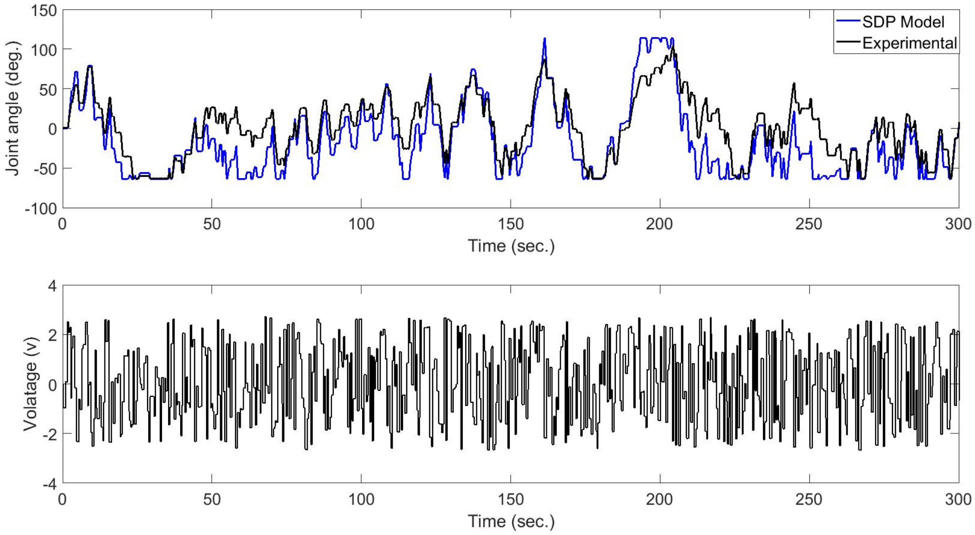


Figure E.8: *Joint 4 SDP model validation example 2, (Top)Comparison of SDP model and experimental output (bottom) pseudo random input voltage.*

Appendix F

Fuzzy membership functions

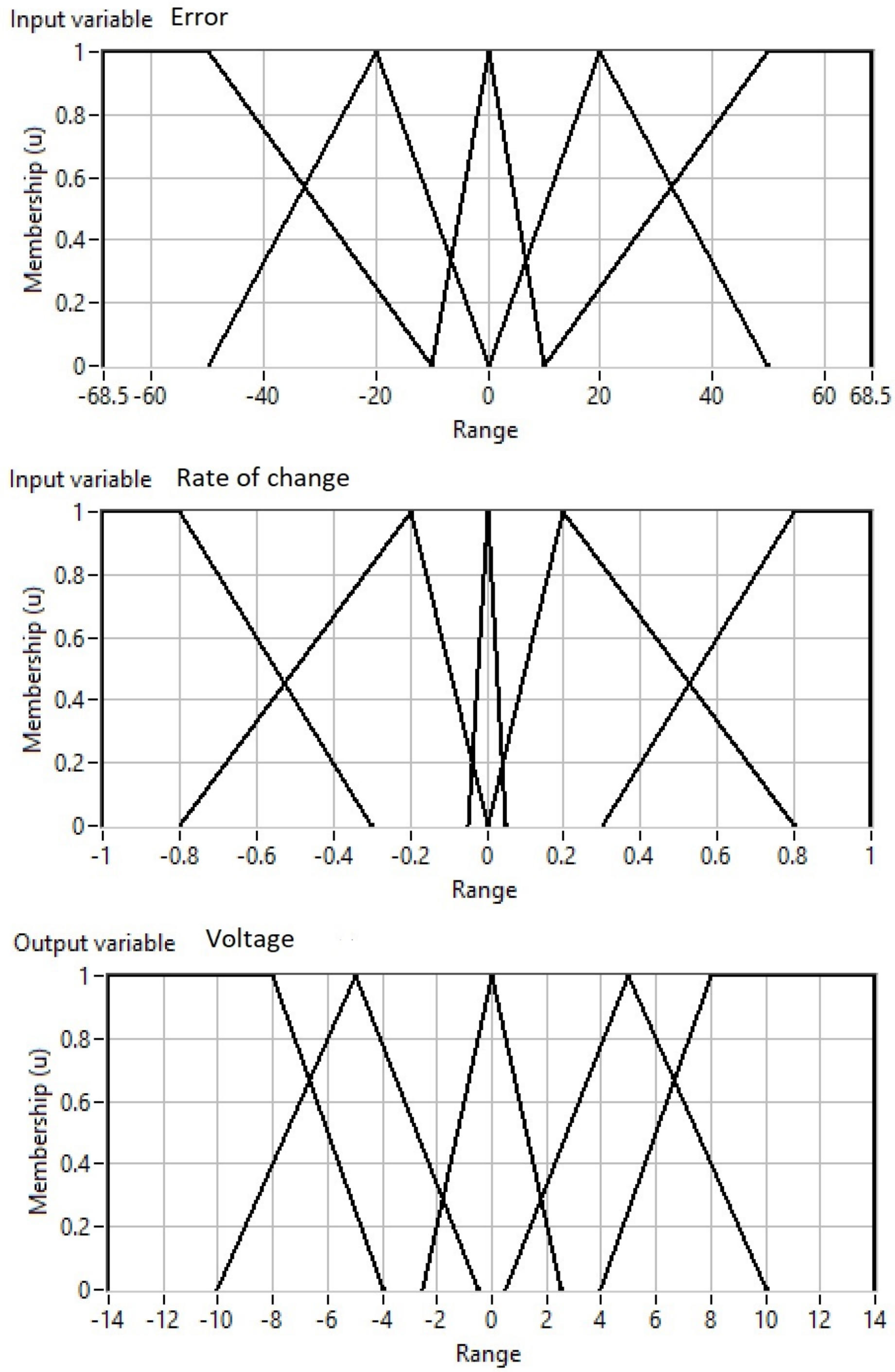


Figure F.1: *2 Input fuzzy controller membership functions for Joint 1, top is for joint error, middle is for rate of change of joint error and bottom is output membership function for CFP voltage*

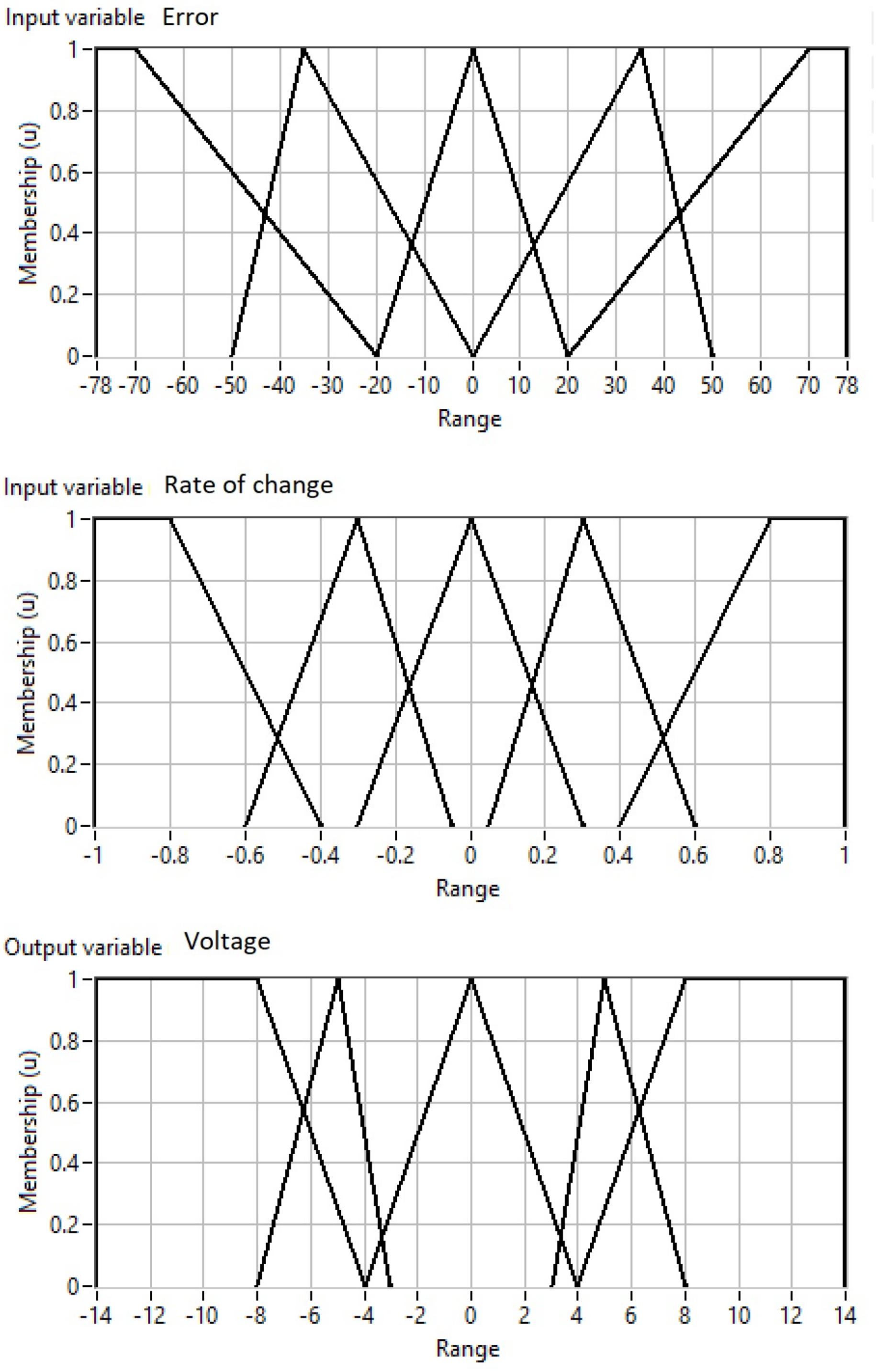


Figure F.2: *2 Input fuzzy controller membership functions for Joint 2, top is for joint error, middle is for rate of change of joint error and bottom is output membership function for CFP voltage*

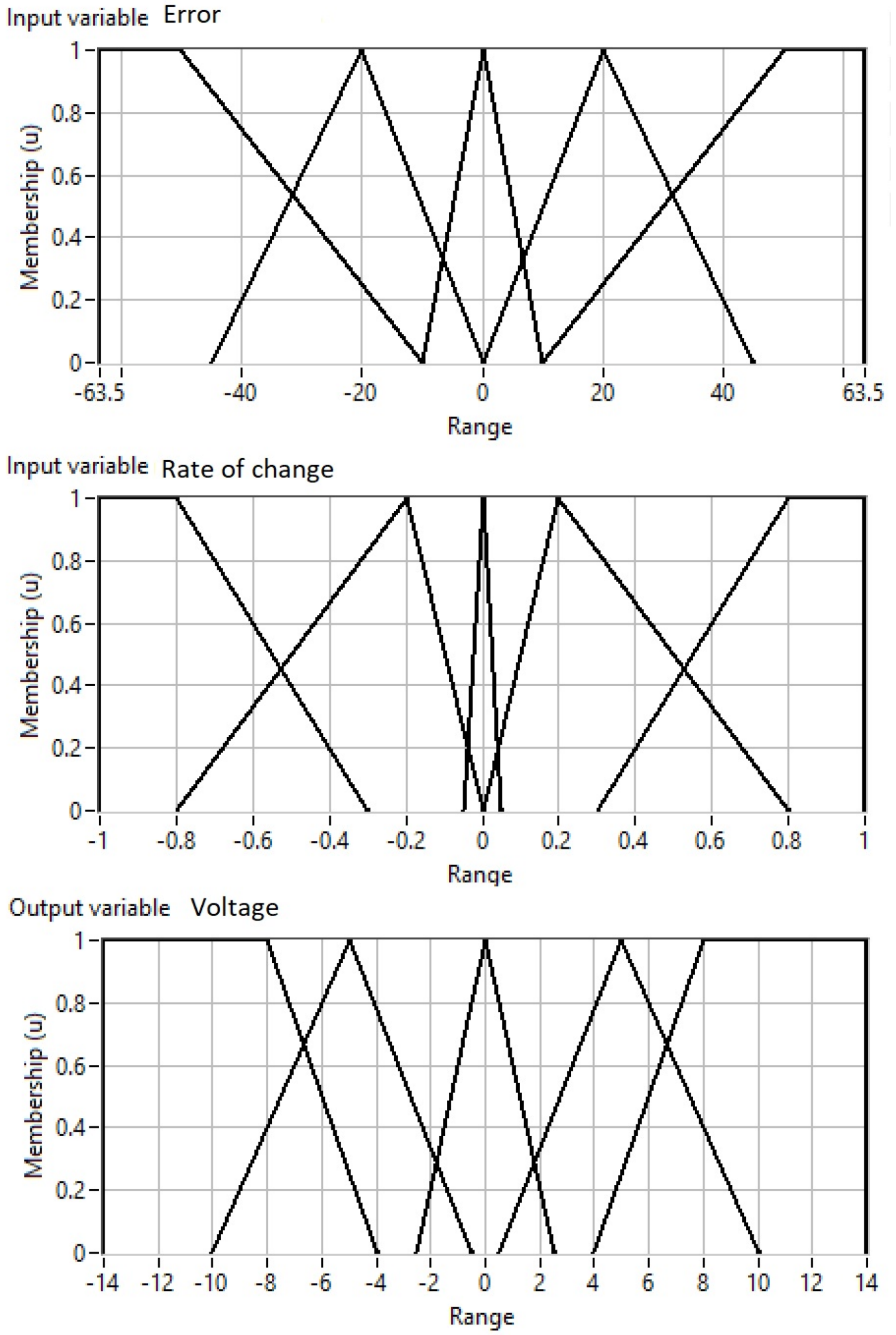


Figure F.3: *2 Input fuzzy controller membership functions for Joint 3, top is for joint error, middle is for rate of change of joint error and bottom is output membership function for CFP voltage*

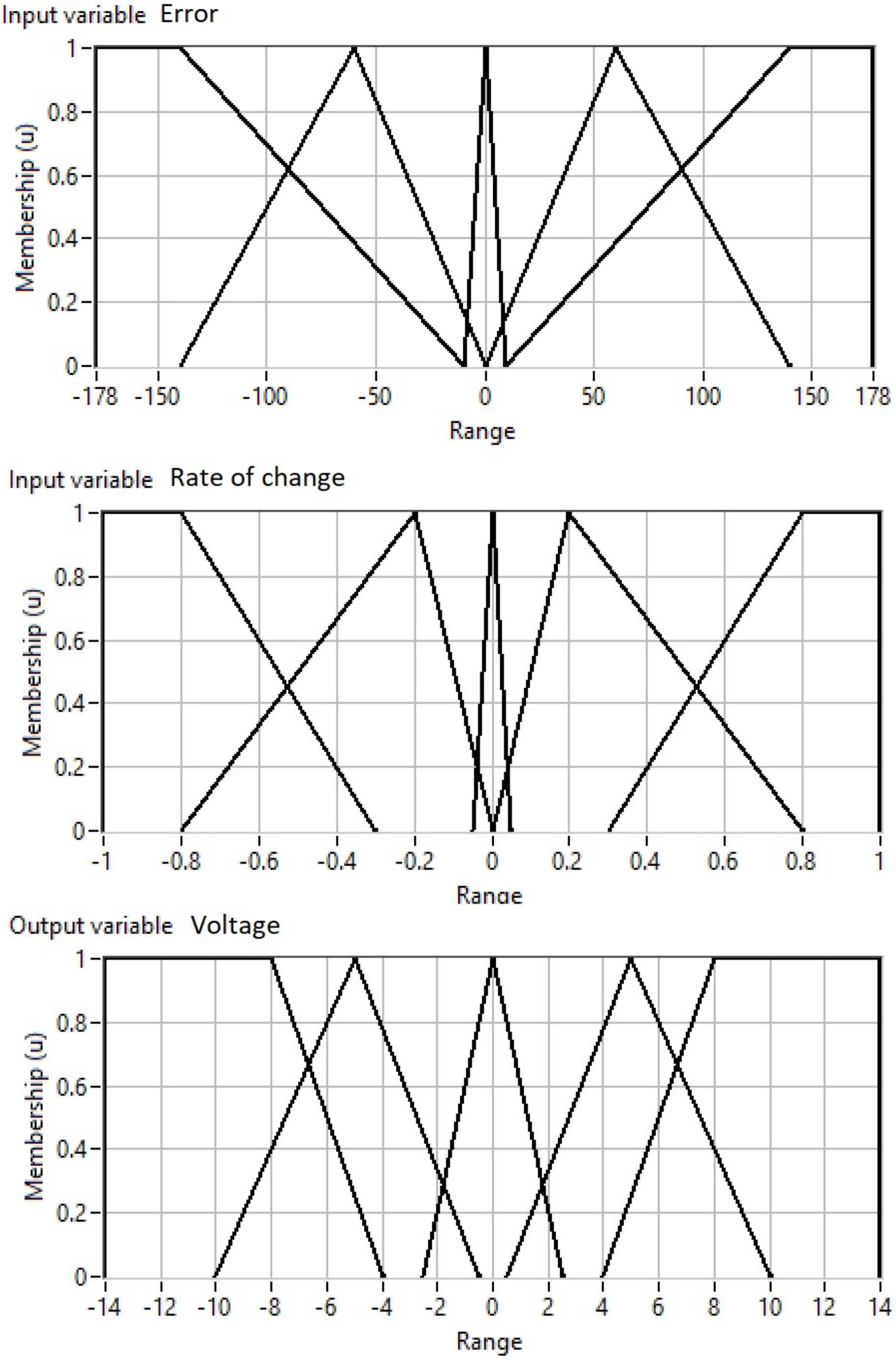


Figure F.4: *2 Input fuzzy controller membership functions for Joint 4, top is for joint error, middle is for rate of change of joint error and bottom is output membership function for CFP voltage*

Appendix G

Controller Comparison

G.1 Simulation results

Table G.1: *Table of results comparing controllers, for a sine trajectory motion, all values are from simulation results, best results are highlighted (table 1 of 2).*

		PIP IDZ (J4 Exponential)			PIP IDZ (J4 linear)		
		a	**b**	**c**	**a**	**b**	**c**
Avg. Euclidian Norm		161.133	**77.242**	189.242	139.229	79.9545	191.3796
MSE set point following	**J1**	37.903	5.044	42.84	37.903	5.044	42.84
	J2	6.018	3.362	13.136	6.018	3.362	13.136
	J3	65.992	9.929	65.992	65.992	9.929	65.992
	J4	39.209	**32.698**	166.593	57.320	49.866	193.897
Variance in end effector position error	**X**	2.558E+03	**6.890E+02**	3.699E+03	1.754E+03	7.778E+02	3.829E+03
	Y	1.198E+04	3.260E+03	1.657E+03	5.381E+03	3.227E+03	**1.655E+03**
	Z	1.451E+04	**3.003E+03**	2.081E+04	1.521E+04	3.569E+03	2.178E+04
Variance in controller output	**J1**	0.098	0.106	0.095	0.098	0.106	0.095
	J2	0.024	0.023	**0.018**	0.024	0.023	**0.018**
	J3	0.112	0.126	0.089	0.112	0.126	0.089
	J4	0.407	0.415	0.326	0.631	0.518	0.301
Variance in CFP input	**J1**	1.532	1.560	1.527	1.532	1.560	1.527
	J2	1.752	1.737	**1.559**	1.752	1.737	**1.559**
	J3	1.889	1.945	1.831	1.889	1.945	1.831
	J4	2.007	2.017	1.963	1.668	1.644	**1.521**
Var of Diff of CFP	**J1**	**0.0276**	0.0322	**0.0276**	**0.0276**	0.0322	**0.0276**
	J2	0.1091	0.0658	0.4490	0.1091	0.0658	0.4490
	J3	**0.0361**	0.0444	**0.0361**	**0.0361**	0.0444	**0.0361**
	J4	0.0393	0.0391	0.0392	1.0818	0.4363	0.0611

Table G.2: *Table of results comparing controllers, for a sine trajectory motion, all values are from simulation results, best results are highlighted (table 2 of 2).*

		PIP IDZ FP			PIP scaled	PIP No	PI scaled
		a	b	c	input	scaling	input
Avg. Euclidian Norm		163.553	78.054	276.28	181.0945	201.894	142.0093
MSE set point following	**J1**	31.848	**4.921**	133.436	39.1107	32.122	24.5504
	J2	5.495	**2.751**	12.141	15.1334	14.508	3.573
	J3	54.567	**7.057**	158.522	62.4317	30.947	47.08
	J4	190.825	68.960	246.93	97.8049	337.738	161.408
Variance in end effector position error	**X**	3.202E+03	7.986E+02	6.955E+03	3.53E+03	3.588E+03	2.882E+03
	Y	9.930E+03	2.431E+03	2.556E+04	1.74E+04	7.350E+03	6.905E+03
	Z	1.776E+04	4.113E+03	5.247E+04	1.67E+04	2.384E+04	1.397E+04
Variance in controller output	**J1**	0.111	0.110	**0.082**	7.553		10.728
	J2	0.026	0.026	0.200	51.645		63.204
	J3	0.135	0.139	**0.061**	160.960		351.903
	J4	0.303	0.355	**0.225**	160.312		142.929
Variance in CFP input	**J1**	1.589	1.566	**1.510**	1.773	47.027	1.795
	J2	1.704	1.684	1.647	1.941	44.187	1.964
	J3	2.051	1.998	**1.751**	2.060	30.576	2.229
	J4	1.617	1.680	1.561	2.002	52.986	1.992
Var of Diff of CFP	**J1**	**0.0276**	0.0277	**0.0276**	0.560	21.3202	0.5144
	J2	0.0409	**0.0410**	0.4090	0.073	25.3976	0.0729
	J3	0.0362	**0.0361**	0.0394	0.262	5.9991	0.0858
	J4	**0.0371**	**0.0371**	**0.0371**	0.074	8.6443	0.0853

Table G.3: *Table of results comparing controllers, for a step input, all values are from simulation results, best results are highlighted (table 1 of 2).*

		PIP IDZ (J4 Exponential)			PIP IDZ (J4 linear)		
		a	b	c	a	b	c
Avg. Euclidian Norm		48.3673	**43.8956**	50.1145	50.9052	46.2566	55.6647
MSE set point following	**J1**	34.3007	**23.8715**	34.9775	34.3007	**23.8715**	34.9775
	J2	28.205	**26.122**	35.3721	28.205	**26.122**	35.3721
	J3	33.2164	32.7161	35.608	33.2164	32.7161	35.608
	J4	**25.8217**	26.553	41.215	30.0981	31.2478	45.4706
Variance in end effector position error	**X**	5.093E+03	4.586E+03	6.108E+03	5.407E+03	**4.522E+03**	6.098E+03
	Y	2.432E+03	2.385E+03	2.618E+03	2.476E+03	2.407E+03	2.618E+03
	Z	1.261E+04	9.302E+03	1.472E+04	1.333E+04	9.518E+03	1.492E+04
Variance in controller output	**J1**	0.104	2.940	0.092	0.104	2.940	0.092
	J2	1.960	0.716	0.076	1.960	0.716	0.076
	J3	0.388	5.339	0.080	0.388	5.339	0.080
	J4	0.569	0.753	0.081	0.471	**0.428**	0.088
Variance in CFP input	**J1**	0.420	1.399	0.358	0.420	1.399	**0.358**
	J2	2.416	1.130	0.515	2.416	1.130	0.515
	J3	0.982	6.523	**0.494**	0.982	6.523	**0.494**
	J4	0.661	0.822	0.439	1.274	1.317	0.980
Rise time	**J1**	1.0186	**0.5895**	1.0882	1.0186	**0.5895**	1.0882
	J2	**0.6225**	0.6630	1.2117	**0.6225**	0.6630	1.2117
	J3	0.9858	**0.9541**	1.1930	0.9858	**0.9541**	1.1930
	J4	**0.5014**	0.4763	1.2085	0.6084	0.5797	1.2395
Overshoot %	**J1**	6.0201	35.3330	2.0067	6.0201	35.3330	2.0067
	J2	43.6242	33.8870	0.0000	43.6242	33.8870	0.0000
	J3	27.6667	40.6780	0.0000	27.6667	40.6780	0.0000
	J4	29.0000	37.7926	0.0000	8.0000	13.0000	8.0000
Var of Diff of CFP	**J1**	0.0621	0.2940	0.0464	0.0621	0.2940	0.0464
	J2	0.6569	0.2195	0.0700	0.6569	0.2195	0.0700
	J3	0.0984	0.6206	0.0453	0.0984	0.6206	0.0453
	J4	0.2545	0.2100	0.0524	3.1635	1.1910	0.2351

Table G.4: *Table of results comparing controllers, for a step input, all values are from simulation results, best results are highlighted (table 2 of 2).*

		PIP IDZ FP			PIP scaled input	PIP No scaling	PI scaled input
		a	b	c			
Avg. Euclidian Norm		61.5741	46.6424	81.9943	60.845	169.0093	67.6596
MSE set point following	**J1**	38.3264	24.8676	61.867	36.7873	53.5855	34.3536
	J2	32.0269	30.2628	38.2212	37.9684	36.44	26.5827
	J3	39.1601	**29.8726**	59.458	39.5158	31.2386	49.8038
	J4	49.9026	36.679	54.2145	34.3233	186.5994	40.166
Variance in end effector position error	**X**	6.935E+03	4.594E+03	1.030E+04	6.25E+03	5.927E+03	6.68E+03
	Y	2.268E+03	2.490E+03	**2.227E+03**	3.07E+03	4.145E+03	2.76E+03
	Z	**1.633E+03**	1.075E+04	2.369E+04	1.44E+04	1.584E+04	1.39E+04
Variance in controller output	**J1**	0.099	0.466	**0.032**	29.095		61.619
	J2	0.156	0.229	**0.057**	142.708		688.258
	J3	0.078	0.516	**0.021**	217.109		425.217
	J4	0.073	0.181	0.046	100.872		66.030
Variance in CFP input	**J1**	0.492	0.562	0.493	1.433	37.635	1.495
	J2	0.663	0.736	**0.458**	0.919	36.465	1.202
	J3	0.605	1.131	0.541	1.570	27.662	1.649
	J4	**0.350**	0.376	**0.350**	0.798	45.223	0.809
Rise time	**J1**	1.061	0.6673	2.2833	1.193	0.5868	0.9886
	J2	0.9055	0.8082	1.5906	1.231	0.617	0.7076
	J3	1.2046	0.9773	2.1401	1.288	0.9695	1.2180
	J4	1.4380	0.6405	2.0375	0.890	0.4238	1.1239
Overshoot %	**J1**	14.527	15.0502	**0**	13.667	N/A	27.8000
	J2	11.6667	14.3333	**0**	0.000	23.3333	15.0000
	J3	12.0805	13.4680	0	21.000	13.333	38.3333
	J4	3.6667	12.3333	**1.333**	23.667	N/A	13.3333
Var of Diff of CFP	**J1**	0.0224	0.0611	**0.0135**	0.6747	17.6857	0.6109
	J2	0.0507	0.0476	**0.0181**	0.0329	16.8577	0.2353
	J3	0.0234	0.0797	**0.0136**	0.7124	5.5776	0.3703
	J4	**0.0192**	0.0237	0.0236	0.0184	7.438	0.0194

Table G.5: *Table of results comparing controllers, for a grasp input, all values are from simulation results, best results are highlighted (table 1 of 2).*

		PIP IDZ (J4 linear)			PIP IDZ FP		
		a	b	c	a	b	c
Avg. Euclidian Norm		84.0841	**70.6377**	**70.6377**	103.355	81.215	157.3374
MSE set point following	**J1**	58.2446	**37.996**	60.5574	66.2976	41.1705	104.1074
	J2	**18.179**	17.4484	27.0883	24.3286	21.2636	30.9257
	J3	3.6668	**2.9273**	5.4478	5.7971	4.0121	9.4664
	J4	**72.8307**	84.3715	84.3715	133.0852	91.889	136.6066
Variance in end effector position error	**X**	**8.652E+02**	1.163E+03	1.163E+03	1.497E+03	1.278E+03	1.182E+03
	Y	**2.488E+03**	2.663E+03	2.663E+03	3.024E+03	3.287E+03	2.825E+03
	Z	1.754E+04	**1.533E+04**	**1.533E+03**	2.229E+04	1.592E+04	2.119E+04
Variance in controller output	**J1**	0.099	2.726	0.079	0.117	0.648	**0.014**
	J2	0.162	0.260	0.032	0.098	0.136	**0.031**
	J3	0.013	0.099	0.005	0.010	0.060	**0.002**
	J4	0.858	1.037	1.037	0.135	0.450	**0.113**
Variance in CFP input	**J1**	0.571	2.463	0.428	1.421	1.563	**0.122**
	J2	0.816	0.848	0.500	0.961	0.979	**0.499**
	J3	0.570	1.278	0.541	0.725	0.901	**0.371**
	J4	1.638	1.672	1.672	1.174	1.260	**1.168**
Rise time	**J1**	0.9992	**0.4427**	1.1025	0.9711	0.5319	2.1640
	J2	0.5560	**0.4767**	1.3200	0.7165	0.6087	1.3817
	J3	1.0500	**0.2856**	1.5506	0.7127	0.3425	1.6713
	J4	0.9328	**0.5765**	1.4459	1.6956	0.6404	1.9332
Overshoot %	**J1**	1.5561	47.4506	**0.0000**	14.5154	19.2007	**0.0000**
	J2	11.1765	18.3432	**0.0000**	15.9763	20.1183	**0.0000**
	J3	3.2258	25.1613	**0.0000**	20.0000	39.3548	**0.0000**
	J4	6.2823	12.3064	5.4217	4.9914	13.1670	**1.9793**
Var of Diff of CFP	**J1**	0.0648	0.5928	0.0706	0.0668	0.2044	**0.0200**
	J2	0.1883	0.1781	0.1428	0.0931	0.0944	0.0389
	J3	0.1293	0.4398	0.107	0.0567	0.1586	**0.0253**
	J4	2.4871	0.7198	0.1947	0.0487	0.1090	**0.0487**

Table G.6: *Table of results comparing controllers, for a grasp input, all values are from simulation results, best results are highlighted (table 2 of 2).*

		PIP scaled input	PIP No scaling	PI scaled input
Avg. Euclidian Norm		99.2322	239.2035	99.2322
MSE set point following	**J1**	61.7398	109.4113	61.7398
	J2	27.5129	21.6784	27.5129
	J3	5.7595	12.6538	5.7595
	J4	91.1732	373.6308	91.1732
Variance in end effector position error	**X**	1.268E+03	8.954E+02	1.268E+03
	Y	2.820E+03	3.779E+03	2.820E+03
	Z	2.054E+04	1.825E+04	2.054E+04
Variance in controller output	**J1**	25.533		25.533
	J2	42.343		42.343
	J3	15.146		15.146
	J4	159.914		159.914
Variance in CFP input	**J1**	1.743	42.447	1.743
	J2	0.006	49.113	0.006
	J3	1.631	31.232	1.631
	J4	1.494	55.191	1.494
Rise time	**J1**	1.7106	N/A	1.7106
	J2	1.4104	1.0749	1.4104
	J3	1.3521	1.0435	1.3521
	J4	0.8869	N/A	0.8869
Overshoot %	**J1**	6.8443	N/A	6.8443
	J2	**0.0000**	40.8284	**0.0000**
	J3	9.6774	49.6774	9.6774
	J4	16.6093	N/A	16.6093
Var of Diff of CFP	**J1**	0.6931	17.9492	0.6931
	J2	**1.03E-04**	24.9192	**1.03E-04**
	J3	0.8489	6.823	0.8489
	J4	0.1052	9.2491	0.1052

G.2 Experimental results

Table G.7: *Table of results comparing controllers, for a sine trajectory motion, all values the average of three experimental runs, best results are highlighted (table 1 of 2).*

		PIP IDZ (J4 Exponential)			**PIP IDZ (J4 linear)**		
		a	**b**	**c**	**a**	**b**	**c**
MSE set point following	**J1**	34.3284	17.8304	38.5135	34.3284	17.8304	38.5135
	J2	4.5741	3.4513	**1.1633**	4.5741	3.4513	**1.1633**
	J3	29.4723	12.0346	58.2137	29.4723	12.0346	58.2137
	J4	58.5378	59.0855	126.4875	68.634	**56.7307**	122.816
Variance in end effector position error	**X**	2.545E+03	2.026E+03	3.031E+03	2.649E+03	2.290E+03	3.008E+03
	Y	1.098E+04	8.147E+03	1.049E+04	1.094E+04	8.939E+03	1.047E+04
	Z	2.292E+04	1.811E+04	2.513E+04	2.291E+04	1.857E+04	2.502E+04
Variance in controller output	**J1**	0.0891	8.9547	**0.0883**	0.0891	8.9547	**0.0883**
	J2	0.0364	0.0696	0.0303	0.0364	0.0696	0.0303
	J3	0.0873	1.5335	0.0662	0.0873	1.5335	0.0662
	J4	14.4371	18.7151	0.4005	3.667	1.636	**0.211**
Variance in CFP input	**J1**	1.5110	2.4821	**1.3552**	1.5110	2.4821	1.3552
	J2	1.1061	**0.8023**	1.1709	1.1061	**0.8023**	1.1709
	J3	1.7426	2.7910	**1.6805**	1.7426	2.7910	**1.6805**
	J4	10.6526	14.3366	1.8956	1.941	**0.930**	1.482
Var of Diff of CFP	**J1**	0.0277	0.384	0.0555	0.0277	0.384	0.0555
	J2	0.255	0.2709	0.1187	0.255	0.2709	0.1187
	J3	0.0747	0.4914	0.0534	0.0747	0.4914	0.0534
	J4	0.7721	1.0368	0.3305	0.4347	0.3043	0.1113

Table G.8: *Table of results comparing controllers, for a sine trajectory motion, all values the average of three experimental runs, best results are highlighted (table 2 of 2).*

		PIP IDZ FP			PIP scaled	PI scaled
		a	b	c	input	input
Avg. Euclidian Norm		170.8278	**77.2116**	276.4220	189.9105	159.9839
MSE set point following	**J1**	40.2781	**5.6797**	147.0428	56.8033	20.5543
	J2	3.9828	1.934	10.5186	12.7770	8.0446
	J3	44.4468	**5.75**	157.5334	64.0240	81.9050
	J4	311.1693	85.443	287.5734	77.0698	144.7568
Variance in end effector position error	**X**	2.856E+03	**7.345E+02**	6.761E+03	4.089E+04	4.465E+03
	Y	7.643E+03	**1.756E+03**	2.375E+04	1.581E+04	1.315E+04
	Z	2.446E+04	**4.846E+03**	5.386E+05	2.059E+04	1.115E+04
Variance in controller output	**J1**	0.1376	0.1239	0.0905	216.1118	142.0624
	J2	0.0241	0.0235	**0.0174**	36.5052	62.3518
	J3	0.1079	0.1114	**0.0579**	368.5113	981.0950
	J4	0.3650	0.4482	0.2792	120.1757	93.0226
Variance in CFP input	**J1**	1.5806	1.5725	1.5109	1.6920	1.5886
	J2	1.6206	1.5898	1.5525	1.5472	1.9091
	J3	1.9345	1.8083	1.7135	1.9264	2.2678
	J4	1.5947	1.6277	1.5431	1.2303	1.3627
Var of Diff of CFP	**J1**	0.0148	0.0213	**0.0131**	0.0263	0.0301
	J2	0.0255	0.0263	**0.0249**	0.1110	0.0540
	J3	0.202	0.2660	**0.0308**	0.0448	0.0527
	J4	0.0236	0.2250	**0.0188**	0.1922	0.1703

Table G.9: *Table of results comparing controllers, for a Step trajectory motion, all values the average of three experimental runs, best results are highlighted (table 1 of 2).*

		PIP IDZ (J4 exponential)			PIP IDZ (J4 linear)		
		a	b	c	a	b	c
Avg. Euclidian Norm		62.8993	73.2533	56.2282	63.3896	84.3869	**54.52**
MSE set point following	**J1**	35.5313	36.3194	32.9924	35.5313	36.3194	32.9924
	J2	26.9576	28.013	33.7779	26.9576	28.013	33.7779
	J3	31.6775	33.5378	32.8307	31.6775	33.5378	32.8307
	J4	51.7806	75.4331	**33.581**	46.7669	35.4607	41.5915
Variance in end effector position error	**X**	5.608E+03	5.805E+03	5.389E+03	5.403E+03	5.117E+03	5.510E+03
	Y	2.225E+03	2.295E+03	2.592E+03	2.232E+03	2.299E+03	2.591E+03
	Z	1.448E+04	1.188E+04	1.301E+04	1.432E+04	1.325E+04	1.423E+04
Variance in controller output	**J1**	0.2028	10.0784	0.1202	0.2028	10.0784	0.1202
	J2	1.0056	2.4066	0.0949	1.0056	2.4066	0.0949
	J3	1.0374	8.1080	0.0822	1.0374	8.1080	0.0822
	J4	14.0612	30.9980	0.1847	2.5534	0.8619	0.2258
Variance in CFP input	**J1**	0.6797	2.8928	**0.4395**	0.6797	2.8928	**0.4395**
	J2	1.3796	2.5515	**0.5491**	1.3796	2.5515	**0.5491**
	J3	1.7898	7.8880	**0.5405**	1.7898	7.8880	**0.5405**
	J4	10.3017	22.9496	0.8279	1.3208	0.4821	1.1721
Rise time	**J1**	0.9291	0.7681	0.8212	0.9291	0.7681	0.8212
	J2	0.7850	**0.7338**	1.1889	0.7850	**0.7338**	1.1889
	J3	1.1155	**1.0210**	1.1439	1.1155	**1.0210**	1.1439
	J4	N/A	N/A	N/A	N/A	1.0809	2.0501
Overshoot %	**J1**	19.7611	27.3311	**0.6356**	19.7611	27.3311	**0.6356**
	J2	28.1122	44.2044	0.5689	28.1122	44.2044	0.5689
	J3	37.2033	38.9011	**0.4711**	37.2033	38.9011	**0.4711**
	J4	29.5056	30.9311	13.2311	0.1011	2.6433	7.5322
Var of Diff of CFP	**J1**	0.0747	0.3133	0.0944	0.0747	0.3133	0.0944
	J2	0.1934	0.4041	0.0705	0.1934	0.4041	0.0705
	J3	0.1642	0.6214	0.0757	0.1642	0.6214	0.0757
	J4	1.1106	2.3445	0.2228	0.3775	0.1923	0.2019

Table G.10: *Table of results comparing controllers, for a Step trajectory motion, all values the average of three experimental runs, best results are highlighted (table 2 of 2).*

		PIP IDZ FP			PIP scaled input	PI scaled input
		a	b	c		
Avg. Euclidian Norm		62.4097	75.6687	80.5895	159.9839	81.2915
MSE set point following	**J1**	36.4724	**23.8106**	53.0819	20.5543	46.1355
	J2	29.4451	**26.6332**	36.7904	8.0446	29.2337
	J3	36.2251	**27.7133**	53.4537	81.9050	52.7150
	J4	55.4766	107.066	54.851	144.7568	52.1419
Variance in end effector position error	**X**	6.643E+03	4.646E+03	9.429E+03	**4.465E+03**	8.887E+03
	Y	**1.976E+03**	2.125E+03	2.228E+03	1.315E+04	2.942E+03
	Z	1.553E+04	**1.169E+04**	2.121E+04	1.115E+04	1.799E+04
Variance in controller output	**J1**	0.0899	0.4345	**0.0260**	142.0624	152.8878
	J2	0.1409	0.2422	**0.0545**	62.3518	699.9753
	J3	0.0715	0.4716	**0.0179**	981.0950	487.8159
	J4	0.0580	4.4572	**0.0507**	93.0226	79.7280
Variance in CFP input	**J1**	1.0714	0.9189	1.1529	1.5886	0.9941
	J2	1.2906	1.4388	1.3186	1.9091	1.5206
	J3	1.0827	1.4711	1.0784	2.2678	1.1935
	J4	0.3160	1.9452	**0.3149**	1.3627	0.6518
Rise time	**J1**	0.8657	**0.6296**	1.6139	0.9876	1.0054
	J2	1.0173	0.7915	1.7033	1.1408	0.8397
	J3	1.2061	1.0048	1.9318	1.3857	1.2456
	J4	1.4026	N/A	1.9380	**0.7896**	0.9924
Overshoot %	**J1**	28.7078	22.4489	17.3600	25.4067	32.2144
	J2	6.8267	12.6156	2.2067	**0.3356**	16.6500
	J3	13.8122	11.0167	8.9978	19.3167	39.5878
	J4	0.8200	41.8156	**0.0000**	6.4144	20.1467
Var of Diff of CFP	**J1**	0.0849	0.2629	**0.0616**	0.0301	0.1799
	J2	0.1156	0.1722	0.1595	**0.0540**	0.2325
	J3	0.1038	0.3629	0.0750	**0.0527**	0.1093
	J4	0.0089	0.1467	**0.0104**	0.1703	0.0766

Table G.11: *Table of results comparing controllers, for a grasp motion, all values the average of three experimental runs, best results are highlighted.*

		PIP IDZ (J4 linear)			PIP IDZ FP			PIP scaled input	PI scaled input
		a	b	c	a	b	c		
Avg. Euclidian Norm		149.0674	121.7317	160.4528	**83.9217**	141.1886	176.6187	190.0765	155.2212
MSE set point following	**J1**	89.9063	49.674	106.2193	**32.0371**	73.894	112.6802	141.1302	96.1738
	J2	41.4319	12.7876	47.4726	**9.0921**	34.6106	26.3356	50.0618	21.7208
	J3	6.0644	9.0059	8.9069	**4.0072**	6.2995	7.0299	11.5533	10.3052
	J4	111.7112	83.389	163.3164	**59.9262**	212.2146	111.9092	141.0443	127.1288
Variance in end effector position error	**X**	1.809E+03	2.543E+03	1.177E+03	9.941E+02	3.035E+03	**5.895E+02**	1.280E+03	3.3240E+04
	Y	6.524E+03	2.115E+03	1.746E+03	**1.010E+03**	4.749E+03	1.597E+03	3.157E+03	2.7000E+03
	Z	2.390E+04	1.549E+04	2.304E+04	**1.106E+04**	2.916E+04	1.256E+04	2.685E+04	2.2652E+03
Variance in controller output	**J1**	0.3360	22.1536	0.1727	0.0605	1.0581	**0.0184**	80.5661	220.3481
	J2	22.7602	0.9046	0.0838	0.0415	0.2517	**0.0253**	80.7377	620.3131
	J3	0.0879	2.9967	0.0146	0.0039	0.0905	**0.0026**	44.3543	51.0302
	J4	28.0820	32.1292	0.8104	**0.0574**	0.2268	0.1150	467.4274	113.4925
Variance in CFP input	**J1**	1.4291	7.0269	0.4242	0.6507	2.0442	0.3789	**0.0345**	1.4591
	J2	14.5924	2.5972	0.4107	**0.3789**	1.7827	0.4515	0.6278	2.2624
	J3	1.7057	5.6487	0.8658	**0.2774**	1.2061	0.2693	1.2298	1.2451
	J4	5.4488	6.0081	1.7311	**0.4583**	1.7080	0.9133	1.8575	1.5361
Rise time	**J1**	0.7525	0.6684	0.9371	0.8251	**0.5994**	1.5805	1.0790	0.9692
	J2	0.6298	0.5933	1.5862	0.7634	0.5962	1.4384	1.7018	**0.4479**
	J3	0.4662	0.4037	1.8186	0.5345	**0.3801**	0.9677	0.8454	0.7724
	J4	0.5759	**0.5709**	1.8110	1.9217	1.9479	1.6803	0.6779	1.0262
Overshoot %	**J1**	8.6541	49.5942	**0.0000**	16.8979	14.9089	**0.0000**	14.2444	34.7160
	J2	62.2426	42.9349	8.0375	11.4615	22.4852	**3.1578**	8.4517	19.0414
	J3	13.9914	65.2860	0.6882	30.8559	39.0151	**0.0000**	2.9548	51.0839
	J4	35.1061	41.5476	**0.0000**	**0.0000**	**0.0000**	**0.0000**	1.7355	14.8680
Var of Diff of CFP	**J1**	0.1660	0.8264	0.1424	0.0402	0.1923	**0.0262**	0.0562	0.1372
	J2	1.0622	0.7432	0.1554	**0.0263**	0.1493	0.0559	0.4135	0.6853
	J3	0.5435	1.1056	0.3025	0.0226	0.1813	**0.0216**	0.5498	0.2830
	J4	0.6661	0.6673	0.3732	**0.0147**	0.0745	0.0457	0.5741	0.4213