

**POSITION AND FORCE CONTROL OF
A SERIAL ARM ROBOT**

Quanser CataLyst-5

**COMMANDE DE POSITION ET FORCE
D'UN ROBOT MANIPULATEUR À
ARCHITECTURE SÉRIE**

Quanser CataLyst-5

A Thesis Submitted to the Division of Graduate Studies
of the Royal Military College of Canada
by

Lance Edward Mooney, BEng
Lieutenant Commander

In Partial Fulfillment of the Requirements for the Degree of
Master of Applied Science in Mechanical Engineering

June, 2015

© This thesis may be used within the Department of National Defence
but copyright for open publication remains the property of the author.

This thesis is dedicated to my family who patiently supported me in completing this very challenging stage of my career.

Acknowledgements

I would like to thank both of my supervisors, Dr. Amor Jnifene and Dr. Karim Khayati for providing answers to many questions and, more importantly, insightful conversation that helped me to find the answers to countless more.

Many thanks to all of the technicians in the Mechanical and Aerospace Engineering Department who provided technical support, advice, tools and supplies as I set to work the CataLyst-5 for open architecture position and force control. Special thanks to Pierre Bisonette who manufactured the parts required to fit the force/torque sensor to the robot and to Dave Neumann who provided excellent advice and technical assistance when troubleshooting electrical problems.

Thanks to my fellow Post Grad student Major Tom Connerty who was always available to discuss ideas or vent frustration about various problems encountered.

A hearty thanks to the faculty of the Mechanical and Aerospace Engineering Department who provided support throughout the last year, with specific thanks to Dr. Billy Allan who provided me with a flexible schedule to complete this thesis.

Abstract

Controlling contact between the end-effector of a robot manipulator with its environment is a critical function for more advanced operations, whether working at intricate assembly tasks or in assisting and interacting with humans. That such controlled interactions will become increasingly important in robotics, drives extensive research in the field of simultaneous position and force control. The research conducted in this thesis aims to develop and implement such control schemes using a serial arm robot.

The kinematics and dynamics of a five degrees of freedom articulated manipulator, the CataLyst-5, were determined for use in designing various control systems to be applied both in simulation and experimentally. Three position controllers of various complexity were developed and compared, including a PD Independent Joint controller, a Joint Space Linearized controller and a Cartesian Space Linearized controller. The Cartesian space controller was used as a foundation to develop two position/force controllers, the first a Hybrid Position/Force controller and the second a Position Based Force controller.

The controllers were implemented to have the robot move along various shape trajectories on a horizontal plane, with the position controllers following the trajectories in free space and the force controllers maintaining contact with a chalkboard drawing surface and applying a specified force to draw the shape with chalk. All controllers were initially applied to a SimMechanics model of the serial arm robot to ensure stability and allow for initial gains tuning. Once proven stable, the controllers were implemented on the CataLyst-5 using Quanser's open-architecture control software.

During both simulation and experimentation, it was found that the highly complex Cartesian Space Linearized controller provided the most accurate trajectory following results, with a trade-off in lengthy simulation and compilation times. Despite successful simulation results, the Hybrid controller could not be configured to maintain contact with the drawing surface during experimentation, necessitating the creation of a Modified Hybrid controller.

The Position Based Force controller provided the best experimental results, maintaining contact with the drawing surface throughout the trajectories.

Résumé

La commande de contact entre l'effecteur d'un robot manipulateur et son environnement est une fonction essentielle pour des opérations avancées, que ce soit dans le travail des tâches d'assemblage complexes ou dans l'assistance et l'interaction avec les humains. Parce que les interactions contrôlées deviendront de plus en plus importantes dans le domaine de la robotique, des recherches approfondies sont conduites dans le domaine de commande simultanée de position et de force. La recherche conduite dans cette thèse vise à développer et mettre en œuvre des systèmes de commande d'un bras de robot sériel.

La cinématique et la dynamique d'un manipulateur CataLyst-5 articulé à cinq degrés de liberté ont été déterminées pour une utilisation dans la conception de divers systèmes de commande applicables à la fois en simulation et en expérimentation. Trois systèmes de commande de position de complexité variable ont été développés et comparés, y compris un contrôleur PD à joints indépendants, une commande linéarisée dans l'espace articulaire et une commande linéarisée de l'espace cartésien. Cette dernière a été utilisée comme une base pour développer deux compensateurs de position et force. Le premier est un contrôleur hybride de position/force et le deuxième est un contrôleur de force dépendant de la position.

Les systèmes de commande ont été mis en œuvre pour déplacer le robot suivant des trajectoires dans l'espace libre, puis selon différentes trajectoires sur un plan horizontal et pour maintenir le contact avec la surface d'un tableau de dessin et l'application d'une force spécifiée pour dessiner la forme avec la craie par le biais de la commande de force. Tous les compensateurs ont été d'abord appliqués à un modèle SimMechanics du robot sériel pour assurer la stabilité et ajuster les gains initiaux. Une fois assurée la stabilité, les systèmes de commande ont été mis en œuvre sur le robot CataLyst-5 en utilisant le logiciel de commande à architecture ouverte de Quanser.

Au cours des phases de simulation et d'expérimentation, il a été convenu que la "très complexe" commande linéarisée dans l'espace cartésien fournit les

résultats les plus précis en suivant les trajectoires, au coût de longues durées de simulation et compilation. Malgré les succès observés dans les résultats des simulations, la commande hybride ne peut pas être configurée pour maintenir le contact avec la surface de contact au cours de l'expérimentation, ce qui nécessite la modification de cette commande hybride. Le compensateur de force en fonction de la position a fourni de meilleurs résultats expérimentaux en maintenant le contact avec la surface de dessin tout au long des trajectoires.

Contents

Acknowledgements	iii
Abstract	iv
Résumé	vi
List of Tables	xi
List of Figures	xii
Nomenclature	xiv
1 Introduction	1
1.1 Motivation	1
1.2 Background	2
1.2.1 Implementation Considerations	4
1.3 Research Objectives	8
1.4 Thesis Organization	9
1.5 Conclusion	9
2 Literature Review	10
2.1 Foundational Force Control Methods	10
2.2 Force Control Categorization	13
2.3 Modifications and Variations on Hybrid Position/Force Control	14
2.3.1 Modern Control Mechanisms	15
2.3.2 Model Adaptation	16
2.3.3 Task Trajectory Generation	16
2.4 Conclusion	17
3 System Modelling	18
3.1 Development of Solution to Direct Kinematic Problem	18

3.1.1	Task Specific DKP	21
3.2	Solution to Inverse Kinematic Problem	23
3.2.1	Task Specific IKP	25
3.3	Development of the Dynamic Model	26
3.3.1	CataLyst-5 Robot Physical Parameter Definition	30
3.4	Model Development	32
3.4.1	Simulink Mathematical Model	32
3.4.2	SimMechanics Model	32
3.4.3	Model Comparison	33
3.5	Contact Modelling	39
3.6	Conclusion	41
4	Control System Development	42
4.1	Path and Trajectory Planning	42
4.1.1	Joint Space versus Cartesian Space Planning	43
4.1.2	Trajectory Generator using a Quintic Polynomial	43
4.1.3	Cartesian Space Trajectory Generator	44
4.1.4	Joint Space Trajectory Generator	47
4.1.5	Trajectory Generation of Various Shapes	48
4.2	PD Independent Joint Error Control	49
4.3	Joint Space Linearized Control	51
4.4	Gains Tuning	53
4.5	Cartesian Space Linearized Control	55
4.6	Hybrid Position/Force Control	57
4.6.1	Force Detection and Control Method Switching	59
4.7	Position Based Force Control	60
4.8	Conclusion	60
5	Simulation Results	62
5.1	Comparison of Position Controllers	63
5.2	Comparison of Force Controllers	68
5.3	Conclusion	72
6	Experimental Setup	73
6.1	CataLyst-5 Robot System	73
6.2	C500C Controller and Power Supply	75
6.3	Open Architecture Control	77
6.4	Quanser Real-Time Control (QuaRC) Software	78
6.5	Manipulator Contact Environment	80
6.6	Controller Implementation	83

6.7 Implementation Issues	85
6.8 Conclusion	86
7 Experimental Results	88
7.1 Comparison of Position Controllers	89
7.2 Comparison of Force Controllers	93
7.3 Conclusion	99
8 Conclusion	101
9 Recommendations for Future Work	104
Bibliography	106
Appendices	112
A Geometric Solution to the DKP and IKP	113
A.1 Solution to the DKP	113
A.2 Solution to the IKP	117
B Maple Equations	121
C Manufacturer's Data	129
D Circle and Spiral Trajectory Generators	137
E Joint Space Controllers - Gains Tuning	139
F Simulation Results - Spiral Trajectory	149
G Force/Torque Sensor	153
H Manufactured Parts	155
I Load Cell Information	158
J RMC CataLyst-5 Open Architecture Operating Procedures	160

List of Tables

3.1	DH Parameters and Joint Limitations	20
3.2	Position and Orientation Using Geometric Solution to DKP	22
4.1	Parameters Used to Calculate PD Independent Joint Control Gains	51
4.2	CataLyst-5 Maximum Joint Velocities	54
4.3	PD Independent Controller Gains, Maximum Joint Velocities & Errors	55
4.4	Joint Space Linear Controller Gains, Maximum Joint Velocities & Errors	55
6.1	CataLyst-5 System Parameters	74
7.1	Joint Space Controller Gains - CataLyst-5	89
7.2	Cartesian Space Controller Gains - CataLyst-5	91
7.3	Force Controller Gains - CataLyst-5	95
G.1	ATI 15/50 Force/Torque Sensor Parameters	153

List of Figures

1.1	Environmental Model Classification	7
3.1	DH Axes - Ready Position	19
3.2	IKP Solution Flow Chart	24
3.3	Location and Orientation of Manufacturer Axes	31
3.4	Impementation of the Simulink Model	33
3.5	Simulink Model - Subsystem Blocks	34
3.6	SimMechanics Model	35
3.7	Zero-Angle Position	36
3.8	Model Comparison - Free Response	37
3.9	Model Comparison - Constant Torque	38
3.10	SimMechanics Contact Model	40
4.1	Cartesian Space Trajectory Planner	45
4.2	Joint Space Trajectory Planner	47
4.3	Line Draw Trajectory Generator	49
4.4	PD Independent Joint Control - Block Diagram	50
4.5	Linearized Joint Space Control - Block Diagram	53
4.6	Cartesian Space Linearized Control - Block Diagram	57
4.7	Hybrid Position/Force Control - Block Diagram	58
4.8	Control Method Selection	60
4.9	Position Based Force Control	61
5.1	PD Independent Control - Star Trajectory	63
5.2	Joint Space Linearized Control - Star Trajectory	64
5.3	Cartesian Space Linearized Control - Star Trajectory	65
5.4	Trajectory Tracking Errors X & Y Axes - Star Trajectory	66
5.5	Control Torques - Star Trajectory	67
5.6	XYZ Control Signals - Star Trajectory	69
5.7	Error in Applied Force	71

5.8	Position Error - Force Controllers	72
6.1	Catalyst-5 Robot System	74
6.2	Catalyst-5 Robot Arm	75
6.3	Catalyst-5 Teach Pendant	76
6.4	C500D Open Architecture Controller	77
6.5	Immediate I/O HIL Blocks	79
6.6	Experimental Setup with Drawing Surface	80
6.7	Non-Inverting Operational Amplifier	81
6.8	Operational Amplifier Calibration	82
7.1	Joint Errors - Joint Space Controllers	90
7.2	Cartesian Space Linearized Controller Errors	92
7.3	Control Currents - Position Controllers	94
7.4	Control Signal - Hybrid Position/Force Controller	96
7.5	Control Signal - Force Controllers	97
7.6	Applied Force Error - Position/Force Controllers	98
7.7	Position Error - Position/Force Controllers	100
A.1	Planar Mechanism View for DKP/IKP Solution	114
A.2	Planar Robot Top View	115
E.1	Calibration Ready (Calrdy) Position	139
E.2	PD Ind. Control - Joint Errors for Joint 1 Step ($-\pi$ to π)	140
E.3	PD Lin. Control - Joint Errors for Joint 1 Step ($-\pi$ to π)	141
E.4	PD Ind. Control - Joint Errors for Joint 2 Step (0 to $\pi/2$)	142
E.5	PD Lin. Control - Joint Errors for Joint 2 Step (0 to $\pi/2$)	143
E.6	PD Ind. Control - Joint Errors for Joint 3 Step ($-\pi/2$ to 0)	144
E.7	PD Lin. Control - Joint Errors for Joint 3 Step ($-\pi/2$ to 0)	145
E.8	PD Ind. Control - Joint Errors for Joint 4 Step (0 to π)	146
E.9	PD Lin. Control - Joint Errors for Joint 4 Step (0 to π)	147
E.10	PD Ind. Control - Joint Errors for Joint 5 Step ($-\pi$ to π)	148
E.11	PD Lin. Control - Joint Errors for Joint 5 Step ($-\pi$ to π)	148
F.1	PD Independent Control - Spiral Trajectory	149
F.2	Joint Space Linearized Control - Spiral Trajectory	150
F.3	Cartesian Space Linearized Control - Spiral Trajectory	150
F.4	Trajectory Tracking Errors X & Y Axes - Spiral Trajectory	151
F.5	Control Torques - Spiral Trajectory	152
G.1	CataLyst-5 Gripper Configurations	154

Nomenclature

Roman Symbols

$[a_i]_i$	Vector from i th frame origin to $(i+1)$ th frame origin, using i th frame coordinates
$\vec{A}, \vec{B}, \vec{C}, \vec{D}$	Position vector of points coinciding with robot joint origins (as per Figure A.1)
a_i	Link Length: distance between Z_i and Z_{i+1} along X_{i+1} axis
b_i	Joint Offset: distance along Z_i where X_{i+1} intersects Z_i
\vec{e}_i	Unit vector directed along axis of actuation of i th joint
\mathcal{F}_i	i th reference frame number
F_{cell}	Force measured by the load cell mounted under the drawing surface
$\vec{g}(\boldsymbol{\theta})$	Vector of gravitational terms
$\vec{G}_X(\boldsymbol{\theta})$	Vector of gravitational terms in Cartesian space
H	Homogeneous transformation matrix
I_i	Inertia tensor of the i th link
i_i	Control current sent to the i th joint
J	Jacobian for the mechanism
$J_{\dot{c}_i}$	Jacobian of joint velocities to velocity of i th link center of mass
J_{ω_i}	Jacobian of joint velocities to angular velocity of i th link
J^*	Moore-Penrose pseudo-inverse of Jacobian

k_d	Derivative gain for various controllers
K_{enc}	Joint encoder ratios
K_g	Joint gear ratio
k_i	Integral gain for various controllers
k_p	Proportional gain for various controllers
K_t	Motor torque constant
\mathcal{L}	Lagrangian of a mechanical system
$M(\boldsymbol{\theta})$	Generalized inertia tensor
m_i	Mass of i th link
M_{dia}	Matrix containing only diagonal elements of generalized inertia tensor
$M_X(\boldsymbol{\theta})$	Generalized inertia tensor in Cartesian space
\vec{P}	Position vector of the end-effector reference frame in base reference frame coordinates
Q	Rotation matrix of the end-effector reference frame from the base reference frame
Q_{i+1}^i	Rotation matrix from i th frame to $(i+1)$ th frame
r	Roll orientation angle of end-effector
R	Radial position of the joint (A,B,C,D,P) reference frame - note $R^2 = X^2 + Y^2$
\vec{S}_{angle}	3×1 vector of angles representing orientation of end-effector as roll, pitch and yaw
s_i	Center of mass of i th link
S	Hybrid control position selection matrix
S'	Hybrid control force selection matrix
T	Total kinetic energy of a mechanical system
U	Measure mV output from load cell Wheatstone bridge

$\vec{v}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta})$	Vector of centrifugal and Coriolis terms
V	Total potential energy of a mechanical system
$\vec{V}_X(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta})$	Vector of centrifugal and Coriolis terms in Cartesian space
$\vec{\mathcal{W}}$	Vector of torques/forces applied to end-effector to affect desired movement
\vec{w}	Wrench (vector of torques and forces) applied by end-effector on environment
$X(t)$	Quintic polynomial representing 6 elements of Cartesian space position/orientation vector, or 5 elements of joint space position vector
X, Y, Z	Position of <i>ith</i> reference frame
X_i, Y_i, Z_i	Axes of <i>ith</i> reference frame

Greek Symbols

α_i	Link Twist: angle to bring Z_i parallel with Z_{i+1} rotated about X_{i+1} axis
β	Pitch orientation angle of end-effector
γ	Yaw orientation angle of end-effector
θ_i	Joint Angle: angle to bring X_i parallel with X_{i+1} rotated about the positive Z_i axis
$\vec{\tau}'$	Control input for the joint space linearized manipulator

Subscripts

i	<i>ith</i> element along kinematic chain (ie. link, joint, etc.)
-----	------------------------------------------------------------------

1 Introduction

In this chapter, the motivation behind conducting research into position and force control of robot manipulators is discussed. A brief background into the field of force control is provided. The objectives of the research conducted is presented and the organization of the thesis is explained.

1.1 Motivation

The subject of controlling the force¹ that a robot manipulator applies to its environment, while simultaneously controlling the end-effector position², is an important area of research in the field of robotics. That the predominant use of robot manipulators in industry involve position only control schemes significantly limits the range of applications to those that involve little to no interaction with the environment. As identified in [1], [2] and other sources, successful manipulator contact with the environment using purely position based control is only possible if the entirety of the task can be accurately planned. This would require a highly accurate kinematic and dynamic model of the manipulator, which is possible but very difficult, and an equally detailed model of the contact environment, which is extremely difficult [3]. As an example, the mating of mechanical parts using position control alone would require the relative locating of the parts to a degree of accuracy greater than the mechanical tolerance of the assembly [2]. If this level of position control precision cannot be achieved, contact forces build until joint actuators saturate or part breakage occurs.

The limitations of position only control of manipulators have led to significant research about incorporating information from additional sensors into the control of these robots. Research has been conducted into force, touch, distance and visual feedback in the field of robotics with the intention that these

¹The term force refers to both the force and torque applied by the end-effector

²The term position refers to both the position and orientation of the end-effector

will allow the robot to operate autonomously in unstructured environments [2], with the largest amount of research being in force sensing and control. Force control of manipulators can be applied in countless tasks including:

- manufacturing processes that require the application of specific forces (polishing, deburring, milling),
- managing the geometric uncertainties that are present with assembly tasks,
- moving or pushing external objects with a controlled force, and
- interacting with other manipulators, either in the transfer of material or grasping objects as would be done with a robot hand [4].

The application of such tasks will only become more prevalent in the future, both in making industrial tasks more efficient, but also in non-industrial related areas such as hospital care, home assistance for the elderly, maintenance of space hardware and even for personal entertainment purposes [5].

In [6], the authors identify slowing sales of manipulators in traditional applications, that primarily consist of positioning tasks within the automotive manufacturing industry. They identify that the future success of the robotics industry depends on growth outside of this traditional market, which will require the integration of sensor based technologies that are primarily vision and force based. The importance of additional inputs in manipulator control is further highlighted in [3] where sensing is recognized as a controls requirement as robots will never work in a perfectly structured world, where the dimensions of all parts are within tolerance, each part is precisely located and the task planning is without fault. As mentioned in [7], the full potential of manipulators can only be realized when they are used for contact tasks that require controlled dynamic interaction with the environment. How to simultaneously control both the position of and the force applied by a manipulator constrained by the environment is an area that has received extensive research and is the focus of this thesis.

1.2 Background

Force control in robotics (meaning robotic manipulators in this thesis) becomes a requirement when the manipulator experiences constrained motion, which is when some aspect of the environment sets constraints on the path that the robot can follow in Cartesian space [2]. [8] and [3] divide the control of a manipulator experiencing constrained motion into 3 states (or 2 states connected by transition):

- motion in free space,
- contact (or transition), and
- exertion of force.

The first state can be accomplished using a purely position based control scheme for which there are a number of industry proven methods. The second two states require some method that controls both the position of the end-effector and the force applied by the manipulator. Force control is always used in conjunction with a position controller [9] as a purely force based controller could only exist as a manipulator with its end-effector fixed to the environment (i.e. buried in concrete) [10], an impractical application. As such when force control is referred to throughout this document it can be assumed to mean position/force control unless otherwise indicated. What follows is a short summary of the topic including a brief history of the use of force control, a description of some of the definitions that are used interchangeably in the subject and some discussion on the various considerations that must be made when implementing force control. A thorough and well organized overview of the subject of force control of robot manipulators can be found in [11] for the reader who is seeking additional insight into the field.

Perhaps the earliest application of force feedback with mechanical manipulators was Goertz's work in the 1940s developing master-slave manipulators for use in radioactive hot lab work and in the early 50s implementing electro-servo manipulators with force reflection to the user [8]. While this may not be considered force control specifically it was an early application involving the monitoring of the interaction of a manipulator with its environment. Another early application implementing force reflection involved research into the development of force feedback for use in a prosthetic arm [12] that allowed the user to exert muscle effort to counter a sensed arm load. In the late 60s and early 70s, research began to look at how to replace the operator in these types of systems with some form of automatic or computer control [13]. The primary motivation behind this research was to allow for the earth based operation of a manipulator working in space to interact with, and move objects in its vicinity. This initiated years of research into the force control of manipulators, that will be summarized in the literature review that follows in the next chapter. Despite this considerable research, the first industrial application did not occur until the late 1990s when Ford installed a force controlled assembly cell on its production line [6].

Various terminology is used regularly and interchangeably in the field of force control. As a summary and to clarify what is used in this document, they are identified as follows. Joint versus Operational (or Cartesian) Space refers

to the workspace in which the desired and actual configurations of the manipulator are described and controlled. Joint space is an n -dimensional space, where n is the number of degrees of freedom of the manipulator. Operational space is (at maximum) a 6-dimensional space (also known as Cartesian space) that refers to the position and orientation of some portion of the manipulator, generally the end-effector, with respect to some reference frame, usually the base frame of the manipulator. The pose of the mechanism in joint space is related to the position of the end-effector in operational space via the solutions to the Direct and Inverse Kinematic Problems (DKP/IKP). The joint angular velocities are related to the end-effector linear/rotational velocities via the Jacobian of the mechanism. For most physical applications, the actual operational space position of the robot cannot be measured [14], instead it is determined using the DKP, making operational space based control less intuitive. Despite this, operational space control is used extensively in force control as it allows for easier inclusion of the values of force applied at the end-effector.

Generally manipulator control schemes can be described as either Independent or Multivariable [15], referred to as decentralized or centralized respectively in [14]. With independent joint control, each axis of the manipulator is controlled as a single input/single output system and the dynamic coupling effects due to motion of the other links are treated as disturbances. With multivariable control, these coupling effects, and the non-linear nature of robot manipulators are taken into account, allowing for more robust control with better stability and faster tracking of trajectories.

Force control schemes can be divided into 2 categories, Direct or Indirect, which are sometimes referred to as explicit or implicit. Direct force control measures the force applied by the manipulator with the aim of following an ordered value as closely as possible [16] [17]. Indirect force control regulates the force applied by the manipulator based on the joint angular positions, essentially the force control portion of the controller is in open loop [2].

1.2.1 Implementation Considerations

Additional sensing is a key element to manipulator force control. Aside from some compliance based control schemes, the majority of proposed force control schemes can be considered direct force. In general, sensors used in robotics can be divided into 2 groups; contact and non-contact (such as robot vision) [18]. While a few force control schemes incorporate robot vision to help identify the geometry of the environment or the proximity of the end-effector to the contact environment, contact sensing is used in virtually all. In addition to the

tasks that have already been mentioned, a number of other control problems can be aided using force sensor information including:

- the monitoring of task execution to ensure safe operation,
- the weighing of objects, and
- the selection of particular workpieces out of a bin [18].

Force sensors can be placed in various locations based on the type of operations the manipulator is to perform. A sensor mounted at the base of the robot could be used for collision protection, called guarded motion. An instrumented platform that contains the piece the manipulator is working on could be used to measure applied force. The most common application is a force/torque sensor installed between the manipulator's tool flange and the attached implement. With this installation, the sensor is subject to the inertia of both the end-effector and any items that are being held. The most sensitive application would be either individual force sensing fingers or perhaps a dedicated force probe [18]. A key consideration with the use of force sensors is that their output is generally quite noisy and taking the derivative of this signal may not provide useful information [4].

Another important consideration when using force sensors is that they are typically the most fragile element in what becomes a closed kinematic chain when the manipulator comes in contact with the environment. As such, significant research has been conducted in how to transition from free to constrained manipulator motion. This transition control is very challenging as even slow impact speeds lead to the deformation of both the end-effector and the environment, resulting in reaction forces that can make an otherwise stable system, unstable [19]. The resulting force control based joint actuation could cause the manipulator to leave the surface and start to bounce between constrained and free motion.

Three similar methods of controlling this transition were proposed in [20] and [21]. The simplest involves using a compliant covering on the end-effector or a soft force sensor, both of which have the disadvantage of being fixed while also limiting the precision of position control. The second method involves varying the controller gains, based on an estimate of the environmental impedance, to provide active damping during transition. This method does not work with stiff environments where the impact results in small amplitudes and high frequencies in the force feedback loop. The third method makes use of a position control loop that is faster than the force control loop, which allows the transient high impact forces to dissipate, but could result in sensor damage. Both [22] and [19] propose using a position controller with a separate position/force controller, then using a contact event driven switch to transi-

tion between the two. As would be expected, transitioning from constrained to free motion does not present a significant control problem [21].

While the detailed dynamic model of the Catalyst-5 is developed in Section 3.3, a number of aspects of the dynamics of manipulators both in free and constrained motion need to be considered. First, the type of joint actuation used affects the system dynamics. For existing industrial manipulators, joints are predominantly actuated via high reduction ratio gearboxes. These gear trains are used as a means both of using smaller actuators, and of placing the actuators away from the joint axis (usually closer to the base) thus reducing the inertia of the robot arm. The high gear ratio has a decoupling effect on the manipulator that reduces the effects of system nonlinearities, which means the changing dynamics experienced by one actuator due to the changing positions of the other joints can be treated as disturbances in basic independent joint PID control schemes. The trade-off with this type of joint actuation is increased friction, elasticity and backlash in the joints and thus less precise control of the manipulator can be achieved. The other type of joint actuation is direct drive where the actuator is coupled directly to the joint axis. This type of actuation while more precise to control requires more complicated control schemes due to the more significant effects of coupling and nonlinearities [14].

The stiffness of the entire system, including the manipulator, force sensor and environment, is a significant factor when considering force control and designing for an appropriate stiffness often involves compromise. For precise position control, a high overall manipulator stiffness is desirable. This same highly stiff system is at risk of becoming unstable when coming in contact with the environment. The actual stiffness of the system depends on a number of different factors. Firstly, the stiffness of the manipulator is a function of its construction, but also depends on its configuration [18]. In addition, the apparent stiffness of the manipulator itself can be controlled by adjusting the control system gains and a number of force control schemes make use of this fact [23]. The stiffness of the sensor is determined when manufactured, but is generally a function the level of force/torque it is designed to measure. The stiffness of the environment the manipulator is required to interact with, varies to the greatest degree and thus requires the greatest level of robustness from the controller [11]. A summary of the considerations regarding environmental stiffness can be found in Figure 1.1.

For a manipulator to be considered truly constrained, both the manipulator and the environment need to be considered infinitely rigid, which is not true for most problems [24]. Usually when the manipulator is in contact with the environment, there is at least some level of elastic deformation that

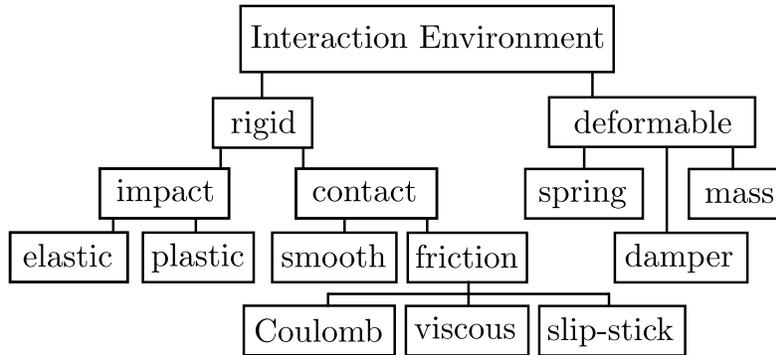


Figure 1.1: Environmental Model Classification (adapted from [11])

result in reaction forces and thus needs to be considered. This deformation can either be considered as occurring just with the environment (as with a machining task) or with both the end-effector and the environment, which is more likely the case. Elastic deformation also occurs within the manipulator itself, in both the links and the joints. The extent to which the links can deform depends upon both the physical structure, including the materials used, and the loading applied to the manipulator due to gravity, acceleration forces and contact with the environment. Generally the manipulator structure is such that the flexibility of the links is considerably less than that of the joints [25]. Elasticity in the joints can be attributed to any chains or belts that are used to transmit torque from separately located actuators or from the reduction gearing used in non-direct drive manipulators. The deformation of reduction gearing is usually more prevalent in harmonic drive reducers due to the inherent flexibility of their components [24]. The elasticity in the joints not only affects the manipulator dynamics, it can have a destabilizing effect on the control system due to the noncolocation of the joint position sensor (usually measuring actuator position) and the joint axis of rotation [26].

In addition to flexibility in the manipulator joints, friction in the joints has a significant effect on controlling robot manipulators. Significant research has been conducted looking at how best to model and compensate for friction that affects position control. While still notable, joint friction becomes less significant when considering constrained motion, as the friction between the end-effector and the environment is orders of magnitude larger [27]. While a number of early position/force control schemes neglected friction between the robot and environment, overcoming this friction to achieve the desired relative motion between the two elements is an important consideration.

All controllers need to be stable within their working environment to be considered effective. Due to the nonlinear dynamically coupled nature of manipulators, especially in constrained motion, the stability of any proposed control system needs to be strongly considered. With the numerous potential sources of instability listed below, this can prove a challenging task [26], [28]:

- robot parameters that can vary (friction, joint wear, end-effector masses)
- external disturbances (sensor noise, interaction forces)
- desire for high response speeds with manipulators
- difficult to model workpiece dynamics
- impact forces on end-effector/workpiece
- actuator saturation
- sensor bandwidth limitations.

Finally, most research into robotics aims to find application in industry. To be considered for implementation, control strategies must be simple enough for the average robotics technician to setup, general, robust and efficient enough to cover a broad range of applications, and responsive to optimization so the performance of the application can be improved [6].

1.3 Research Objectives

The primary objective of the research performed in this thesis is to develop and implement both position and position/force control for use with the CataLyst-5 five degrees of freedom articulated robot arm. The first goal is to create an acceptable model on which to simulate the application of the various control methods, in order to both troubleshoot and optimize them prior to applying them to the actual robot. The next goal is to develop increasingly more complex position control strategies for use with the simulated model and eventually the physical robot.

Using these position controllers as a baseline the next objective is to develop position/force control strategies to allow the robot to come into contact with the environment under control. Once in contact the controllers are to apply a desired force in 1 end-effector degree of freedom while controlling the position in the remaining degrees of freedom. With all controllers proven stable in simulation the final research objective is to configure the robot system to be able to operate in open architecture mode, establish appropriate means of measuring applied force and apply the position and position/force controllers, comparing the results to those found in simulation.

1.4 Thesis Organization

The focus of this thesis is on the dynamic modelling of the CataLyst-5 robot in direct support of the development of position and position/force control methods to be applied both in simulation and to the physical robot. The thesis is divided into 9 chapters. Chapter 1 provides an introduction to the field of robot position/force control including motivation behind conducting this research and the important applications for this type of manipulator control in industry. Chapter 2 summarizes what can be considered the foundational methods of manipulator position/force control, provides a means of categorizing the various control strategies and gives more detail in hybrid position/force control, which is the method of primary focus in this research. A detailed description of the methods used in creating the dynamic model of the CataLyst-5 and the supporting mathematical equations used with the controllers can be found in Chapter 3.

Chapter 4 details the methods used in creating the trajectories to be followed by the CataLyst-5 and explains the 3 position and 2 position/force control methods that were developed and trialed both in simulation and experimentation. The results and discussion of the simulated control of the dynamic model can be found in Chapter 5. Chapter 6 provides a detailed look at the CataLyst-5 robot system, the open architecture control software, the hardware and the method used for measuring the applied force and the steps followed in configuring the robot to be used in open architecture control mode. The results and associated discussion regarding the application of the controllers on the physical robot can be found in Chapter 7. Lastly, conclusions regarding this thesis and recommendations for future development of position/force control with the CataLyst-5 can be found in Chapters 8 and 9, respectively.

1.5 Conclusion

The field of position and force control of robotic manipulators was introduced with the intent of explaining the motivation for the research conducted in this thesis. A summary of the objectives of this research were presented including developing and implementing various control schemes on both a simulated and physical robot, the CataLyst-5. Finally the organization of the thesis was described with a short synopsis of each chapter included.

2 Literature Review

The following chapter starts with a section describing a number of control techniques that can be considered as foundational to the field of manipulator position/force control. A number of means of categorizing the various force control schemes are then presented. The chapter finishes with a section describing the various modifications that have been made to the hybrid position/force control scheme including, the application of modern control mechanisms, the adaptation of the models used for the two control loops and various novel methods of defining the task trajectory required for this method.

2.1 Foundational Force Control Methods

In the field of position and control of robot manipulators, a number of papers present methods that can be considered foundational in that they are regularly referenced in all modern force control research. In [29] the author proposes a method that has become known as Accommodation Control. As a means of controlling fine motions, based on closed loop force feedback, he presents a formal representation of vector force feedback strategies and discusses the need for sensors that can accurately capture the vector of applied forces in sufficient detail to guide assembly tasks. The accommodation control strategy consists of sending a velocity command to the manipulator then multiplying the sensed forces (and torques) by a matrix of damping coefficients that reduce or cancel the motion along certain axes of the end-effector frame, thus limiting the applied force [29].

Active Stiffness Control makes use of the observation that when used for assembly purposes, the inherent compliance of manipulator based systems (joint elasticity, structure flexibility, etc.) is often what allows the assembly to proceed [23]. The author proposes actively controlling the apparent stiffness of the reference frame attached to the end-effector, making stiffness low in the direction where the end-effector is expected to contact the environment

to ensure low contact force and high in the direction where contact is not expected so as to ensure good tracking of the desired position. The end-effector frame stiffness is controlled through the use of a stiffness matrix that defines values appropriate to the task. The required stiffness of each joint related to these values are determined using the Jacobian of the mechanism. Using this method, controlling the joint positions simultaneously controls the end-effector position and limits the force applied to a constraining surface [23].

The primary contribution of [10] is a standardized method of describing the manipulator task geometry for constrained motion. The idea of dividing constrained motion into natural and artificial constraints is formalized in this publication. A formal model of the manipulator is presented using the ideal effector, consisting of a point in position space and a point in force space to represent the position and applied force of the end-effector. A proposed model of the task geometry is represented by the ideal surface, consisting of a smooth hypersurface of the possible positions of the ideal effector. A goal trajectory, that lies along the ideal surface, is used to model the desired behaviour of the ideal effector as a functions. The paper also discusses both passive and active compliant control of robot manipulators, but with the focus on active compliance which is referred to as force control by the author. As the compliance of the mechanism can be controlled, 'force control' is advantageous as it can be adapted to different tasks.

In what is one of the most widely cited papers in the field [30], the authors propose a method of simultaneously controlling both the position and the force applied by the end-effector on the environment with Hybrid Position/Force Control. Hybrid control makes use of the natural and artificial constraints, noted above, as a means of dividing the position and force control problems into subtasks [31], splitting end-effector axes between those that require position movements and those axes along which to apply forces. With the task divided among axes, 2 separate control loops are used in unison, with the appropriate input selected from each loop based on the task definition. These selected outputs are then summed to act cooperatively to control each joint of the manipulator, with each joint contributing to both position and force. The decomposition of the task into purely motion controlled directions and purely force controlled directions is based on the assumption of ideal constraints consisting of rigid, frictionless contact with a perfectly known geometry [4]. That this method, as originally proposed, requires such detailed knowledge of the task and contact environment is a significant deficiency that has undergone considerable investigation.

Published in three parts, the most cited paper in the field of force control, Impedance Control [32] proposes simultaneously controlling the dynamic

behaviour of the manipulator as it interacts with the environment while also controlling the position or velocity. Founded on physical systems theory, the objective is to impose a task appropriate dynamic behaviour on the manipulator, by controlling the gains to give the robot arm a desired impedance. The manipulator may experience both free and constrained motion, thus the behaviour needs to be adaptable, with the controller capable of modulating the impedance based on the phase of the task. The imposed impedance should be based on the dominant dynamic behavior of the manipulator and the task it is performing. As example, inertial effects could be ignored for a manipulator operating under water, while the impedance selection for a robot operating in space should only consider the inertia effects. The choice of the impedance should minimize deviations from desired motions while simultaneously reducing / controlling interaction forces. The manipulator should accommodate a stiff environment (low admittance) with low impedance and alternatively for a contact with a compliant surface (high admittance), where motion may be imposed, the manipulator should have high impedance. The most significant deficiency with impedance control, as originally proposed, is that a specific force cannot be ordered and applied to the contact surface using this technique.

Multiple variations on both the hybrid and impedance control methods have been proposed since they were originally presented. Three of these variations can be considered as foundational force control methods in their own right as they are regularly cited in this research field. The operational space formulation of hybrid position/force control is proposed in [33], describing both the end-effector task description and the manipulator dynamics in the operational space. The operational space task description is advantageous, as constrained motion when viewed from the end-effector reference frame only requires two intuitive elements to be complete; vectors of the force and moment required to maintain the constraint and a specification of the end-effector motion degrees of freedom and direction. The operational space dynamic model, describing how motions along the end-effector axes interact and how the inertia and mass varies with the manipulator configuration, is achieved by modifying the joint based dynamic model. The joint based dynamic model is modified so as to determine the relationship between end-effector position, velocity and acceleration with respect to the virtual forces acting along its axes [33]. In this paper, the author also discusses extending his formulation to redundant manipulators and proposes a new approach for dealing with kinematic singularities by treating the manipulator as redundant with respect to the motion of the end-effector.

The design proposed in [34] improves upon the original hybrid control

scheme by rigorously considering the dynamics of the manipulator in contact with the environment when formulating position and force control mechanisms. This is accomplished through the use of constraint hypersurfaces in the end-effector coordinate system for the purpose of creating a complete dynamic model. Attempting to combine the robustness to environmental uncertainties of impedance control with the hybrid ability to control both end-effector position and applied force, a parallel position/force control scheme aims to correct the deficiencies of the two primary foundational methods by combining them [35]. Using this method, the control action takes place without the use of selection matrices defining force or position control, thus allowing all sensor information to be considered at all times when controlling the manipulator. Conflicting control direction between the position and the force control loops are managed by placing a higher priority on the force control loop, which ensures limited deviation from the desired applied force [35].

2.2 Force Control Categorization

A couple of published papers have attempted to summarize the field of position and force control of manipulators, while also categorizing the multiple control theorems that have been proposed. [36] divides force control into fundamental force control and advanced force control categories.

The fundamental force control category methods include those:

- involving a relation between position and applied force (both position and force based stiffness control),
- applying relation between velocity and applied force (impedance and admittance control),
- applying direct position and applied force (hybrid and hybrid impedance), and
- applying force feedback directly (explicit force control).

The advanced force control category is based on integrating or applying adaptive control, robust control and/or learning methods to the fundamental methods discussed above. As described in [36], learning methods include neural network and fuzzy control techniques as applied to position and force control. Though not a paper summarizing the field of manipulator force control, [37] proposes dividing up the various methods used to regulate contact force into two basic strategies; direct force control and inner position/outer force control. The former includes methods that convert the measured force error directly into appropriate actuator force/torques and includes the joint and operational space formulation of hybrid control and parallel position/force

control. Inner position/outer force control methods involve a force control loop closed around and providing input to an inner position control loop and are designed for use with existing industrial manipulators that generally only have position based servos.

Providing a very thorough overview of the field, including dynamic modelling of the manipulator and its interaction with the environment, a summary of the basic approaches to force control, stability concerns, multi-arm system considerations and notes on task formulation and programming, [11] provides good insight for those new to this area of research. The paper proposes three different basic ideas to simultaneously control the position and applied force with robot manipulators; hybrid position/force control, impedance control and linear optimal control. The latter is not discussed with sufficient detail (or example) to be considered relevant. Both of the force control summary papers, [4] and [5], concur with the idea of categorizing control strategies as either hybrid or impedance based. The control schemes discussed above, that can be considered as a variation of hybrid control include; dynamic hybrid control [34], parallel force control [35] and task space hybrid control [33]. [5] identifies impedance control as a generalization of both stiffness control [23] and admittance control [29], while [11] identifies these foundational methods as subclasses of impedance control. Hybrid impedance control [38] is categorized as an impedance control scheme in [4], though an argument could be made that it could also be considered a hybrid control scheme. As it is the more intuitive to apply of the two core control methods, additional research was conducted into the advances that have been made in the field of hybrid position/force control.

2.3 Modifications and Variations on Hybrid Position/Force Control

Since it was first proposed in [30], hybrid position/force control, is one of the most widely understood methods of implementing position and force control with robot manipulators and as such, many variations of have been proposed. Some of these variations have become what can be considered foundational techniques themselves, while some have combined hybrid with the foundational techniques as discussed above. As a means of reducing the computational complexity of the force control loop, [39] proposed using stiffness control [23] specified in the operational space, with position control in the joint space as part of a hybrid control strategy. Various novel applications of hybrid control as originally proposed by [30] have also been discussed. One such scheme

proposes using hybrid control to control the motion of a manipulator that is constrained by the environment at some point other than the end-effector [40]. This is accomplished through the use of a separate Jacobian referencing the constrained position, in the force control loop.

The desired and applied force can more readily be defined and measured in the operational space and the position control can be managed in the operational space using the manipulator kinematic models. Dividing up control inputs into position and force tasks is more readily achieved with both described in the operational space. Further, manipulator dynamics can be described in the operational space, allowing position control to be linearized and decoupled for better accuracy. Various papers describe applying hybrid control in the operational space, with [41] also proposing a dynamic model of the force sensor for inclusion in the force control loop and [33] proposing a transition impact control loop based on the velocity of the end effector.

Identifying that the original hybrid control methodology does not adequately account for the dynamic effects of the manipulator contacting the environment, [34] proposes a method of correcting this deficiency. The dynamic hybrid controller uses constraint hypersurfaces that are defined in the end-effector coordinate system to model the dynamics of the manipulator's interaction with the environment, with the effects accounted for in the control loops.

2.3.1 Modern Control Mechanisms

Many variations on hybrid control involve the use of more advanced control mechanisms within either the position and/or force control loops. Hybrid velocity/force control was proposed as a means of executing compliant motions over unknown objects [42]. Velocity control is used in lieu of position control as the latter requires global knowledge of the object with which the manipulator is maintaining contact, while velocity control only requires local information. First and second order sliding mode control in both the force and position control loops combined with a detailed model of the dynamics of the force sensor was proposed in [43].

Perhaps the most prevalent modern control method applied to hybrid control is fuzzy logic. A switching hybrid fuzzy controller proposed in [44] implements fuzzy control in both position and force control loops. A mechanism to switch to the force controller once contact is made, as a means of accommodating the reactive forces of the environment makes this method something of a deviation from traditional hybrid control. With what is another deviation from standard hybrid control, [45] uses fuzzy rules for tuning the PI gains of a

force control loop that can be fed directly into the position control servos that exist in most established industrial manipulators. A similar control method, proposes the use of a high level fuzzy logic force controller in [46], allowing the use of existing position only control of industrial manipulators.

In presenting a robot to assist in the rehabilitation of people with a neuromuscular disorder, [47] employs a traditional hybrid approach, but with a joint PD/fuzzy PD position loop and a joint PI/fuzzy PI force control loop. A combination of fuzzy and neural control is implemented in [48] to compensate for unmodelled dynamics (e.g. modelling friction effects between the end-effector and the environment).

2.3.2 Model Adaptation

Feedforward compensation for manipulator dynamics, friction and stiction, used in conjunction with hybrid position/force control is presented in [49], where a learning algorithm modifies the estimate of system parameters used in this compensator, based on the errors found in repetitive tasks. Similarly, [50] proposes a non fuzzy based learning algorithm to improve upon the idealized constrained manipulator dynamic model used to linearize the force control loop of the hybrid controller. Fuzzy logic is used outside of the control loop in [51] and [52], where fuzzy algorithms are used to compensate for uncertainties in the dynamic modelling of the robot manipulator and errors arising due to the contact force with the environment. The proposed method in [51] also provides an adaptive update of the task trajectory commands (compliance selection matrix), an important aspect of hybrid control.

2.3.3 Task Trajectory Generation

As the generation of the compliance selection matrix, especially when there is limited knowledge of the environment, is the biggest challenge in implementing hybrid position/force control, significant research has been conducted in this area. [53] proposes determining the model of unknown environmental parameters, and thus producing the compliance matrix, by looking at the stability of both the position and force control loops. The position and force degrees of freedom are determined based on the type of contact that the manipulator has with the environment (point, line, plane) [54]. The type of contact that's experienced is determined based on the point of contact information received from a wrist mounted force/torque sensor. Standardizing and converting a description of the desired contact path into a useable compliance selection

matrix is also a key part of using hybrid force control. [55] proposes a general and automated approach to this task.

The application of fuzzy methods to task description has received significant attention in recent research. A fuzzy vector method making use of signals from a force/torque sensor is used in [56] to determine the vector normal to an unknown constraint surface. The hybrid controller presented in [56] uses this updated information to adjust the direction in which force is being applied in the force control loop. An outer loop command generator makes use of information from the force sensor to determine the required robot motion profile and the desired chamfering force in [57]. The generated commands are then used in conjunction with the adaptive fuzzy hybrid controller [51] discussed above. In a very complete strategy, [58] proposes using three prediction factors to estimate and then adjust the desired force and position trajectories used in hybrid control. These factors are designed to account for trajectory prediction error, geometric parameter change and contact force error with each prediction taking into account the last prediction, environmental curvature, and stiffness.

2.4 Conclusion

A summary of the literature from the field of position and force control reviewed as part of this research was presented, starting with a summary of what can be considered foundational methods. Various means of categorizing and defining force control methods were then discussed. Lastly, as it was the prime control scheme investigated in this thesis, the foundation control method hybrid position/force control was reviewed in greater detail, including a summary of literature involving advanced control research that builds on this method.

3 System Modelling

In order to evaluate the suitability of the potential position/force control methods on the CataLyst-5 robot, a mathematical dynamic model of the robot was developed. The modelization was accomplished in three steps, details of which are provided in the sections below. The steps are the development of a solution to the Direct Kinematic Problem (DKP), the development of a solution to the Inverse Kinematic Problem (IKP) and the development of the dynamic model using the Lagrangian formulation as described in [59] and [25].

3.1 Development of Solution to Direct Kinematic Problem

The Denavit Hartenberg (DH) axes are defined for the CataLyst-5 serial arm robot in the Ready position as described in [60]. Alignment gauges adhered to the robot's joints allow for the device to be readily configured in this position, thus it is the logical orientation for defining these axes. The orientation and position of the axes for each joint and the end-effector, as shown in Figure 3.1, were placed according to the classical DH notation [61] as summarized in [59] and below:

- the i th joint connects the $(i-1)$ th link to the i th link
- \mathcal{F}_i is attached to, and moves with, the $(i-1)$ th link
- Z_i is the axis of motion of the i th joint
- X_i is the common perpendicular between, and directed from, Z_{i-1} to Z_i
 - if Z_{i-1} and Z_i intersect the direction of X_i is determined using the right hand rule
 - if Z_{i-1} and Z_i are parallel, X_i is placed such that it passes through the origin of \mathcal{F}_{i-1}
- Y_i is determined using the RH rule.

Using these axes, and based on the definitions below, the DH parameters for the robot were determined and can be found in Table 3.1, which also

3.1. Development of Solution to Direct Kinematic Problem

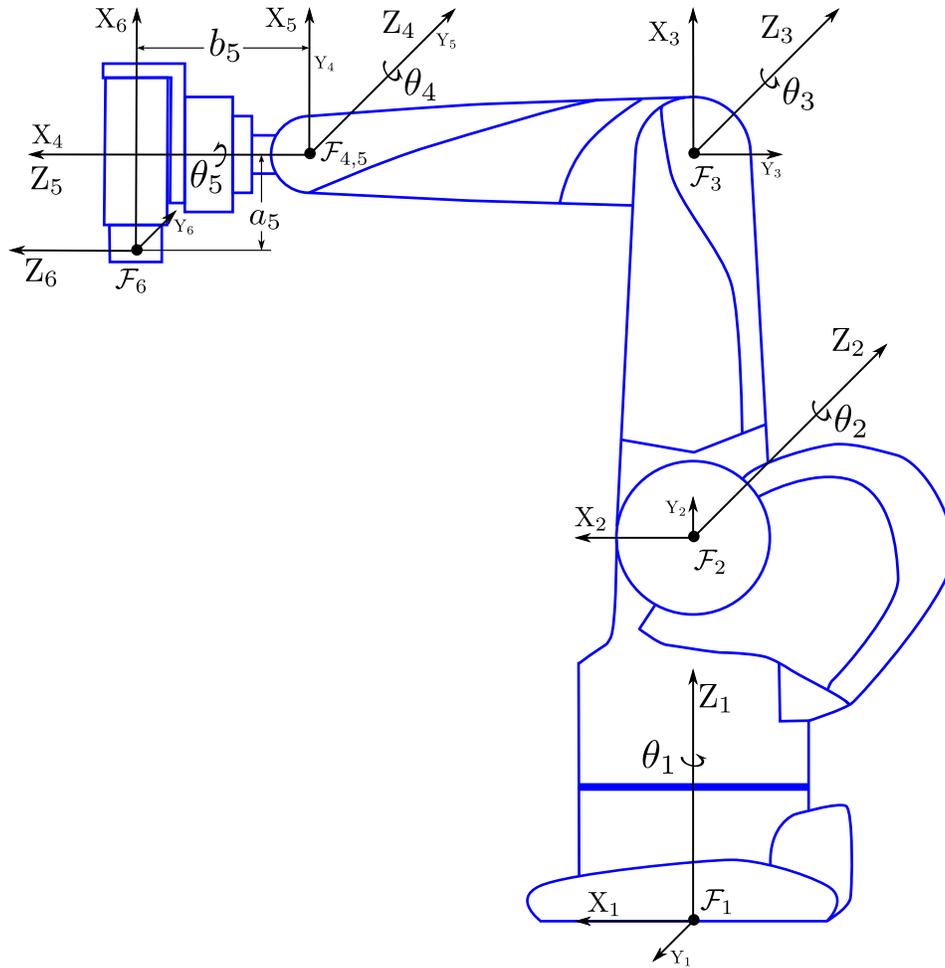


Figure 3.1: DH Axes - Ready Position

contains the physical joint limits as found in [60].

- a_i is link length which is defined as the distance between Z_i and Z_{i+1} along X_{i+1}
- α_i is the link twist which is defined as the angle to bring Z_i parallel with Z_{i+1} rotated about the positive X_{i+1} axis
- b_i is the joint offset which is defined as the distance along Z_i where X_{i+1} intersects Z_i
- θ_i is the joint angle and is defined as the angle to bring X_i parallel with X_{i+1} rotated about the positive Z_i axis.

3.1. Development of Solution to Direct Kinematic Problem

Table 3.1: DH Parameters and Joint Limitations (from [60])

Joint	a	α	b	θ	Range	Ready
1	0mm	90°	254mm [10"]	θ_1	-179° to 180°	0
2	254mm [10"]	0°	0mm	θ_2	0 to 109°	90°
3	254mm [10"]	0°	0mm	θ_3	-125° to 0	-90°
4	0mm	90°	0mm	θ_4	-19° to 199°	90°
5	58.7mm	0°	116mm	θ_5	-179° to 180°	0

Using the DH parameters from Table 3.1, the orientation of \mathcal{F}_{i+1} relative to \mathcal{F}_i can be found using Eq. 3.1. Similarly, the coordinate vector of the origin of frame \mathcal{F}_{i+1} with respect to frame \mathcal{F}_i can be found using Eq. 3.2.

$$Q_{i+1}^i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix} \quad (3.1)$$

$$[\vec{a}_i]_i = \begin{bmatrix} a_i \cos \theta_i \\ a_i \sin \theta_i \\ b_i \end{bmatrix} \quad (3.2)$$

NOTE: Throughout the report the notation $[\dots]_i$ refers to the term within brackets being provided with respect to the i th reference frame. [59]

The orientation of the end-effector with respect to the base reference frame, as a function of joint angle, is obtained using Eq. 3.3, which pre-multiplies each of the respective reference frames by the rotation matrices. The position of the end-effector (the origin of \mathcal{F}_6) can be found using Eq. 3.4 which rotates each of the frame position vectors to a common reference frame, in this case the base reference frame, then sums them.

$$Q = Q_2^1 Q_3^2 Q_4^3 Q_5^4 Q_6^5 \quad (3.3)$$

$$\vec{P} = [\vec{a}_1]_1 + Q_2^1 [\vec{a}_2]_2 + Q_2^1 Q_3^2 [\vec{a}_3]_3 + Q_2^1 Q_3^2 Q_4^3 [\vec{a}_4]_4 + Q_2^1 Q_3^2 Q_4^3 Q_5^4 [\vec{a}_5]_5 \quad (3.4)$$

Eqs. 3.3 and 3.4 treat the orientation and position of the end effector separately, as is done throughout this document. An alternative representation

3.1. Development of Solution to Direct Kinematic Problem

combining orientation and position, the homogeneous transformation matrix as used in [15], is found below and can be considered equivalent.

$$H = \begin{bmatrix} Q_{XX} & Q_{YX} & Q_{ZX} & P_X \\ Q_{XY} & Q_{YY} & Q_{ZY} & P_Y \\ Q_{XZ} & Q_{YZ} & Q_{ZZ} & P_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

As Eqs. 3.3 and 3.4 provide lengthy equations that do not readily provide insight into a potential solution to the IKP, a more direct geometric solution to the DKP was determined, with details of the derivation found in Appendix A. The more concise equations solving the DKP are:

$$Q = \begin{bmatrix} -\cos \theta_5 \sin \phi \cos \theta_1 + \sin \theta_5 \sin \theta_1 & -\sin \theta_5 \sin \phi \cos \theta_1 + \cos \theta_5 \sin \theta_1 & \cos \phi \cos \theta_1 \\ -\cos \theta_5 \sin \phi \sin \theta_1 - \sin \theta_5 \cos \theta_1 & \sin \theta_5 \sin \phi \sin \theta_1 - \cos \theta_5 \cos \theta_1 & \cos \phi \sin \theta_1 \\ \cos \theta_5 \cos \phi & -\sin \theta_5 \cos \phi & \sin \phi \end{bmatrix} \quad (3.6)$$

and

$$\vec{P} = \begin{bmatrix} [a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3) + b_5 \cos \phi + |a_5| \cos \theta_5 \sin \phi] \cos \theta_1 - |a_5| \sin \theta_5 \sin \theta_1 \\ [a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3) + b_5 \cos \phi + |a_5| \cos \theta_5 \sin \phi] \sin \theta_1 + |a_5| \sin \theta_5 \cos \theta_1 \\ b_1 + a_2 \sin \theta_2 + a_3 \sin(\theta_2 + \theta_3) + b_5 \sin \phi - |a_5| \cos \theta_5 \cos \phi \end{bmatrix} \quad (3.7)$$

where

$$\phi = \theta_2 + \theta_3 + \theta_4 - 90^\circ \quad (3.8)$$

The concise equations presented above were compared with the analytical solution found at Eqs. 3.3 and 3.4 using a loop of one-thousand random vectors of joint angles with the exact same position and orientation provided each time, proving the validity of the geometric solution.

A number of poses/joint angle combinations were checked using the DKP to further validate the geometric solution. The results of this validation can be found in Table 3.2, with the positions checked being the ready position, as shown in Figure 3.1, the zero-angle position as shown in Figure 3.7 and a pose with the robot arm straight up.

3.1.1 Task Specific DKP

As the task to be performed in simulation and experimentation will consist of the robot following trajectories on an X-Y plane of the world coordinate

3.1. Development of Solution to Direct Kinematic Problem

Table 3.2: Position and Orientation Using Geometric Solution to DKP

Pose Name	$\vec{\theta}$ (deg)	\vec{P} (mm)	Q
Ready	$\begin{bmatrix} 0 \\ 90 \\ -90 \\ 90 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 370 \\ 0 \\ 449 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$
Zero-Angle	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 449 \\ 0 \\ 138 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$
Straight Up	$\begin{bmatrix} 0 \\ 90 \\ 0 \\ 90 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 58.7 \\ 0 \\ 878 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

system, Joint 5 will be fixed at 0 radians and the orientation of the end effector will be such that ϕ is always equal to 0 radians. As the orientation for this task is restricted, a simplified 3×1 vector method of representing orientation was developed. This vector representation is also a requirement for Cartesian space control that involves calculating the error in both position and orientation in determining the required control effort. The vector representation, referred to as Simple Angle, uses roll, pitch and yaw of the end effector reference frame with respect to the base reference frame. The orientation angles are 0 when the end-effector is positioned in the ready position. The simple angle orientation vector is given by the following equation.

$$\vec{S}_{angle} = \begin{bmatrix} roll \\ pitch \\ yaw \end{bmatrix} = \begin{bmatrix} r \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \theta_5 \\ -\phi \\ \theta_1 \end{bmatrix} \quad (3.9)$$

Concatenating Eq. 3.9 with Eq. 3.7 gives a 6×1 vector that can completely describe the orientation and position of the CataLyst-5 when used for the designated experimental tasks of this thesis.

3.2 Solution to Inverse Kinematic Problem

As input to the IKP solution, both the orientation matrix of the end-effector (Q) and the end-effector position vector (\vec{P}) are provided, and the intent is to find a combination of joint angles ($\vec{\theta}$) that allows for this configuration. A summary of the geometric solution to the IKP that was developed for the CataLyst-5 can be found below and in Figure 3.2, with the details of this solution provided in Appendix A.

As the CataLyst-5 robot is planar between joints 2 and 4, knowledge of the positions of the origins of the reference frames attached to these joints can be useful in determining the joint angles. The first joint angle θ_1 can be determined from Eq. 3.10, where C_X and C_Y are the X and Y positions of the origin of \mathcal{F}_4 .

$$\theta_1 = \text{atan2}(C_Y, C_X) \quad (3.10)$$

With the value of θ_1 known, the radial positions of the origin of \mathcal{F}_4 (C_R) and a point D located a distance b_5 along the Z_5 axis (D_R), can be found and used to determine the value of ϕ . Based on the differences of radial and vertical positions of points C and D, ϕ can be found using Eq. 3.11.

$$\phi = \text{atan2}(D_Z - C_Z, D_R - C_R) \quad (3.11)$$

With the value of ϕ known, θ_5 can be determined using the values of X_{6Z} (value of \mathcal{F}_6 X axis orientation unit vector along Z world axis) and Y_{6Z} (value of \mathcal{F}_6 Y axis orientation unit vector along Z world axis) as follows in Eq. 3.12.

$$\theta_5 = \text{atan2}\left(-\frac{Y_{6Z}}{\cos \phi}, \frac{X_{6Z}}{\cos \phi}\right) \quad (3.12)$$

As Eq. 3.12 is undefined for values of $\phi = \pm\frac{\pi}{2}$, an alternative solution is needed along with an algorithm to choose the correct equation based on the value of ϕ . These alternative solutions are shown below with Eqs. 3.13 and 3.14.

$$\text{for } \phi = \frac{\pi}{2}, \theta_5 = \text{atan2}(-X_{6Y}, -X_{6X}) - \theta_1 \quad (3.13)$$

$$\text{for } \phi = -\frac{\pi}{2}, \theta_5 = \theta_1 - \text{atan2}(X_{6Y}, X_{6X}) \quad (3.14)$$

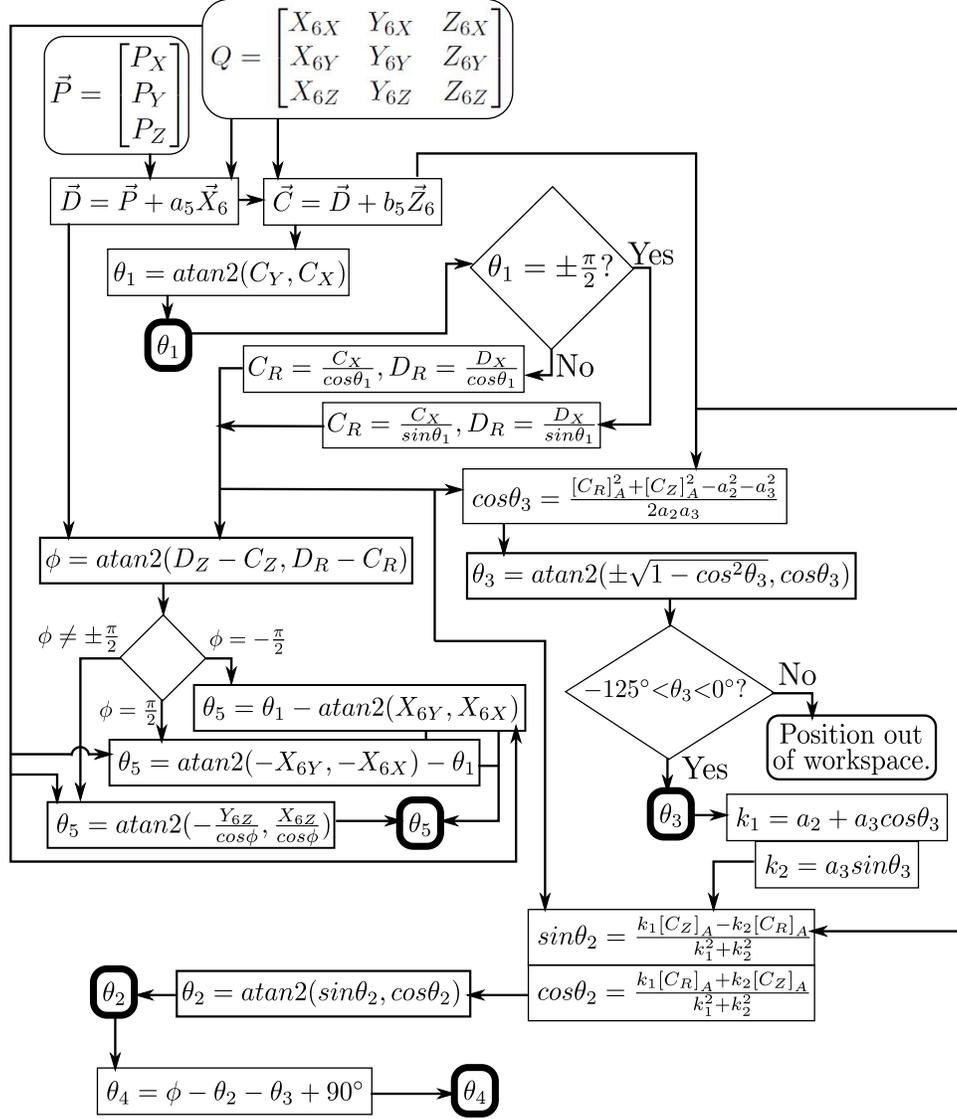


Figure 3.2: IKP Solution Flow Chart

The cosine of θ_3 can be found using Eq. 3.15 which makes use of the radial (R) and vertical (Z) positions of point C, but this time with respect to point

A (origin of \mathcal{F}_2).

$$\cos \theta_3 = \frac{[C_R]_A^2 + [C_Z]_A^2 - a_2^2 - a_3^2}{2a_2a_3} \quad (3.15)$$

Two potential values of θ_3 can be determined using Eq. 3.16 which requires that the algorithm used with this IKP solution be capable of determining which (if any) of the values of θ_3 are within the range of motion of the joint.

$$\theta_3 = \text{atan2}(\pm\sqrt{1 - \cos^2 \theta_3}, \cos \theta_3) \quad (3.16)$$

With an acceptable value of θ_3 determined, θ_2 can be found using Eqs. 3.17, 3.18 and 3.19 where $k_1 = a_2 + a_3 \cos \theta_3$ and $k_2 = a_3 \sin \theta_3$.

$$\sin \theta_2 = \frac{k_1 [C_Z]_A - k_2 [C_R]_A}{k_1^2 + k_2^2} \quad (3.17)$$

$$\cos \theta_2 = \frac{k_1 [C_R]_A + k_2 [C_Z]_A}{k_1^2 + k_2^2} \quad (3.18)$$

$$\theta_2 = \text{atan2}(\sin \theta_2, \cos \theta_2) \quad (3.19)$$

Finally, θ_4 can be determined by rearranging Eq. 3.8 to give the following.

$$\theta_4 = \phi - \theta_2 - \theta_3 + 90^\circ \quad (3.20)$$

The analytical solution was proved accurate by first finding the orientation and position of a random vector of joint angles using the concise DKP equations presented at Eqs. 3.6 and 3.7. The orientation matrix and position vector were then used as input to the IKP solution with the joint angles found compared to the original random vector. This process was repeated thousands of times using a loop, with the exact joint angles found each time. The IKP solution applied to the orientations and positions found in Table 3.2 returned the same original joint angles.

3.2.1 Task Specific IKP

As the task to be performed in experimentation will consist of the robot following trajectories on the X-Y plane of the world coordinate system, a slightly simplified version of the solution to the IKP problem was created. As the required task dictates that ϕ and θ_5 both be equal to 0, the matrix defining the orientation of the end effector (Q), a contributing component to the IKP solutions described above, is greatly simplified as shown in Eq. 3.21.

By observation, with θ_5 fixed, θ_1 can be found using Eq. 3.22, where X and Y are the Cartesian coordinates of the end-effector. The remaining joint angles can be found using a slightly simplified version of Eqs. 3.15 through 3.20.

$$Q = \begin{bmatrix} 0 & \sin \theta_1 & \cos \theta_1 \\ 0 & -\cos \theta_1 & \sin \theta_1 \\ 1 & 0 & 0 \end{bmatrix} \quad (3.21)$$

$$\theta_1 = \text{atan2}(Y, X) \quad (3.22)$$

3.3 Development of the Dynamic Model

With the goal of developing a mathematical dynamic model to represent the CataLyst-5 robot to both simulate the application of various position/force control methods and for use in linearized controllers, it was determined that the Lagrangian formulation, as derived in [15] and [25], would best provide a suitable model. The Lagrangian formulation uses an energy based approach to create the dynamic model of the serial arm robot, where the Lagrangian of the mechanical system is defined as the difference between its kinetic and potential energy, as shown in Eq. 3.23.

$$\mathcal{L} = T - V \quad (3.23)$$

From the definition of the Lagrangian given above, the equations of motion can be expressed by Eq. 3.24, where q_i is the i th generalized coordinate of the mechanical system (θ_i for the CataLyst-5 with only revolute joints) and f_{q_i} is the generalized force (or moment) associated with the i th coordinate.

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = f_{q_i}, \quad i = 1, 2, \dots, n \quad (3.24)$$

The kinetic energy of a serial arm manipulator is a function of the joint velocities and the inertia of its rigid links as shown in Eq. 3.25, where $\dot{\boldsymbol{\theta}}$ is the joint velocities and $M(\boldsymbol{\theta})$ is the generalized inertia tensor.

$$T = \frac{1}{2} \dot{\boldsymbol{\theta}}^T M(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}} \quad (3.25)$$

The generalized inertia tensor is given by the Eq. 3.26 where m_i is the mass of i th link and I_i is inertia tensor of the i th link.

$$M(\boldsymbol{\theta}) = \sum_{i=1}^5 (m_i J_{\dot{c}_i}^T J_{\dot{c}_i} + J_{\dot{\omega}_i}^T I_i J_{\dot{\omega}_i}) \quad (3.26)$$

The remaining components of $M(\boldsymbol{\theta})$ are as follows:

- Jacobian of Joint Velocities to Velocity of i th Link Center of Mass

$$J_{\dot{c}_i} = [j_{\dot{c}_{i1}}, j_{\dot{c}_{i2}}, j_{\dot{c}_{i3}}, j_{\dot{c}_{i4}}, j_{\dot{c}_{i5}}] \quad (3.27)$$

where $j_{\dot{c}_{ij}}$ is the j th column of $J_{\dot{c}_i}$ given by

$$j_{\dot{c}_{ij}} = \begin{cases} \vec{e}_j \times \vec{r}_{ji} & \text{if } j \leq i \\ 0 & \text{if } j > i \end{cases}$$

\vec{e}_j is the unit vector directed along the axis of actuation of the j th joint (z-axis when using DH parameters).

$$\vec{e}_j = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.28)$$

\vec{r}_{ji} is a vector from the origin of the j th reference frame to the center of mass of the i th link, given by Eq. 3.29, where \vec{a}_i is the vector from reference frame i to $i+1$ as found in Eq. 3.2 and \vec{s}_i is the vector from reference frame $i+1$ to the center of mass of the i th link.

$$\vec{r}_{ji} = \vec{a}_j + \vec{a}_{j+1} + \dots + \vec{a}_i + \vec{s}_i, (i, j = 1, 2, \dots, n, j \leq i) \quad (3.29)$$

- Jacobian of Joint Velocities to Angular Velocity of i th Link

$$J_{\omega_i} = [j_{\omega_{i1}}, j_{\omega_{i2}}, j_{\omega_{i3}}, j_{\omega_{i4}}, j_{\omega_{i5}}] \quad (3.30)$$

where $j_{\omega_{ij}}$ is the j th column of J_{ω_i} given by

$$j_{\omega_{ij}} = \begin{cases} \vec{e}_j & \text{if } j \leq i \\ 0 & \text{if } j > i \end{cases}$$

It is important to note that the column components that make up $J_{\dot{c}_i}$ and J_{ω_i} must be in the same reference frame, so an appropriate rotation matrix must be applied. For the derivation of $M(\boldsymbol{\theta})$, as detailed in Appendix B, all vector elements of these Jacobian matrices were rotated to the $(i+1)$ th reference frame.

The gravitational potential energy of a serial arm manipulator is a function of the vertical position of the center of gravity of each of the links with respect to some arbitrary horizontal reference frame (usually chosen to go through the origin of the base reference frame). The total potential energy can be

calculated using Eq. 3.31, where g is the gravitational constant and \vec{e}_g is the directional vector of the force of gravity in the base reference frame coordinate system.

$$V = -g\vec{e}_g^T \left(\sum_{i=1}^5 m_i \vec{c}_i \right) \quad (3.31)$$

For the above equation, \vec{c}_i is the vector position of the center of mass of the i th link as determined from the following equation where the appropriate rotation matrices bring all vectors to the base reference frame.

$$\vec{c}_i = [\vec{a}_1]_1 + Q_2^1 [\vec{a}_2]_2 + Q_2^1 Q_3^2 [\vec{a}_3]_3 + \cdots + Q_2^1 Q_3^2 \cdots [\vec{a}_i]_i + \cdots + Q_2^1 Q_3^2 \cdots \vec{s}_i \quad (3.32)$$

With the kinetic and potential energies of a serial arm robot defined, the general equation of motion found at Eq. 3.24 can be combined with the Lagrangian Eq. 3.23 to give the following form, where the generalized moments on the right hand side will be defined later in this section.

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\boldsymbol{\theta}}} \right) - \frac{\partial T}{\partial \boldsymbol{\theta}} + \frac{\partial V}{\partial \boldsymbol{\theta}} = \vec{\tau} - J^T \vec{w} \quad (3.33)$$

Expanding Eq. 3.33 gives the following more useful form of the equation of motion, where $M(\boldsymbol{\theta})$ is written as M and bold $\boldsymbol{\theta}$ is used to indicate the vector of joint angles for clarity.

$$M\ddot{\boldsymbol{\theta}} + \dot{M}\dot{\boldsymbol{\theta}} - \frac{1}{2}\dot{\boldsymbol{\theta}}^T \frac{\partial M}{\partial \boldsymbol{\theta}} \dot{\boldsymbol{\theta}} + \frac{\partial V}{\partial \boldsymbol{\theta}} = \vec{\tau} - J^T \vec{w} \quad (3.34)$$

Eq. 3.34 is presented in the more compact form found at Eq. 3.35 in [25], [15] and the majority of robot dynamics literature.

$$M(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \vec{v}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) + \vec{g}(\boldsymbol{\theta}) = \vec{\tau} - J^T \vec{w} \quad (3.35)$$

The first term of Eq. 3.35, the generalized inertia tensor, is defined at Eq. 3.26. The second term of the dynamic equation, $\vec{v}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta})$ is a vector of centrifugal and Coriolis terms given by Eq. 3.36, where \dot{M} is the first time derivative of the generalized inertia tensor.

$$\vec{v}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) = \dot{M}\dot{\boldsymbol{\theta}} - \frac{1}{2}\dot{\boldsymbol{\theta}}^T \frac{\partial M(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \dot{\boldsymbol{\theta}} \quad (3.36)$$

The partial derivative of M with respect to $\boldsymbol{\theta}$ is defined as

$$\frac{\partial M(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial M}{\partial \theta_1} \\ \frac{\partial M}{\partial \theta_2} \\ \vdots \\ \frac{\partial M}{\partial \theta_5} \end{bmatrix}$$

The final term of the dynamic equation is as follows, which includes the partial derivative of V with respect to $\boldsymbol{\theta}$.

$$\vec{g}(\boldsymbol{\theta}) = \frac{\partial V}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial V}{\partial \theta_1} \\ \frac{\partial V}{\partial \theta_2} \\ \vdots \\ \frac{\partial V}{\partial \theta_5} \end{bmatrix} \quad (3.37)$$

The terms on the right side of the dynamic equation Eq. 3.35 are the vector of joint control torques ($\vec{\tau}$) and the applied end-effector wrench ($J^T \vec{w}$). The end-effector wrench term will be determined based on input from the force/torque sensor installed between the robot arm and the gripper. The wrench vector has the following form, where n_i is the moment about the i th axis and f_j is the force along the j th axis.

$$\vec{w} = \begin{bmatrix} n_x \\ n_y \\ n_z \\ f_x \\ f_y \\ f_z \end{bmatrix} \quad (3.38)$$

The Jacobian matrix used in the wrench term of the dynamic equation is the Jacobian for the mechanism, that translates joint velocities to task space velocities. It is derived using the following equation, where \vec{e}_i is the unit vector giving the direction of actuation of the i th joint and \vec{r}_j is the vector from the j th reference frame to the end-effector reference frame.

$$J = \begin{bmatrix} \vec{e}_1 & \vec{e}_2 & \cdots & \vec{e}_5 \\ \vec{e}_1 \times \vec{r}_1 & \vec{e}_2 \times \vec{r}_2 & \cdots & \vec{e}_5 \times \vec{r}_5 \end{bmatrix} \quad (3.39)$$

The Lagrangian formulation of the dynamic model described above was used with Maple[®] to derive algebraic matrices of each of the elements of

Eq. 3.35. These algebraic matrices were converted in Maple to a format that would be accepted by MATLAB[®] so that Simulink[®] function blocks could be created for system modelling and control. The Maple derivation of the dynamic equations can be found in Appendix B.

3.3.1 CataLyst-5 Robot Physical Parameter Definition

For the Lagrangian derivation of the dynamic model described above, a number of parameters of the CataLyst-5 were required, including link mass (m_i), link center of mass (s_i) and link inertia tensor (I_i). These values were all obtained from the manufacturer, Thermo-Fisher, but were presented with respect to reference frames differing from those used in Figure 3.1. No diagram was provided with the manufacturers data, as found in Appendix C, only a description that the, “coordinate frames are oriented such that the z-axis is parallel with the axis of revolution and the x-axis is aligned with the length of the link.” An interpretation of this description and an evaluation of the values given for the location of the center of mass determined the manufacturer’s reference frames to be located as shown in Figure 3.3 (shown in brackets).

The first parameter used from the manufacturers data, the mass of each link in pounds mass, could be used directly for each link save the end-effector. The mass of the tool flange was added to the mass of the gripper and the mass of the force torque sensor as found in documentation for each of those items. The mass of the yaw bracket, connecting the force torque sensor to the gripper was determined from the Solidworks[®] model created for its manufacture and added to find the total assembly mass (m_5).

The locations of the centers of mass (CofM) as shown on Figure 3.3 were converted from the values provided with respect to the manufacturer’s reference frames such that they would be with respect to the appropriate body attached reference frame used in modelling the robot. As example, for the dynamic model calculations, the center of mass of the shoulder (m_1) was converted such that its position is provided with respect to reference frame \mathcal{F}_2 . The center of mass of the end-effector assembly was determined using Eq. 3.40 where the center of mass of each element (j) was first determined with respect to reference frame \mathcal{F}_6 ($[CofM_i]_6$). As the necessary information was not available in the supporting documentation, the location of the centers of mass of the gripper and force/torque sensor were determined using the assumption that they were both of a uniform density. This provided the location of the center of mass of the force/torque sensor at the geometric center of the cylindrical shape. The center of mass of the gripper was determined by dividing it

3.3. Development of the Dynamic Model

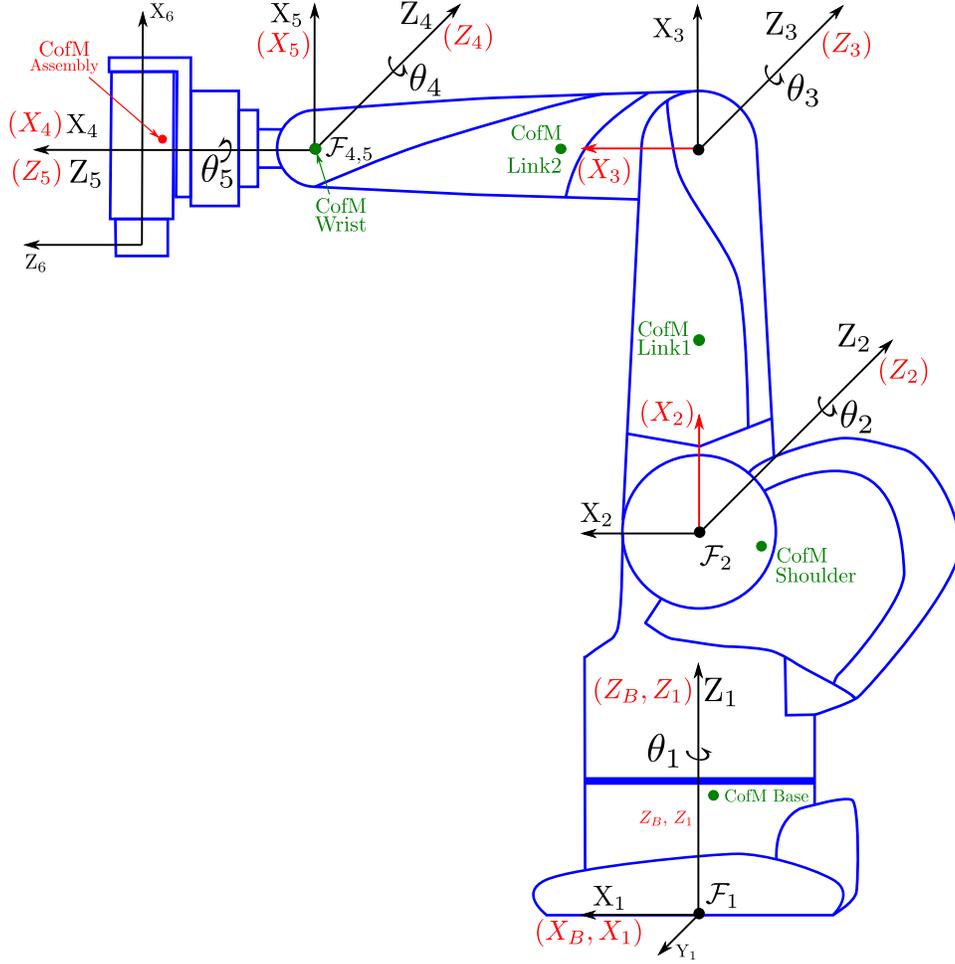


Figure 3.3: Location and Orientation of Manufacturer Axes

into 2 rectangular prisms with a mass proportional to their volume. Each of these rectangular prisms were included separately in Eq. 3.40.

$$[s_5]_6 = \frac{\sum_{j=1}^5 (m_i [CofM_i]_6)}{\sum_{j=1}^5 m_i} \quad (3.40)$$

The final parameter modified from the manufacturer's data were the inertia

tensors of each link. The inertial properties of each link were provided in a number of different formats in the document provided by the manufacturer, most useful was the inertia tensor provided at the center of mass. These values were provided with respect to the reference frame at the base of each respective link. For the dynamic model the inertia tensors were required with respect to the body attached reference frame (ie \mathcal{F}_{i+1} for the i th link). As the inertia tensor was provided at the center of mass, the only conversion required was to align the frames of reference. This was accomplished by creating a rotation matrix between the manufacturers i th reference frame and the DH convention $(i+1)$ th reference frame using Euler angles. These rotation matrices modified the manufacturer supplied inertia tensors using the following equation.

$$[I_i]_{i+1} = (Q_{i+1}^i)^T [I_i]_i Q_{i+1}^i \quad (3.41)$$

Similar to what was done for the center of mass, the overall inertia tensor of the end-effector assembly was determined by first calculating the individual inertia tensors at the centers of mass of each of the individual geometric components. The individual inertia tensors were transferred to the assembly center of mass using the parallel axis theorem, then summed. The inertia tensor at center of mass of the yaw bracket was determined from the Solidworks model, and that of the tool flange was as provided from manufacturer's data.

3.4 Model Development

3.4.1 Simulink Mathematical Model

Re-arranging and solving Eq. 3.35 for $\ddot{\theta}$ gives the following:

$$\ddot{\theta} = M(\theta)^{-1} \left[\vec{\tau} - J^T \vec{w} - \vec{v}(\dot{\theta}, \theta) - \vec{g}(\theta) \right] \quad (3.42)$$

The Simulink function blocks of each element of Eq. 3.42 were used to create the Mathematical Model of the robot as found in Figures 3.4 and 3.5. The two inputs to the model are actuation torque ($\vec{\tau}$) and end-effector wrench (\vec{w}) and after integrating the joint acceleration vector twice the model output is joint position (θ).

3.4.2 SimMechanics Model

As the dynamic equations used with the Simulink model of the CataLyst-5 are highly complex, leading to slow simulations, a second model was created

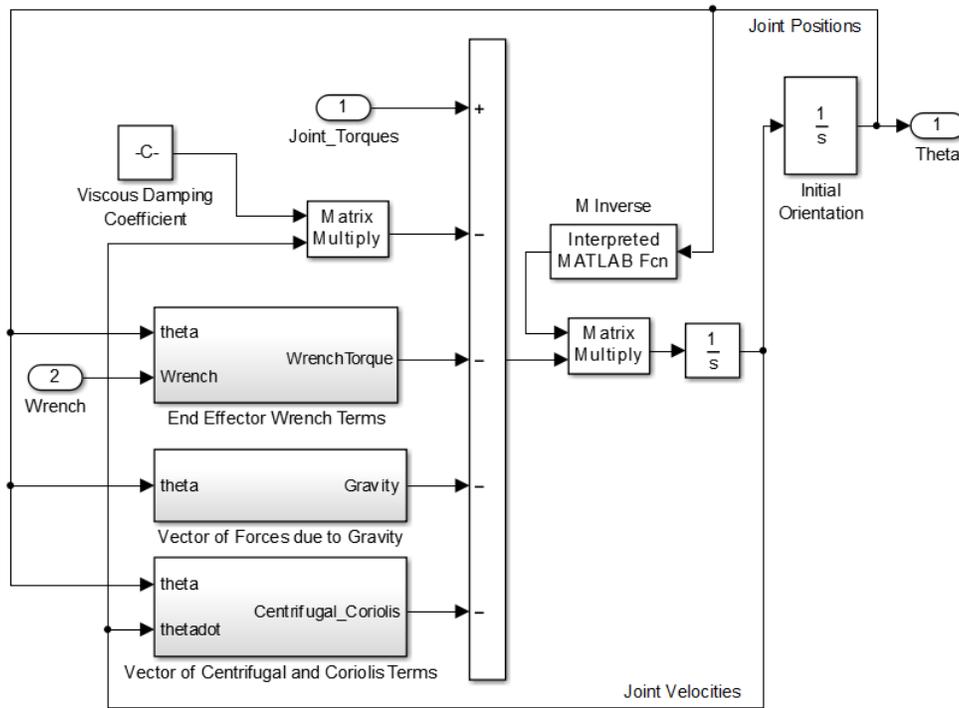


Figure 3.4: Impementation of the Simulink Model

in SimMechanics for comparison. The SimMechanics Model as found in Figure 3.6 makes use of joint and solid body blocks to represent the CataLyst-5 robot. The joint blocks are torque actuated as input and provide a joint position as output. A Transform Sensor block was added to the model and provides both the orientation (via rotation matrix) and the Cartesian location, in world coordinates, of the end-effector.

3.4.3 Model Comparison

A comparison was conducted of both models as a means of assessing their suitability, and determining which to use for control systems development. The first validation consisted of starting the models from the same Zero-Angle position ($\vec{\theta} = [0 \ 0 \ 0 \ 0]^T$), shown in Figure 3.7 and with no torque applied to the joints, comparing the angles of each of the joints as the simulated robot moved to a position with the arm hanging limp. A plot of the joint angles as found in Figure 3.8 shows that both models follow a nearly identical path when

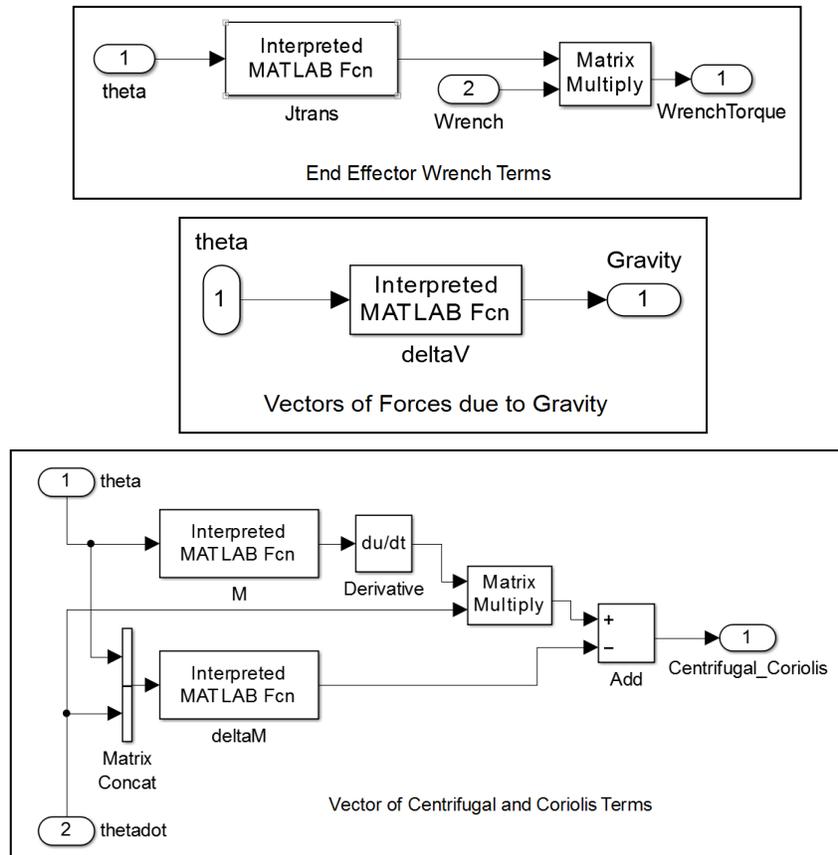


Figure 3.5: Simulink Model - Subsystem Blocks

moving to a limp position. Simple viscous damping was added to both models to ensure they remained stable as they 'fell' limp. The values of the viscous damping applied to each joint was determined through qualitative observation of the Mechanics Explorer simulation from the SimMechanics Model. The damping coefficients were adjusted until the simulated robot arm fell and oscillated in a reasonable manner.

Examining the plot of the joint angles, joints 1 and 3 reach an expected final position of 0 radians, with some initial oscillation as the arm 'falls'. The very small oscillation shown for joint 1 can be attributed to the coupling of the joints. Joint 2 comes to rest very near $-\pi/2$ radians as would be expected. Joint 4 comes to rest just short of $\pi/2$ radians as is expected due to the center of mass of the end-effector assembly being 3.45mm above the $Y_5 - Z_5$ plane.

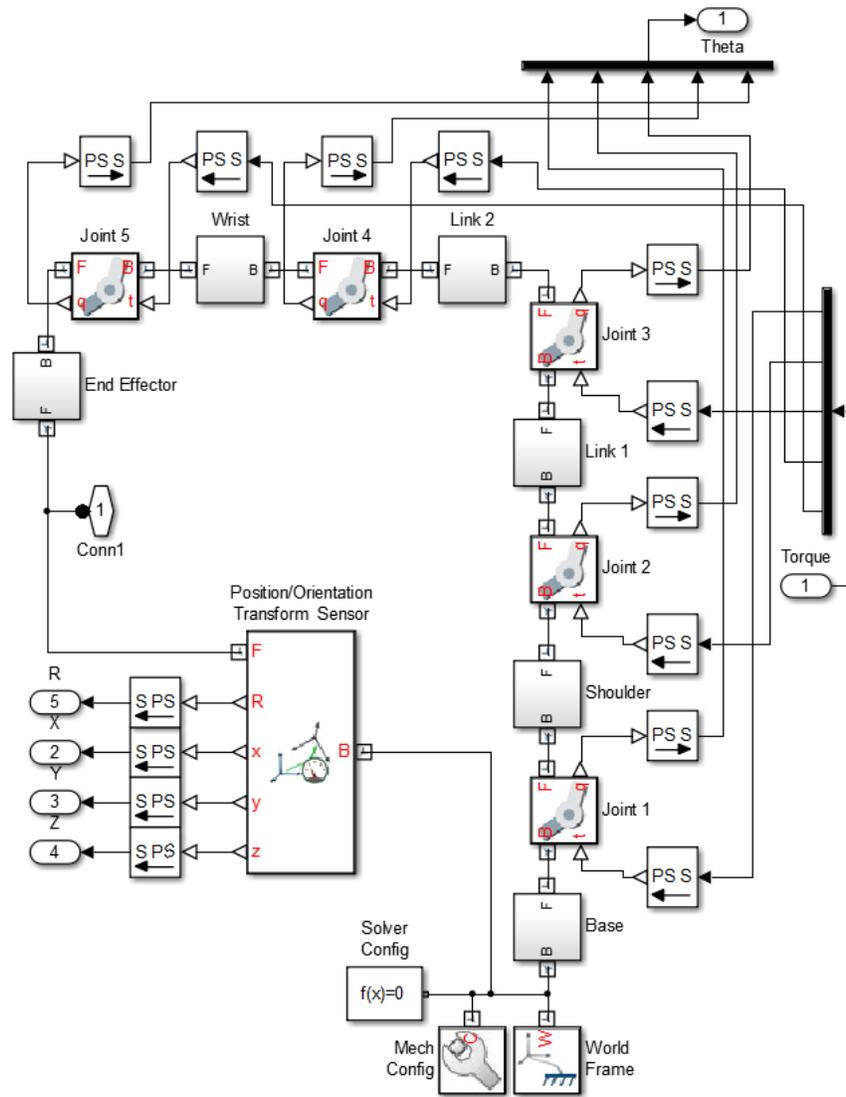


Figure 3.6: SimMechanics Model

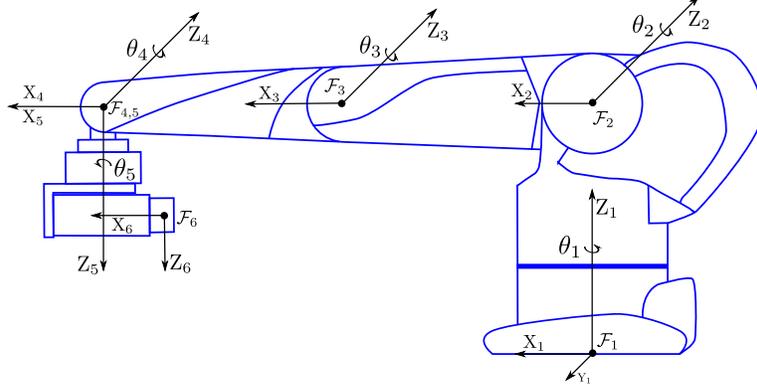


Figure 3.7: Zero-Angle Position

Joint 5 experiences a very small movement away from the zero position, which can be attributed to the center of mass of the end effector being slightly offset from the $X_5 - Z_5$ plane.

The second assessment of the models involved calculating the torque required to hold the arm at the zero-angle position shown in Figure 3.7, the same starting position as the previous test. The torque was determined to be $(\vec{\tau} = [0 \ 15.7 \ 5.2 \ 0.0348 \ 0]^T \text{Nm})$ using Eq. 3.37, the gravitational term of the dynamic equation. As would be expected, and shown in Figure 3.9 the response of the Mathematical Model is for all of the joints to stay at exactly 0 radians as the applied torque was calculated using the same equations and directly counters the gravitational forces within the model. The SimMechanics Model under constant applied torque shows joints 2 and 3 falling slightly below 0 radians, with joint 2 stabilizing and joint 3 recovering and moving slightly above 0 radians. Joint 4 does not fall below 0 radians as would be expected, but moves to a stable position just above 0 radians. The movements of joints 1 and 5 away from 0 are likely due to coupling effects and can be considered negligible. Extending the duration of this test to 120s, shows joint 2 continuing to fall very slowly, joint 3 climbing very slowly and joint 4 peaking and slowly returning towards 0 radians. As this is essentially an open loop control of the model, these very small movements away from the zero-angle position can be considered acceptable and the SimMechanics Model valid.

Comparison of the two CataLyst-5 models as developed, demonstrates that either is suitable for use in control system development and testing. As it is less computationally intensive and provides a means of determining the end-effector position and orientation, without employing the solution to the

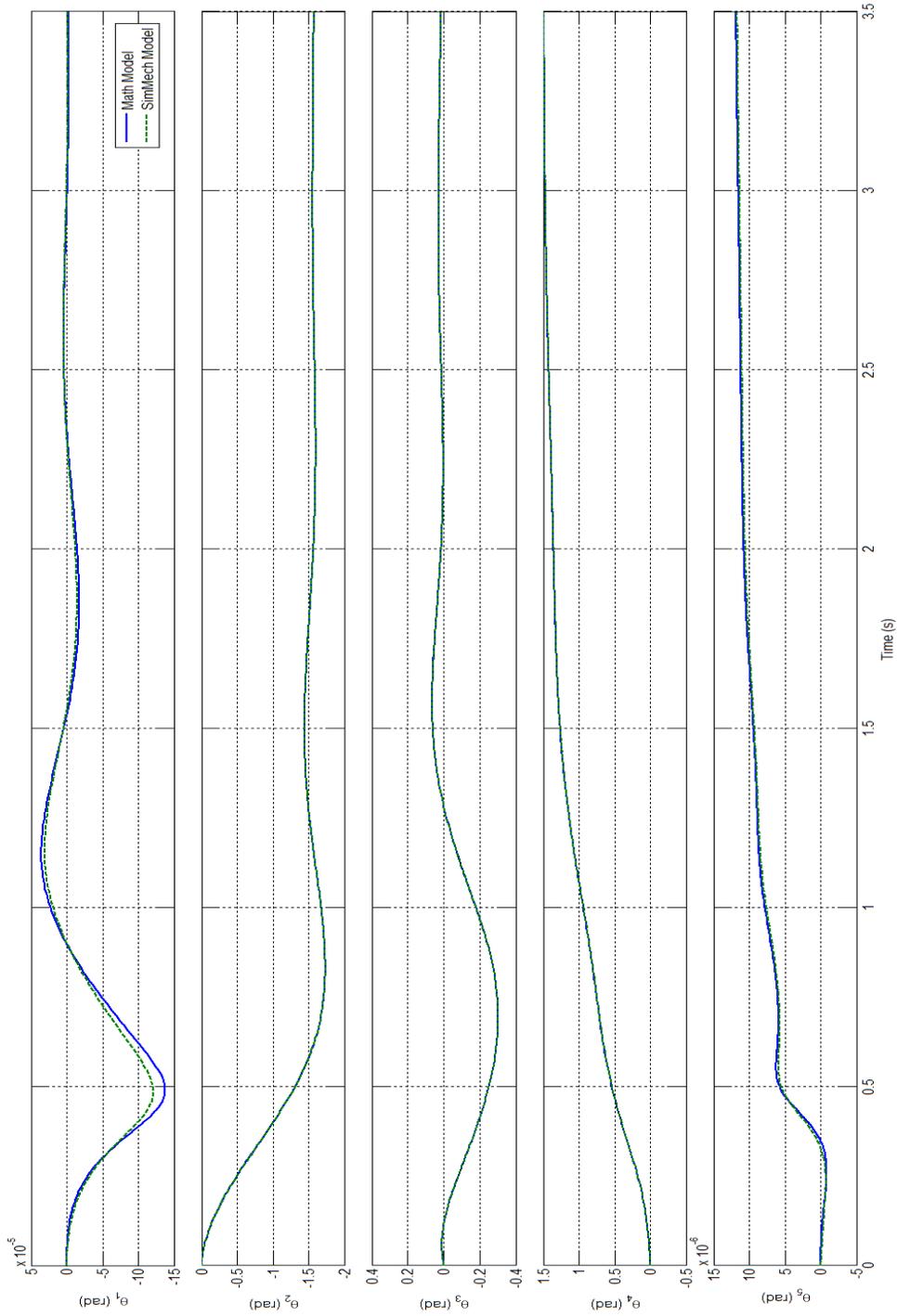


Figure 3.8: Model Comparison - Free Response

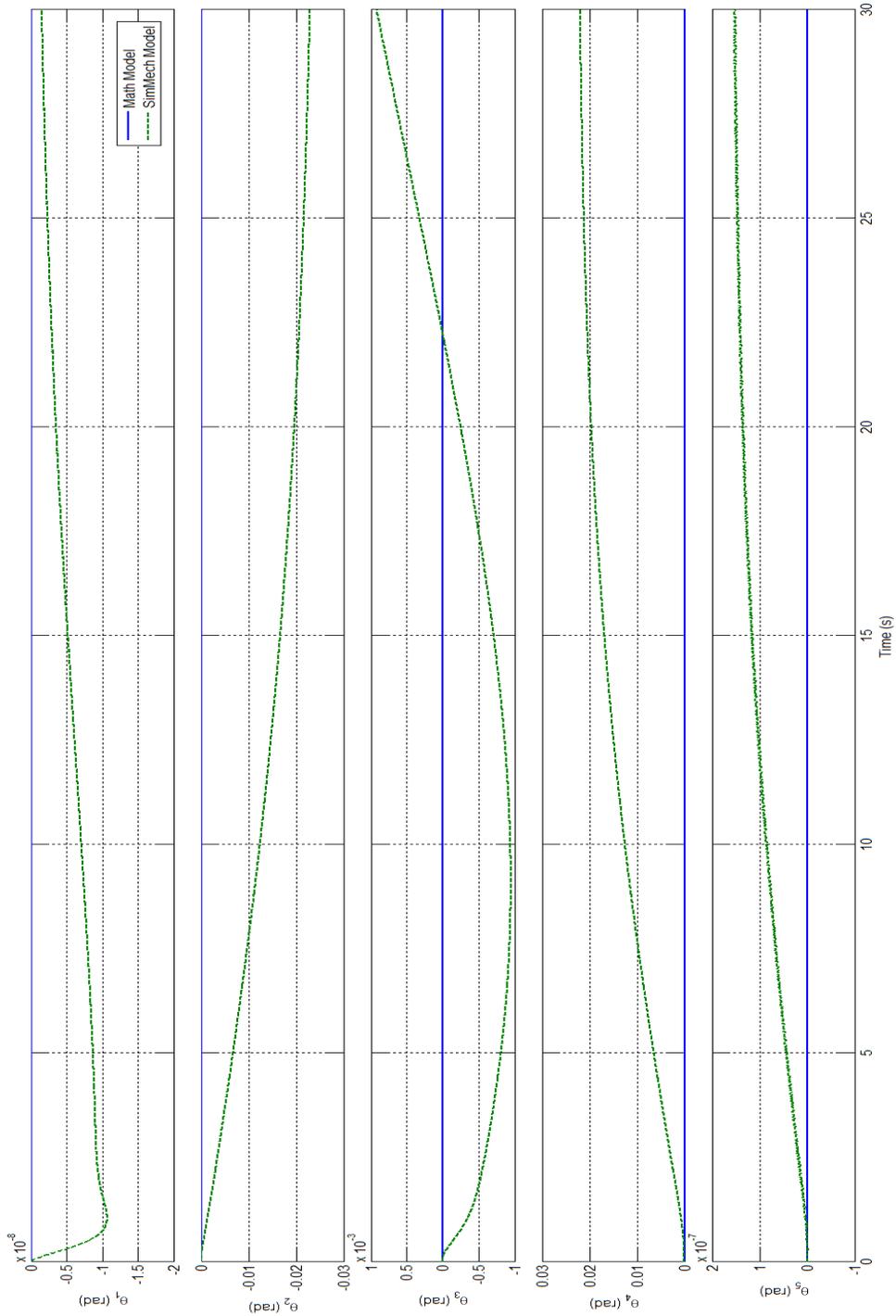


Figure 3.9: Model Comparison - Constant Torque

DKP, the SimMechanics model will be primarily used for all simulations. The elements of the mathematical model will be used within some of the control systems, and the model itself used as necessary to confirm results.

3.5 Contact Modelling

In addition to the advantages mentioned, the SimMechanics model is also the only model that is able to simulate contact with a work surface, a necessary requirement for testing position/force controllers prior to implementation on the actual robot. As shown in Figure 3.10, contact between the end effector of the robot and the drawing surface is simulated using a series of joints and masses with the surface force itself generated from a Linear Hard Stop block.

The model consists of a Mass block (Surface), translated and oriented with respect to the robot base, to provide a visual representation of the drawing surface. Attached to this surface with an initial position immediately below the end-effector in the ready position is a Planar Joint that allows for unhindered translation along an X-Y plane parallel to the base X-Y plane and rotation about the Z-axis. A Spherical Joint, providing three rotational degrees of freedom about the origins of the two frames to which it is attached, is connected to the planar joint to ensure that the end-effector orientation is not held fixed by the contact model. Lastly, a Prismatic Joint connects the spherical joint to the end-effector of the previously described SimMechanics robot model. The initial position of the prismatic joint is selected such that the origin of its follower will be coincident with the end-effector reference frame.

Not shown is an orientation block that ensures the Z-axes of the prismatic joint and end-effector are properly aligned. Small masses are added between each of the joints to avoid degenerate mass distribution errors when running the simulation. The token mass between the prismatic and spherical joints is 0.0001kg while the one between the spherical and planar joints has an inertia tensor of only diagonal elements of 0.0001kgm².

The contact force experienced by the model when it encounters the drawing surface is simulated using a number of blocks from the Simscape Mechanical Foundation Library in Simulink. The velocity of the prismatic joint is output to a Velocity Source block that provides an ideal velocity source between ports C and R based on the input signal. The ideal velocity signal creates movement in the Force Sensor block that outputs an ideal force (doesn't account for inertia, friction, etc.). This ideal force is based on the parameters found in the Hard Stop block, where an upper and lower position bound can be defined along with contact stiffness and damping values at each of these limits.

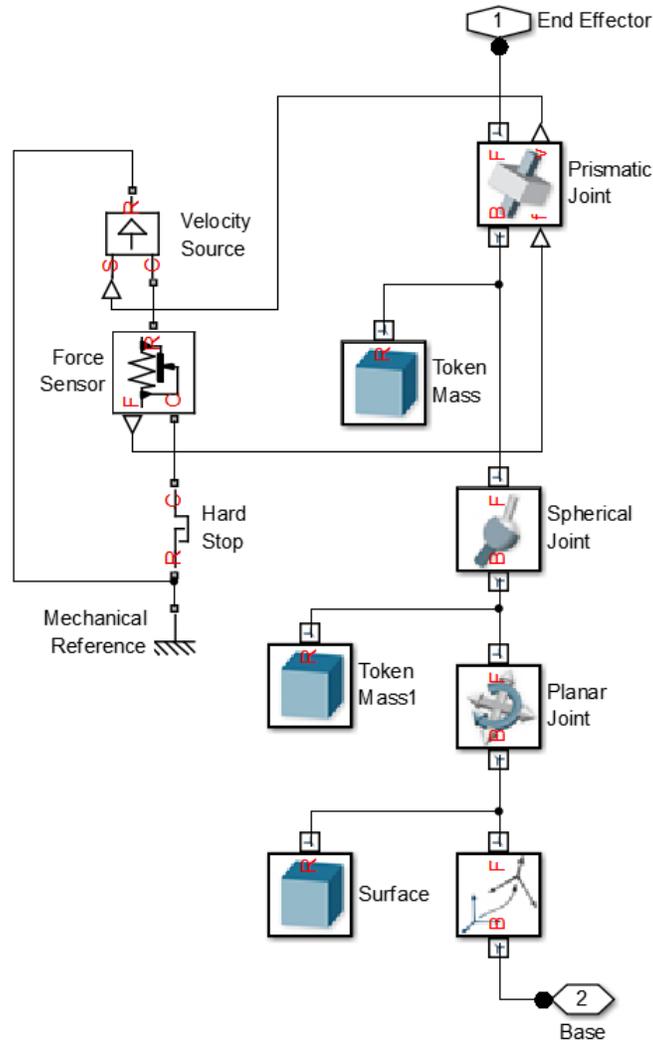


Figure 3.10: SimMechanics Contact Model

Two different methods were used to determine an appropriate contact surface stiffness and associated damping value. Based on the standard described in [62], most chalkboards consist of 0.4mm thick galvanized ASTM A526-80 sheet steel with Z275 zinc coating, all mounted on 11mm thick MDF. Based on this, it was assumed that the surface stiffness for a chalk board could be considered similar to the Young's Modulus for steel, which was determined to

be $200 \times 10^9 \text{N/m}$.

As this value was found to be too stiff for the simulation model to function, a second approach was taken that involved determining the stiffness of the load cell used under the drawing surface (detailed in Chapter 6). Measuring the force required to displace the cell a certain distance provided a surface stiffness of 55870N/m . Considering the contact model to be a second order critically damped system with a representative mass of 10kg yields a damping constant of 14950Ns/m .

Despite the significantly smaller stiffness values obtained using the second method, when conducting simulations the model would still fail to solve. After a significant number of simulations it was determined that the largest stiffness and damping values that could be used with the contact model were 1500N/m and 325Ns/m respectively. The significance of using surface stiffness values in simulation that are significantly less than would be expected in experimentation will be discussed in later chapters.

3.6 Conclusion

Details of the modelling of the CataLyst-5 that was conducted in support of control system development was presented. This modelling included the development of an equation based and concise solution to the direct kinematics of the manipulator, a summary of the solution the inverse kinematic problem and task specific solutions to the robot kinematics. The Lagrangian formulation of the dynamics of the CataLyst-5 was explained, followed by details of how these equations were used to create a Simulink based model of the manipulator for use in testing various control schemes. The model of the CataLyst-5 that was created in SimMechanics, as comparison and validation of the Simulink model, was explained. Lastly, it was detailed how the SimMechanics model was used as the foundation for modelling contact between the manipulator and the environment.

4 Control System Development

The following chapter explains the development of various control systems that can be used to simultaneously control both the position of the end-effector of the CataLyst-5 and the force the robot arm applies to the surroundings, and the trajectories that will be followed when implementing these controllers. Section 4.1 discusses path and trajectory planning and looks at when this planning is completed in joint or Cartesian space. The quintic polynomial that forms the basis of the trajectory generation algorithm used in this research is then presented. The following two subsections detail the Cartesian space and joint space trajectory generators that were developed. The final subsection details how multiple copies of the two different generators can be used in sequence to generate the time based joint and Cartesian trajectories required to trace pre-defined shapes on a drawing surface.

Sections 4.2 and 4.3 describe the development of the joint space position controllers, PD Independent and Joint Space Linearized. Optimized tuning of the gains for each of these controllers is then discussed. The development of the Cartesian Space Linearized position controller that forms the basis of the force controllers is explained in Section 4.5. Sections 4.6 and 4.7 detail the two position/force controllers that were developed for use with the CataLyst-5, the Hybrid Position/Force controller and the Position Based Force controller. A subsection following Hybrid control details the method used to detect contact with the drawing surface and control the transition from purely position to position/force control.

4.1 Path and Trajectory Planning

Path planning in robotics refers to the selection of the desired points that a robot must pass through and any obstacles that must be avoided. The path can be defined in any number of coordinate frames, but is typically defined for serial arm robots in a Cartesian coordinate frame where the origin coincides

with the base of the robot. Trajectory planning as defined in [25] refers to the generation of the, "...time history of position, velocity and acceleration for each degree of freedom..." required to achieve a particular path.

4.1.1 Joint Space versus Cartesian Space Planning

Generally path and trajectory planning schemes are either created in the joint space or the Cartesian space. Joint space planning methods can be used when only the initial and final position of the robot end effector are of concern (way points can be added for obstacle avoidance). This planning method determines the joint angles required for the starting point and the end point using the solution to the robot's inverse kinematic problem. Then a planning algorithm determines the achievable changes in angles as a function of time required to complete the movement. The motion of the end effector in Cartesian space is not generally well defined as each joint is actuated to either achieve its motion in the fastest or most efficient manner.

Cartesian space planning methods are used when the end-effector must follow a particular path. This method determines the time history of the end-effector's Cartesian position and orientation required to follow the desired path. This time history is then converted to the required joint angles in real time using the solution to the inverse kinematic problem. As the IKP must be solved along the path, this method is more computationally intensive than joint space planning methods. As such, joint space should be the default with Cartesian space planning only being used as necessary. As the tasks that will be performed by the CataLyst-5 involve tracing a defined pattern, Cartesian space planning will be the primary method used.

4.1.2 Trajectory Generator using a Quintic Polynomial

As the physical robot has mass and inertia, a function needs to be used to generate a smooth Joint or Cartesian space trajectory, that allows for achievable accelerations and velocities for the mechanism. As such a linear function between the start and end points is not realizable. In [25] and various other references discussing trajectory generation, a quintic polynomial function as described by Eq. 4.1, is presented to be used for this trajectory generation, where X can represent each of the elements of the position (and orientation) vector in Cartesian space, or the elements of the joint position vector, as a function of time.

$$X(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (4.1)$$

Taking the first and second derivatives of this equation gives the trajectory of the velocity, and acceleration respectively, which are given by:

$$\dot{X}(t) = a_1 + 2a_2t + 3a_3t^2 + 4a_4t^3 + 5a_5t^4 \quad (4.2)$$

$$\ddot{X}(t) = 2a_2 + 6a_3t + 12a_4t^2 + 20a_5t^3 \quad (4.3)$$

Combined with the following six constraints for initial (t_i) and final (t_f) time, which assume the manipulator starts and stops at each desired waypoint,

$$\begin{aligned} X(t_i) &= X_{start} \\ X(t_f) &= X_{end} \\ \dot{X}(t_i) &= \dot{X}(t_f) = 0 \\ \ddot{X}(t_i) &= \ddot{X}(t_f) = 0 \end{aligned}$$

Eq. 4.1 can be solved for a_j to obtain the following, where $T = t_f - t_i$ is the duration, i and f indicate the values of each parameter at start and end times for each movement.

$$\begin{aligned} a_0 &= X_i \\ a_1 &= \dot{X}_i \\ a_2 &= \frac{\ddot{X}_i}{2} \\ a_3 &= \frac{20X_f - 20X_i - (8\dot{X}_f + 12\dot{X}_i)T - (3\ddot{X}_i - \ddot{X}_f)T^2}{2T^3} \\ a_4 &= \frac{30X_i - 30X_f + (14\dot{X}_f + 16\dot{X}_i)T - (3\ddot{X}_i - 2\ddot{X}_f)T^2}{2T^4} \\ a_5 &= \frac{12X_f - 12X_i - (6\dot{X}_f + 6\dot{X}_i)T - (\ddot{X}_i - \ddot{X}_f)T^2}{2T^5} \end{aligned}$$

4.1.3 Cartesian Space Trajectory Generator

Figure 4.1 shows a Simulink implementation for a trajectory planner that makes use of the quintic polynomial defined by Eq. 4.1 to generate the required positions and velocities to move in a straight line between start and

end positions defined in Cartesian space. Other inputs to the planner include a signal of when to begin this particular part of the trajectory and at what percentage of maximum speed the movement should be conducted. The inputs are passed to various blocks with the following functions. The Cart Quintic Polynomial block in Figure 4.1 incorporates Eqs. 4.1 and 4.2 to calculate Cartesian positions and velocities required to provide a smooth, straight line trajectory for the end-effector when moving from the start position to end position. The time input for these functions (starting at $t = 0$) is provided from the Switched Clock block that contains an integrator that switches from integrating 0 to integrating 1 when the begin signal is received. This triggering function for the Cartesian trajectory planner is required so that multiple planners can be used consecutively for movements between consecutive waypoints.

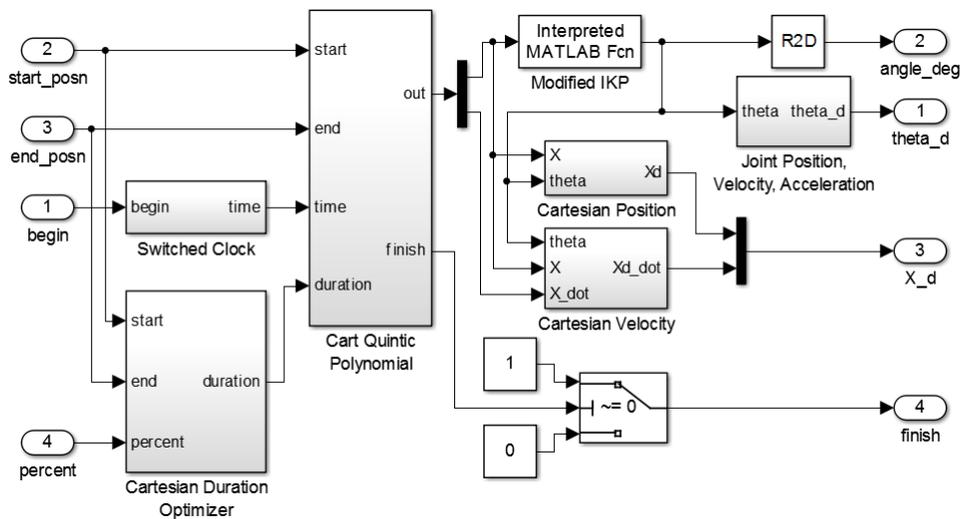


Figure 4.1: Cartesian Space Trajectory Planner

The duration for the quintic polynomial is determined by the Cartesian Duration Optimizer block which divides the distance the robot is required to travel, along each of the axes, by a user defined percentage of the maximum Cartesian velocities and outputs the longest duration of the three axes. With the same duration used for the movements along each of the Cartesian axes, the end effector will follow a straight line path between the start point and the end point. The maximum velocities are based on those used in Quanser's Cartesian position controller of 100mm/s for each axis, as found in Table 8 of

[63].

As the primary output of the Cartesian Quintic Polynomial block consists of only the end-effector position and velocity coordinates, required to create a straight line between the start and end positions, additional blocks are required to determine orientation and angular velocity of the end-effector. The Cartesian Position block uses the simple angle method of representing orientation as described in 3.1.1 to determine the 3×1 orientation vector coinciding with each set of joint angles. The orientation vector is concatenated with the position vector to provide a 6×1 vector (\vec{X}), as defined in Eq. 4.4, which represents the desired orientation (roll, pitch, yaw) and position required for the desired movement.

$$\vec{X} = \begin{bmatrix} r \\ \beta \\ \gamma \\ X \\ Y \\ Z \end{bmatrix} \quad (4.4)$$

The Cartesian Velocity block in Figure 4.1 determines the angular velocity of the end-effector that coincides with the Cartesian velocity for each movement. As the desired task fixes the end effector in roll and pitch, the angular velocity of the end effector is equivalent to the angular velocity of joint 1, which can be calculated using Eq. 4.5 where V_R is the radial velocity of the end-effector, R is the radial position of the end-effector and X, Y, \dot{X}, \dot{Y} are the Cartesian positions and velocities. This calculated yaw speed is concatenated with zeros for the other elements of the angular velocity and the Cartesian velocities to provide a 6×1 vector of linear and angular velocities.

$$\dot{\theta} = \frac{V_R}{R} = \frac{\dot{Y} \cos \theta_1 - \dot{X} \sin \theta_1}{\sqrt{X^2 + Y^2}} \quad (4.5)$$

To provide the time history of the joint angles as an overall output and also as input to the blocks defined above, the Modified IKP block determines the joint angles in radians using the simplified version of the solution to the IKP described in section 3.2.1. The joint angles are also input to the Joint Position, Velocity and Acceleration block, where the first derivative is taken to provide the joint velocities. The joint accelerations, which are only required for the Joint Space Linearized Controller, are provided as a 5×1 matrix of zeros. The position, velocity and acceleration matrices are concatenated to provide an overall vector of the desired trajectory in joint space. The joint angles from

the modified IKP block are also converted to degrees which are provided as an additional output from the trajectory planner. The other output from the Cartesian Quintic Polynomial block provides a signal of '1' when the desired end point has been reached along all Cartesian directions. This signal is used as a trigger to initiate subsequent trajectory planning blocks.

4.1.4 Joint Space Trajectory Generator

The completed Cartesian Space Trajectory Planner was used in creating the Joint Space Trajectory Planner found in Figure 4.2. The key difference between the two planners is that with the joint planner, the start and end positions are converted into joint angles. Time dependent joint angles between these start and end poses are created in the Cartesian Quintic Polynomial block which was modified to generate 5 joint trajectories vice 3 Cartesian trajectories. The block uses Eqs. 4.1, 4.2 and 4.3 to create an output signal consisting of a vector of the joint positions, velocities and accelerations required to achieve the desired movement.

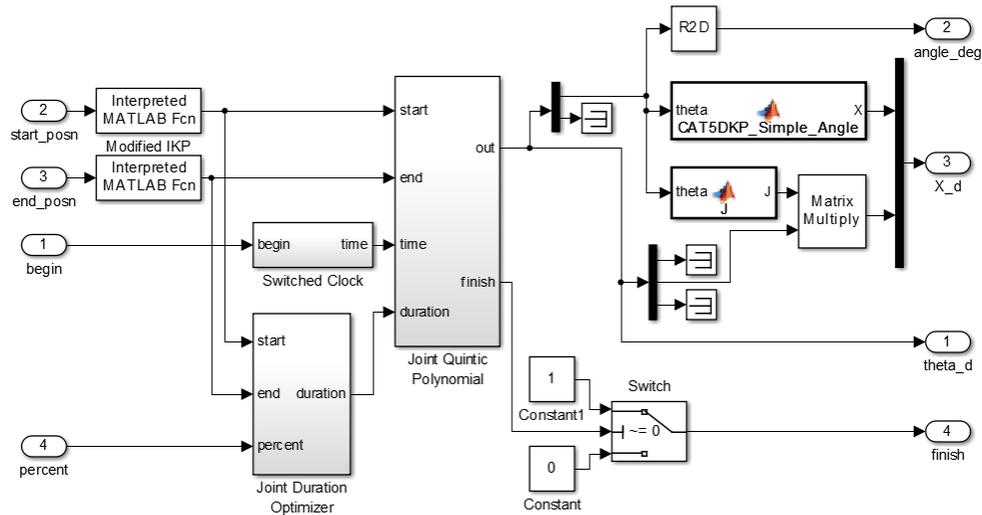


Figure 4.2: Joint Space Trajectory Planner

The joint angles are converted to Cartesian position and orientation trajectories using the position vector from Eq. 3.7 concatenated with the simple angle orientation representation discussed in Section 3.1.1. The Cartesian and angular velocities coinciding with the joint trajectory are determined by multiplying the joint velocities by the manipulator Jacobian in Eq. 3.39. The

final change that was made in creating the joint space planner was that the duration optimizer was modified to divide the angular distance each joint is required to move by a user defined percentage of the maximum joint velocities used in Quanser's joint position controller, as found in Table 7 of [63].

4.1.5 Trajectory Generation of Various Shapes

With both Cartesian and joint space trajectory generators built, Simulink models were created to generate a variety of shapes to be traced by the CataLyst-5. These shapes include a straight line, a rectangle, a 5 point star, a circle and a spiral trajectory. The first three shapes make use of the trajectory planners as designed. The trajectory generators for both the circle and the spiral require a modification of the Cartesian Trajectory Planner block as detailed in Appendix D.

As example of the Cartesian and joint space generators being used to draw various shapes, the Simulink model for the line trajectory is shown in Figure 4.3. It makes use of a joint space planner to move the arm from the ready position to a point just above the drawing plane. A Cartesian space planner is used to move the end-effector slowly (5% of max speed) along the negative Z-axis until contact is made with the drawing plane. The joint and Cartesian space trajectories of both of these first two movements are stored in .mat files as arrays which can be used with any of the controllers either in simulation or experimentation.

Once contact has been made with the drawing plane, a series of Cartesian Trajectory Planner blocks are used to create the necessary trajectory for the desired shape. As shown in Figure 4.3, a single block is used to generate a .mat file with the necessary data. Not shown in Figure 4.3, but following all of the shape trajectory blocks is a third set of blocks that creates a .mat array from a Cartesian block that slowly raises the end-effector along the Z-axis to a point above the drawing plane and a Joint block that moves the arm from this point back to the ready position. Various time delays are included between all trajectory planning blocks to account for any time variation between the desired movement and the actual movement of the robot.

Three separate arrays of trajectory data are created for each of the desired shapes to allow for a transition between purely position based control, for approach and departure movements, and position/force control once in contact with the drawing plane.

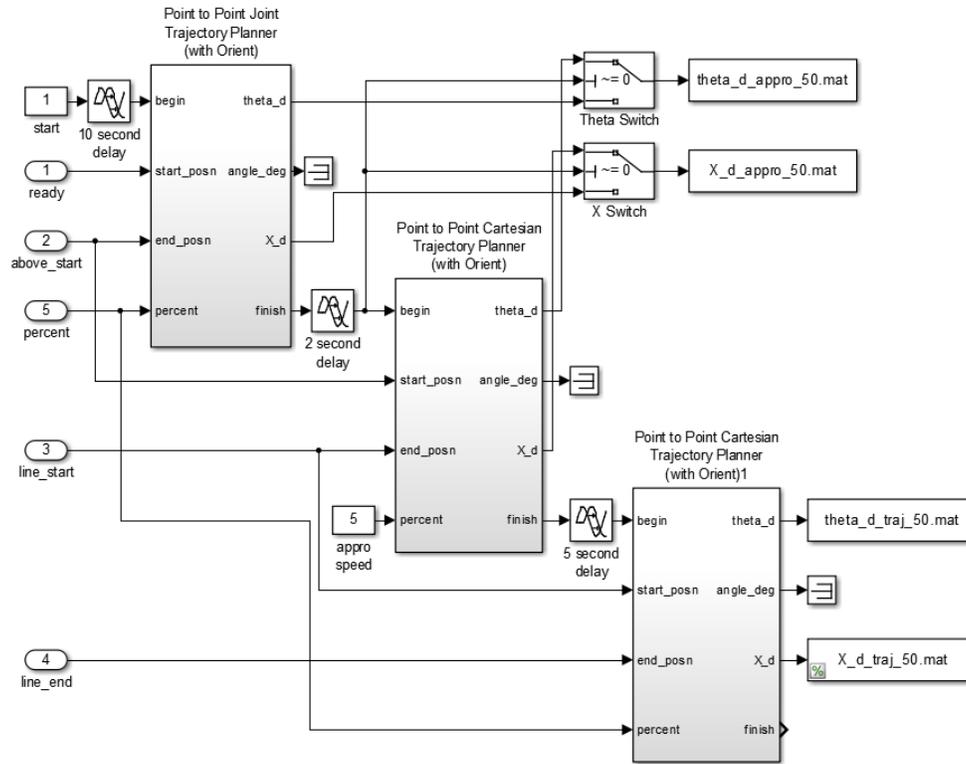


Figure 4.3: Line Draw Trajectory Generator

4.2 PD Independent Joint Error Control

As it is the most widely used method of controlling serial arm manipulators in industry, a joint proportional and derivative (PD) independent controller was developed as a baseline position controller for the CataLyst-5 robot. A block diagram depicting the PD control system for an individual joint is shown in Figure 4.4 where k_p and k_d are the proportional and derivative gains respectively, K_t is the motor torque constant, K_g is the joint gear ratio and K_{enc} is the joint encoder ratio. The filter block consists of a low pass filter that rejects frequencies greater than ω and derivative block that differentiates the filtered error signal. The saturation block ensures that large currents do not reach the amplifiers within the C500D power supply. The portion of the block diagram enclosed in the dashed red box is the controller created in Simulink, while the other blocks are the physical components of the power supply and

robot arm.

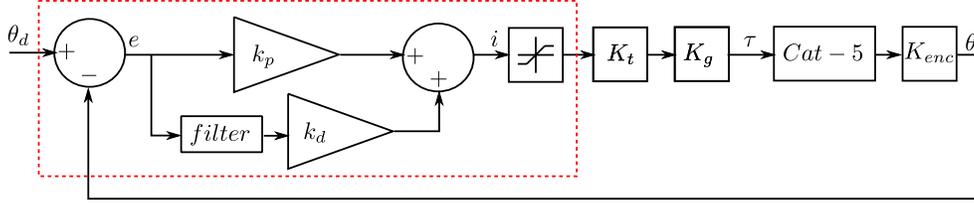


Figure 4.4: PD Independent Joint Control - Block Diagram

To determine appropriate proportional and derivative gains for use with the control system, a representative transfer function for the CataLyst-5 was determined. The dynamics of the CataLyst-5 robot were represented using solely the diagonal elements of the generalized inertia tensor (M_{dia}), neglecting the remaining terms of the system dynamics found in Eq. 3.35. The magnitude of M_{dia} was determined for each joint in the robot arm configuration that would result in the largest inertia value. The transfer function for each joint of the CataLyst-5 is found in Eq 4.6

$$\begin{aligned}
 & \mathcal{L}[\tau = M_{dia}\ddot{\theta}] \\
 \Rightarrow T(s) &= M_{dia}\Theta(s)s^2 \\
 \frac{\Theta(s)}{T(s)} &= \frac{1}{s^2M_{dia}} \tag{4.6}
 \end{aligned}$$

In determining the transfer function for the remainder of the control system, the dynamics of the saturation block, the joint encoder block and the low pass filter were omitted. The open loop transfer function for the system is as follows.

$$G(s) = (k_p + k_d s) K_t K_g \left(\frac{1}{s^2 M_{dia}} \right)$$

The derivation of the closed loop transfer function for the system is found in Eq. 4.7.

$$\begin{aligned}
 T(s) &= \frac{\Theta(s)}{\Theta_d(s)} = \frac{G(s)}{1 + G(s)} \\
 &= \frac{K_t K_g k_d s + K_t K_g k_p}{M_{dia} s^2 + K_t K_g k_d s + K_t K_g k_p} \tag{4.7}
 \end{aligned}$$

As shown in the denominator of Eq. 4.7, the dynamics of each independent joint controller can be considered as a second order linear system for gain tuning purposes. It was assumed that zero overshoot would be desirable for a serial arm robot controller, thus $\zeta = 1$ was selected to create a critically damped system. This meant that the system would have two identical negative real roots equivalent to the natural frequency (ω_n) of the system. Using this information, Eq. 4.8 can be used to determine the initial values of the proportional and derivative gains where (ω_n) is determined based on a 2% settling time (T_s).

$$\left(s + \frac{4}{T_s}\right)^2 = s^2 + \frac{K_t K_g k_d}{M_{dia}} s + \frac{K_t K_g k_p}{M_{dia}} \quad (4.8)$$

Solving the equation above gives the following equations for k_p and k_d .

$$k_p = \frac{16M_{dia}}{(\zeta T_s)^2 K_t K_g} \quad (4.9)$$

$$k_d = \frac{8M_{dia}}{(\zeta T_s) K_t K_g} \quad (4.10)$$

The values and associated robot configuration used in calculating k_p and k_d are found in Table 4.1. The configuration angles are the joint angles used to calculate M_{dia} , the maximum inertia value for each joint. The initial gains were calculated with a settling time of 0.2s, with further tuning based on simulation using the SimMechanics model.

Table 4.1: Parameters Used to Calculate PD Independent Joint Control Gains

Joint	Config Angle	M_{dia} Nm	K_t Nm/A	K_g	k_p	k_d
1	[0 0 0 $\pi/2$ 0]	0.885	0.0701	72	70.3	7.03
2	[0 0 0 $\pi/2$ 0]	0.776	0.0701	72	61.5	6.15
3	[0 $\pi/2$ - $\pi/2$ $\pi/2$ 0]	0.194	0.0701	72	15.4	1.54
4	[0 $\pi/2$ - $\pi/2$ $\pi/2$ 0]	0.0212	0.0701	19.6	3.90	0.390
5	[0 $\pi/2$ - $\pi/2$ $\pi/2$ 0]	0.00172	0.0701	9.8	1	0.1

4.3 Joint Space Linearized Control

The second position controller that was developed for use with the CataLyst-5 is a joint space linearized controller as described in [25], known as computed

torque control in [14] and [31] and inverse dynamics control in [15]. Using the system dynamics of the serial arm robot described by Eq. 3.35, and neglecting the wrench term (\vec{w}), the estimated joint torques ($\vec{\tau}$) required for any movement can be found using the following equation.

$$M(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \vec{v}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) + \vec{g}(\boldsymbol{\theta}) = \vec{\tau} \quad (4.11)$$

Linearized control as described in [25] proposes the following control law involving the dynamic equations,

$$\vec{\tau} = M(\boldsymbol{\theta})\vec{\tau}' + \vec{v}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) + \vec{g}(\boldsymbol{\theta}) \quad (4.12)$$

which when combined with Eq. 4.11 and given that the matrix M is invertible gives the following which shows that with a control input of $\vec{\tau}'$ the system is now linear and decoupled.

$$\ddot{\boldsymbol{\theta}} = \vec{\tau}' \quad (4.13)$$

With the system linearized and decoupled, $\vec{\tau}'$ can be selected to provide appropriate control of a linear second order system, with the control law proposed in [25] presented below, where $\boldsymbol{\theta}_d$ is the desired joint angle, k_d and k_p are diagonal matrices of derivative and proportional control gains, respectively and $\vec{e} = \boldsymbol{\theta}_d - \boldsymbol{\theta}$.

$$\vec{\tau}' = \ddot{\boldsymbol{\theta}}_d + k_d\dot{\vec{e}} + k_p\vec{e} \quad (4.14)$$

The control law defined above was used to create a joint space linearized controller in Simulink, the block diagram depicting this controller is shown in Figure 4.5. In the diagram, the blocks $M(\boldsymbol{\theta})$ and $\vec{v}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) + \vec{g}(\boldsymbol{\theta})$ provide the inverse dynamics ahead of the system causing the linearization and decoupling. The portion of the block diagram enclosed in the dashed red box is the linear controller. The elements of the block diagram in the green box titled 'For Implementation' will be required when the controller, which outputs torque values, is used on the actual robot where control currents are input to the C500D power supply.

The initial gains for the system can be determined by combining Eqs. 4.13 and 4.14, using $\vec{e} = \boldsymbol{\theta}_d - \boldsymbol{\theta}$ and setting the resulting equation equal to 0, with any disturbances encountered by the robot arm including any unmodelled dynamics such as friction accounted for in the error term.

$$\ddot{\vec{e}} + k_d\dot{\vec{e}} + k_p\vec{e} = 0 \quad (4.15)$$

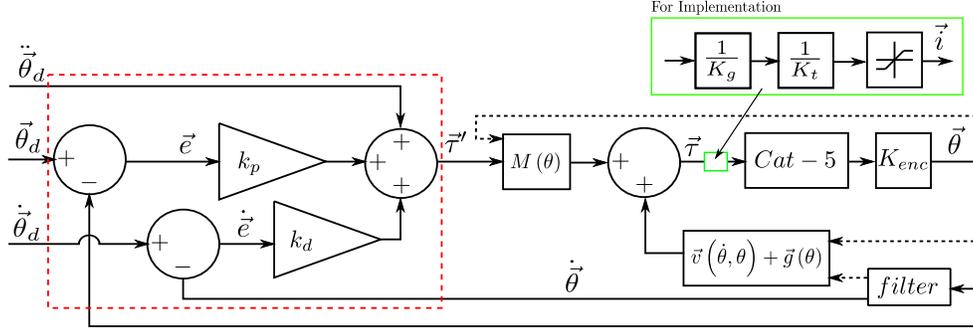


Figure 4.5: Linearized Joint Space Control - Block Diagram

Considering the fact that the error dynamics in Eq. 4.15 are uncoupled, taking the Laplace transform of any error channel results in the following, where K_p and K_d are the gains for the error channel being investigated.

$$s^2 E(s) + K_d s E(s) + K_p E(s) = 0 \quad (4.16)$$

From Eq. 4.16, the characteristic equation is:

$$s^2 + K_d s + K_p = 0 \quad (4.17)$$

which can be compared directly to the characteristic equation of a second order system ($s^2 + 2\zeta\omega_n s + \omega_n^2$) with $\zeta = 1$ selected to create a critically damped system with no overshoot, the initial proportional and derivative gains for each channel can be calculated using the equations below.

$$\begin{aligned} K_p &= \omega_n^2 \\ K_d &= 2\omega_n \end{aligned}$$

The initial gains were calculated with a 2% settling time of 0.2s, resulting in values of $K_p = 400$ and $K_d = 40$ with further tuning based on simulation using the SimMechanics model.

4.4 Gains Tuning

With the aim of optimizing the two controllers described in sections 4.2 and 4.3 an iterative process was used that involved moving each of the joints individually using a step input through the joint trajectory generator described

in section 4.1 and adjusting the gains and the maximum velocities used with the trajectory generators. The intent was to have the simulated robot conduct the movement in as short of time as possible, while not exceeding 90% of the maximum applied torque at the joint and keeping the joint error and overshoot to a minimum.

The initial maximum joint velocities, used with the joint trajectory generators, were determined by comparing the maximum design joint velocities from [60], the maximum velocities used by Quanser with their controller [64] and the maximum velocities that were determined experimentally using the CataLyst-5 in closed architecture control mode. The maximum torque at each joint was determined by multiplying the maximum control current (3.5A) by the motor torque constant and the gear ratio for each of the respective joints. The maximum velocities and the 90% of maximum torque value for each of the joints can be found in Table 4.2.

Table 4.2: CataLyst-5 Maximum Joint Velocities (from [64])

Joint	Thermo-Fisher Max Velocity $^{\circ}/s$	Quanser Max Velocity $^{\circ}/s$	Experimental Max Velocity $^{\circ}/s$	90% Max Torque Nm
1	210	50	140	15.89
2	210	25	94	15.89
3	210	25	92	15.89
4	551	500	228	4.33
5	1102	250	443	2.16

The detailed description and results of the iterative gains tuning process can be found in Appendix E, with summaries of the results presented below in Tables 4.3 and 4.4.

Table 4.3 summarizes all of the gains, maximum joint velocities that were determined for the PD Independent controller. The table also summarizes the maximum errors, overshoots and applied torques as found for each of the joints when the step inputs were applied.

Table 4.4 summarizes all of the gains, maximum joint velocities that were determined for the joint space linearized controller. The table also summarizes the maximum errors, overshoots and applied torques as found for each of the joints when the step inputs were applied.

In general, the PD Independent joint controller is capable of moving the individual joints at a faster speed than the linearized joint space controller, but with a trade-off of steady state errors for the joints that support the mass

Table 4.3: PD Independent Controller Gains, Maximum Joint Velocities & Errors

Joint	Gains		Max Joint Velocity °/s	Maximum Error rad	Maximum Overshoot rad	Maximum Torque Nm
	P	D				
1	230	50	160	0.1242	7.19e-4	14.96
2	61.5	13	90	1.346	1.09e-5	17.65*
3	175	13	120	0.01447	1.11e-5	13.32
4	40	1.8	150	0.08374	4.71e-4	4.64
5	40	2	180	0.07743	1.35e-7	2.13

*Maximum control current/torque applied.

Table 4.4: Joint Space Linear Controller Gains, Maximum Joint Velocities & Errors

Joint	Gains		Max Joint Velocity °/s	Maximum Error rad	Maximum Overshoot rad	Maximum Torque Nm
	P	D				
1	1600	80	135	0.01063	0	16.04
2	400	40	90	1.299	2.19e-6	17.65*
3	30000	400	120	0.01196	1.73e-6	15.37
4	10000	200	120	0.08276	1.43e-6	4.65
5	12000	250	140	0.1007	1.0e-5	2.07

*Maximum control current/torque applied.

of the robot links and end-effector assembly. This steady state error could potentially be reduced or eliminated with the application of feed-forward gravity compensation. The Joint Space Linearized controller better compensates for steady state errors due to the mass of the robot links, but requires higher actuation torques for the individual joint movements necessitating slower maximum joint speeds. In addition, when conducting movements of individual joints with the linearized controller, significant movements are observed in some of the other joints resulting in unacceptable errors.

4.5 Cartesian Space Linearized Control

The final position controller that was developed for use with the CataLyst-5 is a Cartesian Space Linearized Controller that, similar to the controller

defined in Section 4.3, makes use of the dynamic equations of the mechanism to both decouple the manipulator and linearize the control. When linearizing in Cartesian space, the dynamics of the manipulator must be defined in Cartesian space as shown in the following equation from [25], where $\vec{\mathcal{W}}$ is an imaginary vector of torques and forces that when applied to the end-effector will provide the required orientation and position movements.

$$\vec{\mathcal{W}} = M_X(\boldsymbol{\theta})\vec{\mathcal{W}}' + V_X(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) + G_X(\boldsymbol{\theta}) \quad (4.18)$$

The elements of the dynamic equation are converted from joint space using the following equations.

$$\begin{aligned} M_X(\boldsymbol{\theta}) &= J^{-T} M(\boldsymbol{\theta}) J^{-1} \\ \vec{V}_X(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) &= J^{-T} \left(\vec{v}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) - M(\boldsymbol{\theta}) J^{-1} \dot{J} \dot{\boldsymbol{\theta}} \right) \\ \vec{G}_X(\boldsymbol{\theta}) &= J^{-T} \vec{g}(\boldsymbol{\theta}) \end{aligned}$$

As the Cartesian space dynamics require taking the inverse of the non-square Jacobian, the Moore-Penrose pseudo-inverse as shown below, and defined in [25], is used.

$$J^* = (J^T J)^{-1} J^T$$

With the dynamics of the manipulator defined in Cartesian space, the system can be linearized and decoupled with $\vec{\mathcal{W}}'$ selected to provide appropriate control of what can now be considered a linear second order system. The vector required to control the linearized mechanism can be determined using the equation below, where \mathbf{X}_d is the vector of desired end-effector position and orientation, k_d and k_p are matrices of derivative and proportional control gains respectively and $\vec{e}_x = \mathbf{X}_d - \mathbf{X}$. Eq. 4.19 differs from Eq. 4.14 used with the joint space linearized controller in that the desired Cartesian acceleration is neglected. This term was omitted, as a reliable means of determining this value from the joint position encoder readings could not be determined. Omitting the acceleration term was deemed acceptable as it was observed that the term did not contribute significantly to the control signal in the joint linearized controller for the non-aggressive movements being used in this research.

$$\vec{\mathcal{W}}' = k_d \dot{\vec{e}}_x + k_p \vec{e}_x \quad (4.19)$$

With $\vec{\mathcal{W}}'$ defined, the controller shown in block diagram form at Figure 4.6 could be created in Simulink. In the diagram, the blocks $M_X(\boldsymbol{\theta})$ and

$\vec{V}_X(\dot{\theta}, \theta) + \vec{G}_X(\theta)$ provide the inverse dynamics ahead of the system, causing the linearization and decoupling. The portion of the block diagram enclosed in the dashed red box is the controller. As the output of the controller (\vec{W}) consists of an equivalent set of torques and forces applied at the end-effector, the control signal is multiplied by the transpose of the Jacobian to convert it to a vector of joint torques. The elements of the block diagram in the green dashed line box titled 'For Implementation' will be required when the controller, which outputs torque values, is used on the actual robot where control currents are input to the C500D power supply.

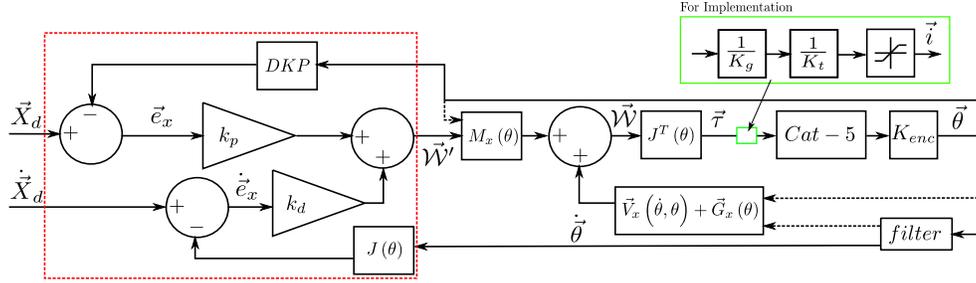


Figure 4.6: Cartesian Space Linearized Control - Block Diagram

Using the same methods as were used in Section 4.3, the characteristic equation of the closed loop system is equated directly to that of a second order system. Again, $\zeta = 1$ is selected to create a critically damped system with no overshoot and the initial proportional and derivative gains can be calculated using the equations below.

$$\begin{aligned}
 k_p &= \omega_n^2 \\
 k_d &= 2\omega_n
 \end{aligned}$$

The initial gains were calculated with a 2% settling time of 0.2s, resulting in values of $k_p = 400$ and $k_d = 40$.

4.6 Hybrid Position/Force Control

The Cartesian Space Linearized controller developed in the previous section serves as the foundation of the Hybrid Position/Force controller found in block diagram form in Figure 4.7.

The position/force controller adds an additional control loop that measures the wrench vector (\vec{w}), as defined in Eq. 3.38, applied by the robot on the

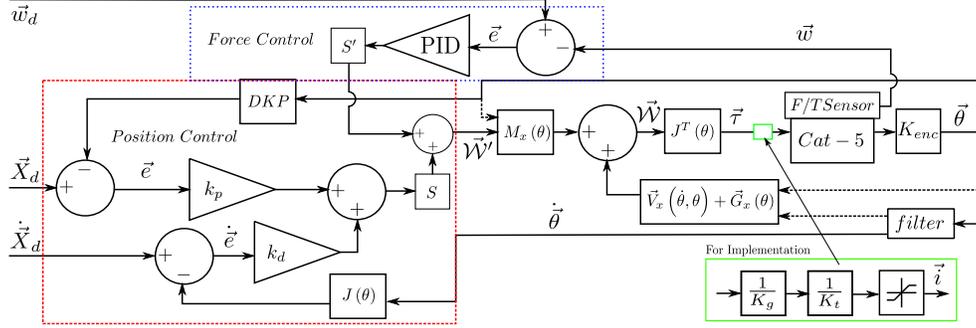


Figure 4.7: Hybrid Position/Force Control - Block Diagram

environment and compares it to the desired wrench (\vec{w}_d). The error between the signals is multiplied by appropriate gains to create a control signal in Cartesian space consisting of the required torques and forces (\vec{W}') applied at the end effector to adjust the applied wrench. This control signal is the same as that created by the Cartesian space linearized controller, thus the two can be added and applied to the decoupled and linearized manipulator.

In the application used in this thesis, the force control loop makes use of a Proportional, Integral and Derivative controller to adjust the end-effector position along the Z-axis based on the measured applied force. The gains used were determined numerically using the SimMechanics model and adjusted as necessary when applying the controller to the actual robot. The initial gain values determined from simulation are $k_{p_f} = 16$, $k_{i_f} = 1.5$ and $k_{d_f} = 0.5$.

As the signals from the position and force control loops will conflict when the manipulator motion is constrained, some method is required to select only desired portions of from each loop to pass to the linearized manipulator. This is accomplished by selecting the control signal from the force controller along the axes that force (or torque) is to be applied to the environment, with the remaining axes controlled using position control. The control signal selection is accomplished with the 6×6 diagonal selection matrices S and S' , where a 1 on the diagonal indicates the signal is to be selected along the associated axis and a 0 cancels the signal. As shown in Eq. 4.20 the selection matrices are complimentary in that where a 1 occurs in the S matrix, a 0 must occur in the same position in the S' matrix.

$$S + S' = I_{6 \times 6} \quad (4.20)$$

As the position/force control being implemented in this thesis involve controlling the force along the Z-axis, the S and S' matrices are as follows.

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad S' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

4.6.1 Force Detection and Control Method Switching

As the approach to and departure from the contact surface is accomplished using only position based control, a means of switching to position/force control is required. Also, initial contact with the surface needs to be accomplished in a controlled fashion that neither damages the robot, nor the surface. This contact detection and switching is accomplished in two steps. The trajectory of the robot approaches the contact surface with a very slow speed. When force is detected by the sensor, a Triggered Subsystem captures the Z position at that instant and passes this value as the new desired Z value, replacing the stored values that have the end effector continuing to move in the negative Z direction. This captured Z position value will remain the desired Z value until the trajectory input into the controller is greater than the captured value, indicating that the departure portion of the trajectory is underway. The triggered subsystem also passes a constant value of 0 for the desired Z velocity signal into the derivative portion of the controller.

The detection of contact also initiates the switching from position only to position/force control using the system shown in Figure 4.8. The detection event activates the force controller, using the force measured by the sensor as input, and configures a switch block to pass this control signal where it is added to the control signal along the Z-axis from the position controller. The contact detection also configures a switch block to pass a value of 0 vice the actual control signal from the position controller. There is a delay between adding the force control signal and removing the position control signal to allow the system to normalize at the desired force while the more stable and dominant position controller is still active. When the desired shape trajectory is complete a step block sends a signal switching back to purely position based control with the captured Z position as the desired value until the end-effector is required to depart the contact surface.

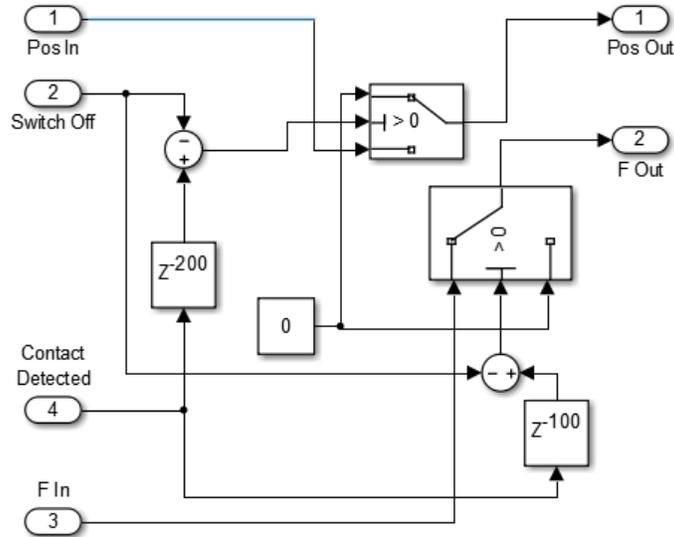


Figure 4.8: Control Method Selection

4.7 Position Based Force Control

A second position/force controller was developed based on the switching method described above. The Position Based controller, shown in Figure 4.9, measures the applied force and adjusts the captured desired Z position, that is used as input to the Cartesian Space Linearized controller, based on the error in applied force. This controller takes advantage of the fast, stable and highly accurate control provided by the position controller by using small gain values to adjust the desired Z-position, thus controlling the desired force. A PID controller was designed to adjust the desired Z-position with the initial gains determined experimentally, using the SimMechanics model, to be $k_{p_f} = 0.001$, $k_{i_f} = 0.001$ and $k_{d_f} = 0.0001$.

4.8 Conclusion

The chapter started with an explanation of the methods that were used to create trajectory generating algorithms in both joint and Cartesian space in Simulink using a quintic polynomial. How these algorithms were then used to generate various shapes to be traced by the CataLyst-5 was then presented. The development and initial gains tuning of three position controllers, PD Independent, Joint Space Linearized and Cartesian Space Linearized was pro-

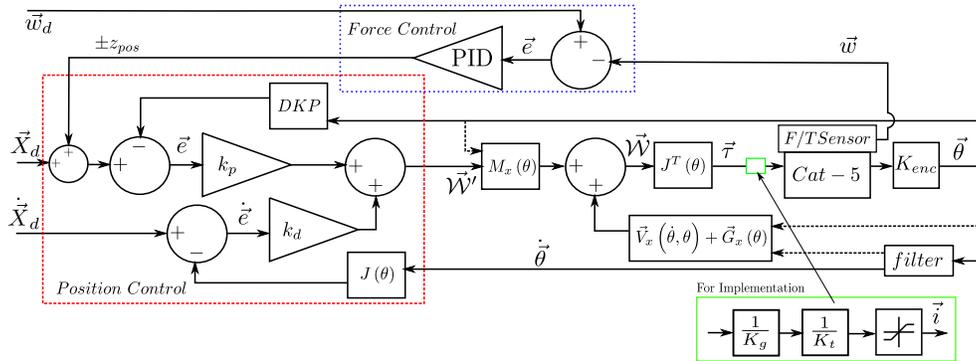


Figure 4.9: Position Based Force Control

vided in detail. Lastly, based on the Cartesian space position method, the creation of two position and force controllers Hybrid Position/Force and Position Based control was explained.

5 Simulation Results

The three position controllers that were developed for the CataLyst-5 were used to control the SimMechanics model of the robot in following the shape trajectories described in Section 4.1, with primary focus on the 5 point star. The accuracy of each of these controllers in following the desired trajectory are compared, along with the control currents sent to the C500D.

The two force controllers were used with the SimMechanics model and various simulation results are collected and discussed. As a baseline, the control action from the Cartesian Space Linearized controller, when following the star trajectory in free space, is investigated and compared to the control action from the two force controllers when following the same trajectory while in contact with the surface. Also a comparison of the force controllers' accuracy in applying the desired force is discussed. Lastly, the errors in the desired trajectory while in contact with the surface are investigated for each of the controllers.

For both the position and position/force simulation results, the overall controlled movement involves the robot arm starting from the ready position and moving down to a point just above the drawing plane at the shape starting point. The end-effector is then moved slowly down along the Z-axis, in a guarded movement that allows for controlled impact with the drawing surface. For the position controllers the desired shape is then traced along the X-Y plane in free space, after which the end-effector is moved up along the Z-axis then returned to the ready position completing the movement. For the force controllers, once contact is detected with the surface the desired shape is traced along the X-Y plane with the end-effector again returning to the ready position upon completion. The order in which the lines for the 5 point star are drawn are shown in Figure 5.1, where 1 indicates that start (and end) point with the subsequent points that the end-effector moves to in numerical order.

5.1 Comparison of Position Controllers

For the comparison of the position controllers, the simulation results were collected with the three controllers following trajectories generated at 100% of the maximum joint and/or Cartesian speeds that were determined in Section 4.4.

The accuracy of each of the three controllers in following the star trajectory can be seen in the following figures. As shown in Figure 5.1, using the simplest control method, the PD Independent Joint controller results in the greatest overall error between the desired and actual trajectory. The nearly constant error along the X-axis can be attributed to the sag of joints 2, 3 and 4 due to gravity as was observed when tuning the gains. This error could be lessened or removed with the addition of feed-forward gravity compensation or integral control action. Integral action was not implemented with this controller due to the possibility that its slow response to joint errors could lead to instability at higher joint speeds. Feed-forward gravity compensation was not considered when designing this controller, but could prove to be an easy to implement improvement of accuracy on what is a very fast method of controlling the robot in joint space.

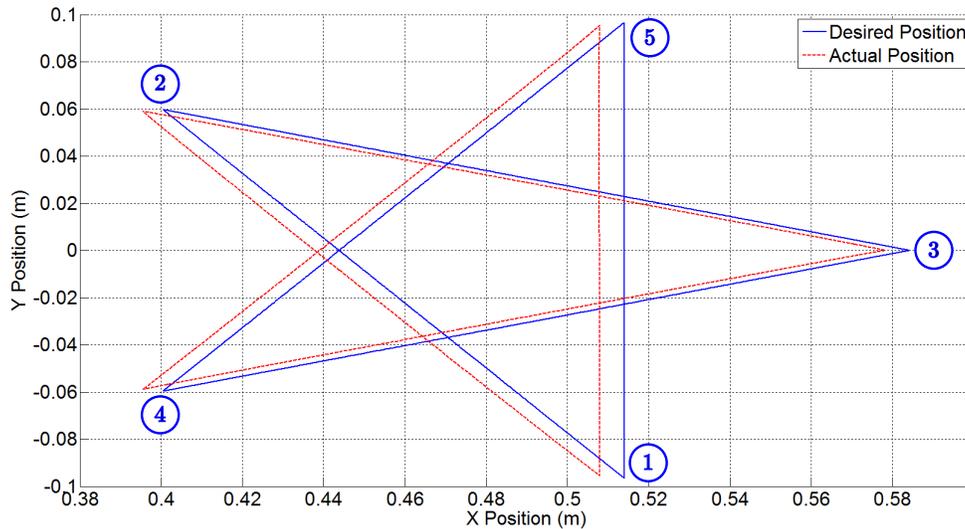


Figure 5.1: PD Independent Control - Star Trajectory

The Joint Space Linearized controller when following the star trajectory, as shown in Figure 5.2, removed the steady state error due to gravity and significantly reduced the overall error compared to the PD Independent controller. The most notable error with this controller is when the trajectory is

changing in both the X and Y direction. The cause of this error is difficult to determine as it is not consistently in any one direction. Further tuning of the gains for this controller could potentially reduce this error. The addition of integral control action could also help to reduce this error. Significantly more complex than PD Independent control, the Joint Space Linearized controller does not provide notably greater benefit than could be achieved by adding feed-forward gravity compensation to the previous controller, especially for the relatively simple movements being followed in this research.

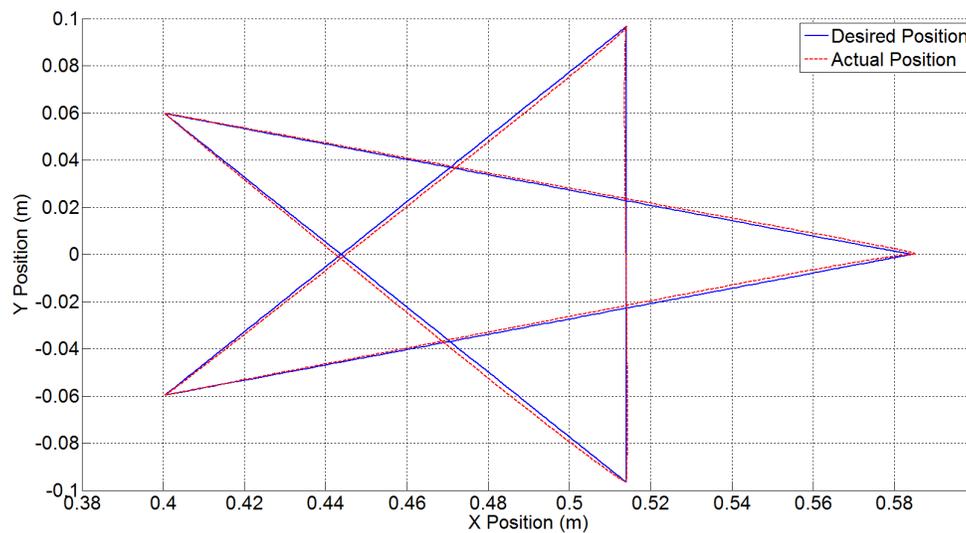


Figure 5.2: Joint Space Linearized Control - Star Trajectory

The final and most complex method, the Cartesian Space Linearized controller, results in the best tracking of the star trajectory, as shown in Figure 5.3. The results might be expected as the control signal is based on the error of the position and orientation of the end-effector, though the actual position is only calculated based on joint positions using the solution to the DKP. That the desired trajectory can be followed nearly exactly using only the joint positions highlights the benefits of this controller. That said, any flexibility in the manipulator or joints could result in inaccuracy in end-effector position, though no more so than would be realized with the joint space controllers. The biggest disadvantage of this controller is that the highly complex nature of this type of control significantly limits its potential application in industry.

As further comparison, the magnitudes of the errors between the desired and actual trajectories along the X and Y axes for each of the three controllers

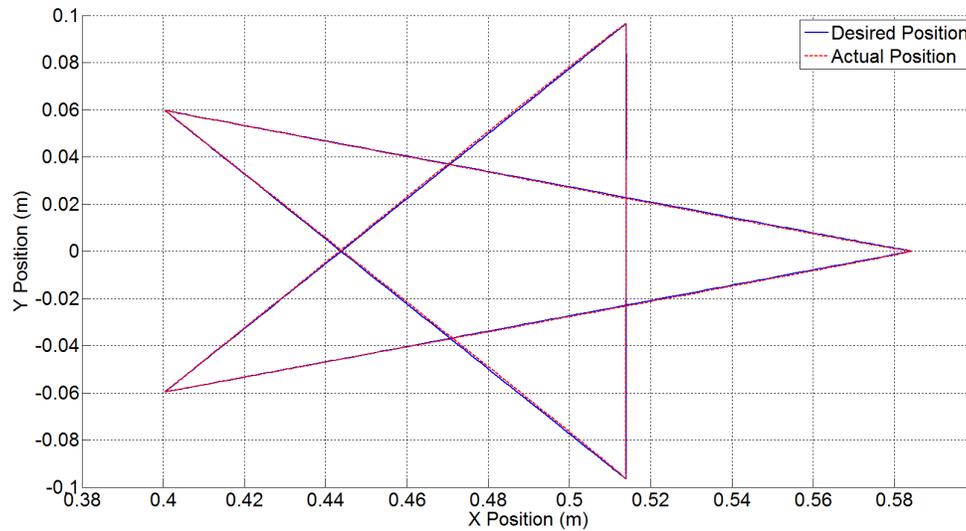


Figure 5.3: Cartesian Space Linearized Control - Star Trajectory

is shown in Figure 5.4. The plot further highlights the large, nearly constant error along the X-axis for the PD Independent controller. The five notable error regions for the two linearized controllers occur during each of the five line segments that make up the star trajectory with the magnitudes of these errors for the Cartesian Linearized controller nearly half the magnitude of those of the Joint Linearized method. Of note, the error along the X-axis for the two linearized controllers are opposite in sign and very similar in shape if not magnitude. The plot of errors again confirms what was determined above that the more complex of the three controllers provides the most accurate trajectory following results.

Looking at the control torques generated by each of the three controllers in Figure 5.5, it is interesting to note that the magnitudes of the torque values for each of the controllers are nearly identical. The biggest difference between each of the controllers is that the control signals from the two joint space based methods contain a notable amount of noise compared to the more complex Cartesian space controller.

Additional simulation results showing the same plots provided above, but for the spiral trajectory can be found in Appendix F. The same observations as were discussed above can be made for the results found when the simulation is following the spiral trajectory.

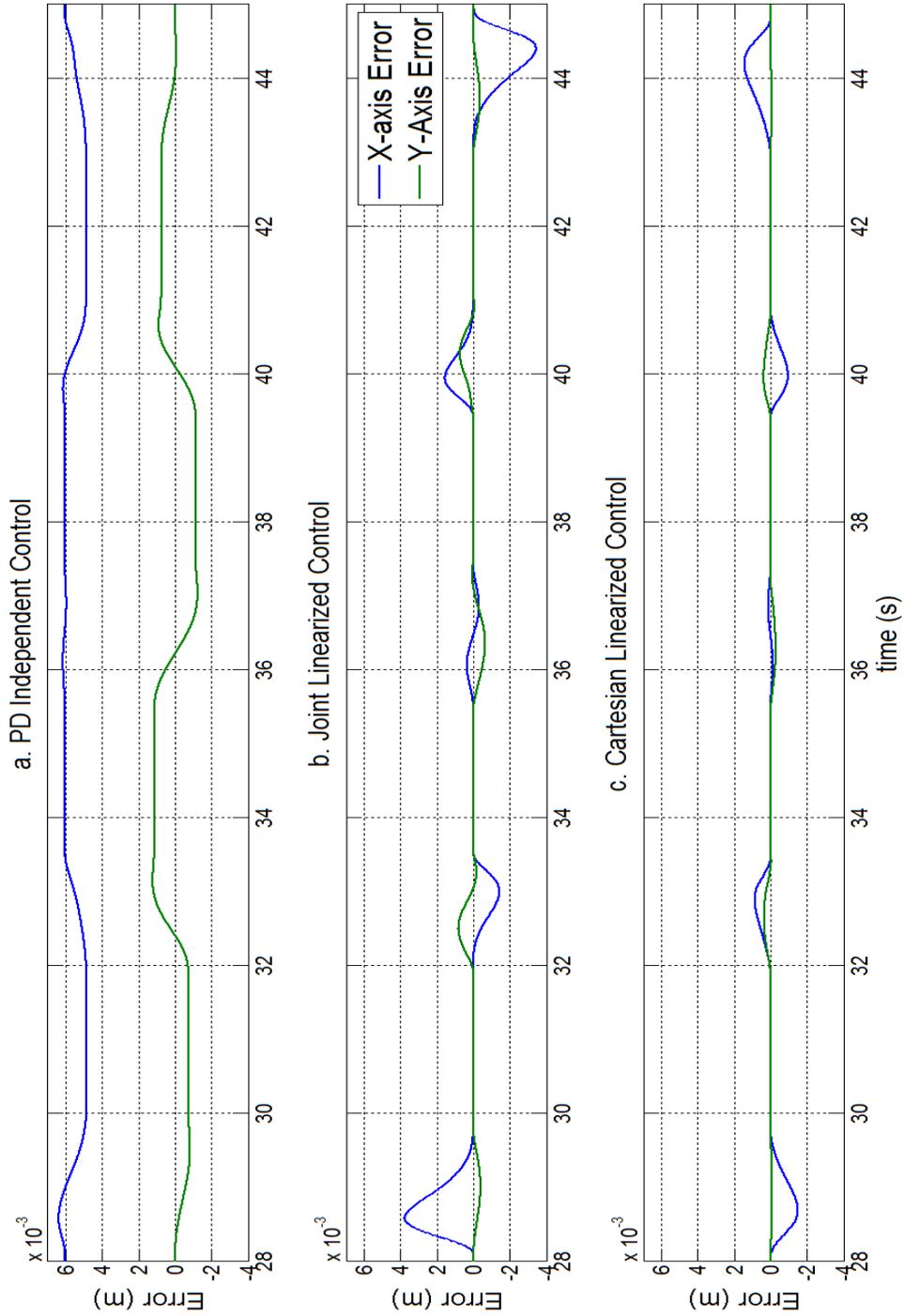


Figure 5.4: Trajectory Tracking Errors X & Y Axes - Star Trajectory 66

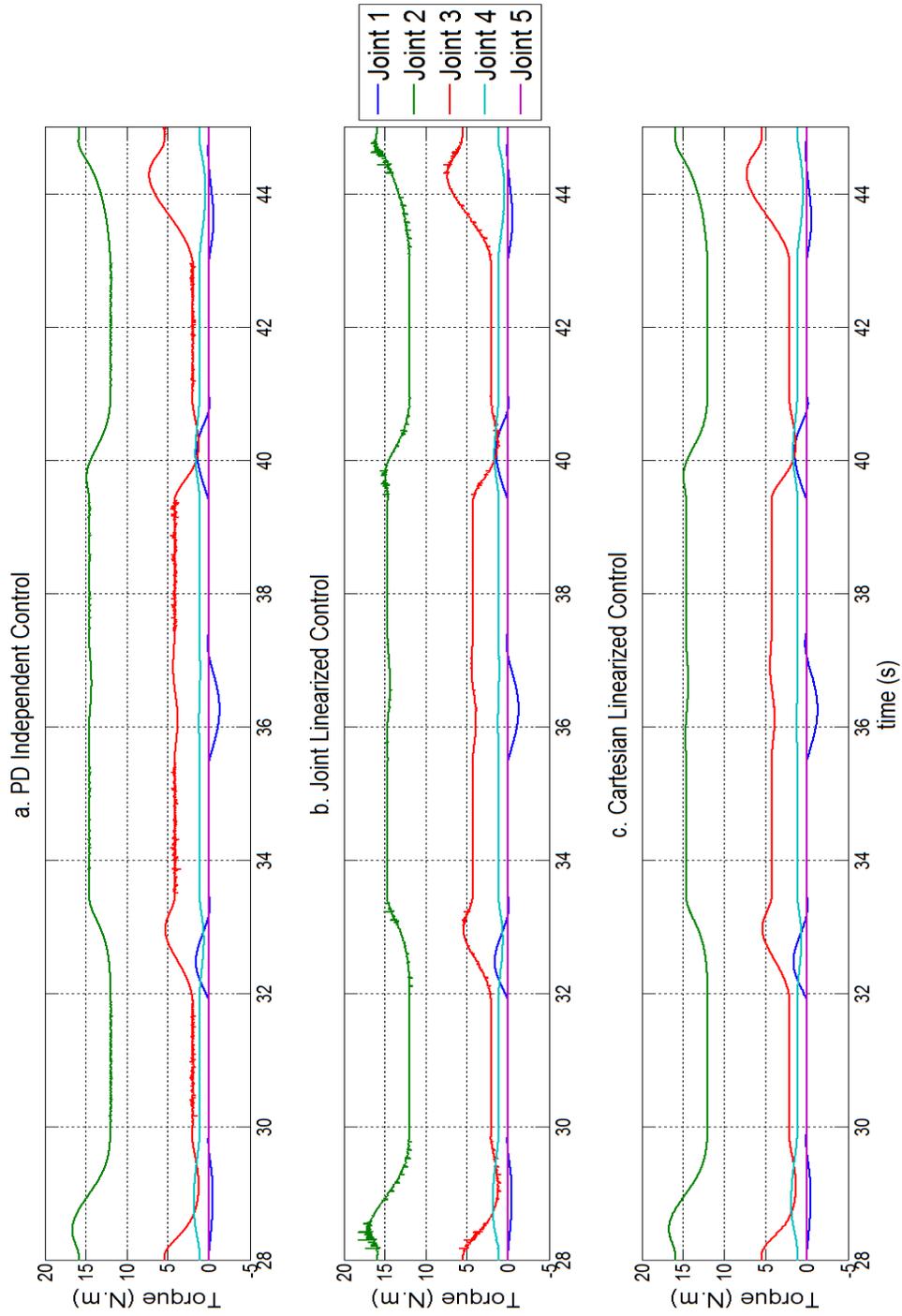


Figure 5.5: Control Torques - Star Trajectory

5.2 Comparison of Force Controllers

For comparison of the two force controllers, simulation results were generated with the controllers following trajectories that were generated at 50% of the maximum joint and/or Cartesian speeds that were determined in Section 4.4.

As both of the force control methods that were developed make use of the Cartesian Space Linearized controller for position control, it is useful to investigate the control signal from this controller when following the desired trajectory in free space, as a baseline. Figure 5.6.a shows the control effort (\vec{W} from Figure 4.6) from the Cartesian Space controller when following the star trajectory. Each of the five distinct areas of the plot where control action is being made coincides with the five lines that are drawn in generating the star. Of interest from this plot is that despite the movements only being in the X and Y directions when the star portion of the trajectory is being followed, a very large control action is required along the Z-axis just to maintain the end-effector vertical position along the desired X-Y plane.

The large control action along the Z-axis is due to the CataLyst-5 being an articulated robot. Joints 2 through 4 are simultaneously responsible for bringing the end effector closer to (or farther away from) the base while also maintaining the vertical position of the end-effector. For the trajectories being followed any movement in the Y direction is primarily achieved by joint 1, thus the relatively small control action in that direction. That the movements in the X direction are the larger portion of the trajectory and are achieved by joints 2 through 4 requires the larger control action as these movements are essentially balanced against the control in the Z direction. That such a large control action is required along the Z-axis is significant because when position/force control is executed, the force controller needs to provide similar magnitudes of actuation in order to maintain contact with the surface and provide the desired applied force.

Figures 5.6.b and 5.6.c show the control actions from the Hybrid Position/Force controller and the Position Based Force controller respectively. Comparing the control signal from the Hybrid controller in constrained motion with that of the free movement, it can be observed that the control action in the X direction has the same general shape but is offset from the 0 axis. The control action also has a larger magnitude when end-effector movements are follow each of the five lines. The control action in the X direction for the Position Based Force controller is similarly offset, though with magnitudes that are very comparable, if not identical, to those of the controller in free space.

The control signal in the Y direction changes notably when the Hybrid

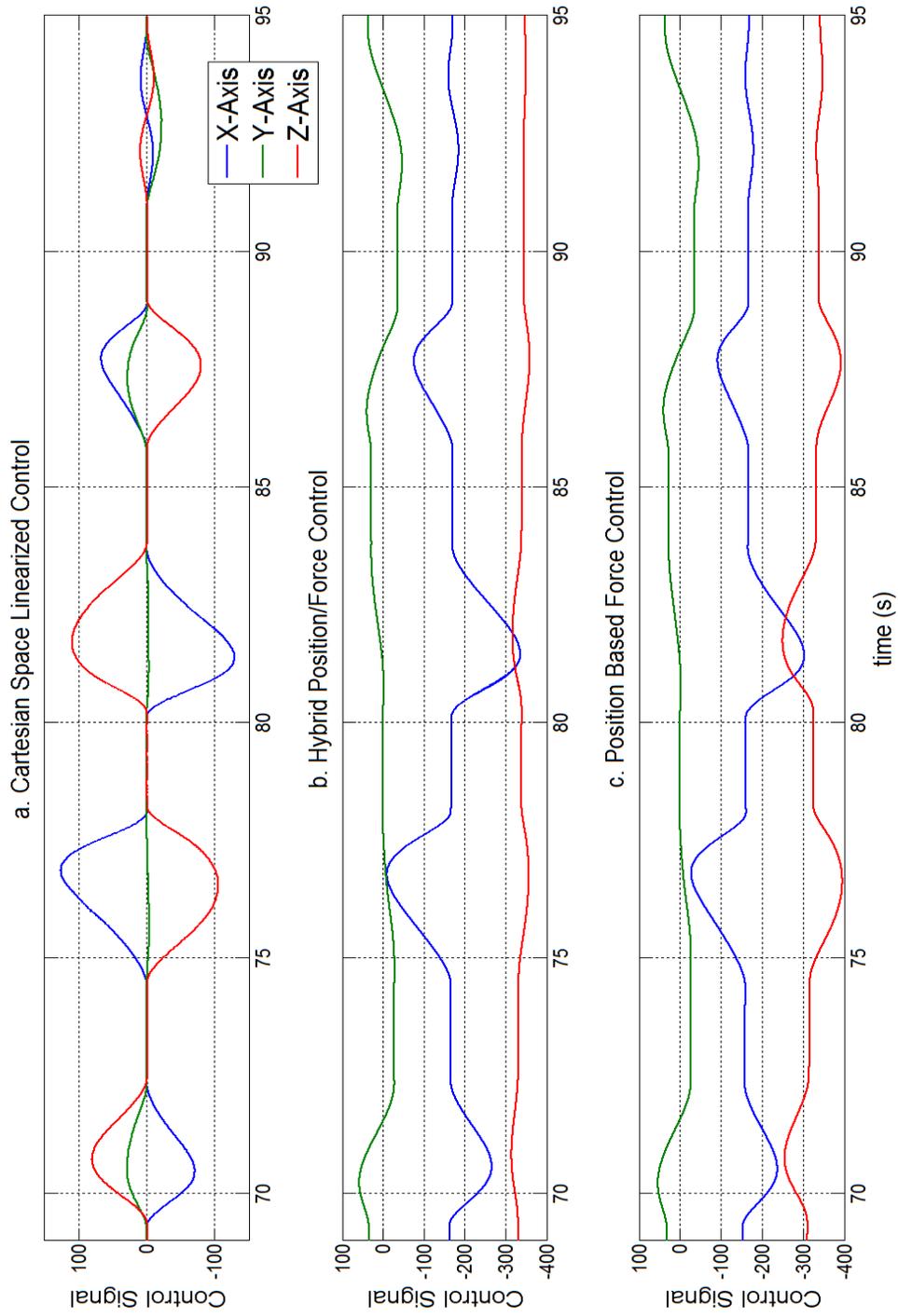


Figure 5.6: XYZ Control Signals - Star Trajectory

controller is in contact with the surface, in that it remains non-zero for periods of the simulation when lines are not being sketched (no movement in X-Y plane) and has slightly larger magnitudes when movements are conducted. In experimentation this could be attributed to the surface friction between the chalk and the drawing surface. That this friction was not included in the SimMechanics model leaves this variation without a definitive explanation. One potential source is that the end-effector position when in constrained vice free motion is at a higher position along the Z-axis, as determined by when contact is made with the drawing surface. This higher end-effector position could result in a larger control actuation requirement in the Y direction due to the different configuration of the robot arm. The control signal in the Y direction from the Position Based Force controller is essentially identical to that of the other force controller in both shape and magnitude.

Where the two force controllers differ most significantly is in their control action in the Z direction. Both provide a control action that is offset in the negative direction when compared to the controller in free motion. The Hybrid controller provides a control signal that while similar in shape to the free motion baseline, is much smaller in magnitude. The control signal from the Position Based controller is very similar to the baseline signal in shape and only slightly smaller in magnitude.

The difference in control signal magnitude between the two methods is a direct result of how they adjust the Z position of the end-effector based on the error in applied force. The Hybrid controller must provide the entire control signal in the Z direction from the force control subsystem. If the force controller were to have large enough gains to provide a Z-axis control signal similar to that in Figure 5.6.a there would be problems with it maintaining contact with the drawing surface. The Position Based controller is able to take advantage of the large gains in the Cartesian Space Linearized controller and thus only requires small gains to adjust the desired position based on the error in applied force. This results in a control method that can maintain contact and provide the large control signal in the Z direction required for an articulated robot like the CataLyst-5.

The effects of significant difference in control signals in the Z direction between the two force controllers can be observed in Figure 5.7 where the magnitudes of the error in applied force is significantly larger for the Hybrid controller. The error for both controllers follows the same general shape with both following a general trend towards 0N error from an initial error due to a larger than desired force being applied on initial contact. The peaks towards the positive direction occur when the end-effector is moving towards the robot base (1 \rightarrow 2 and 3 \rightarrow 4 in Figure 5.1) as the direction of joint rotation

required for this movement causes greater than desired force to be applied to the drawing surface. Similarly the peaks towards the negative direction occur when the end effector is moving away from the base of the robot ($2 \rightarrow 3$ and $4 \rightarrow 5$ in Figure 5.1) as the joint rotation causes the end effector to want to lift off the drawing surface.

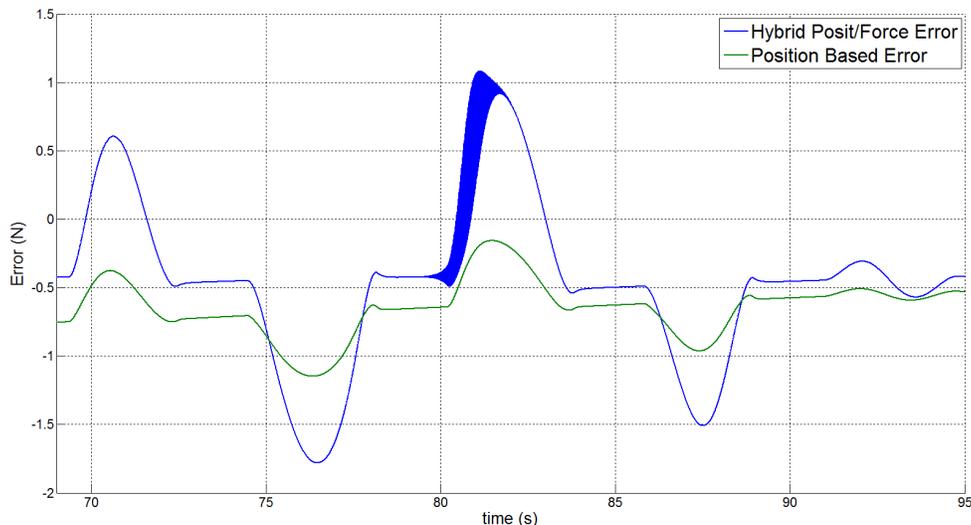


Figure 5.7: Error in Applied Force

The dark band that appears on the Hybrid error plot from about 80s to 82s is due to a rapid oscillation of the calculated force error. This oscillation initiates when the end effector is at the farthest point from the robot base, where accurate control of the Z position is the most difficult. The oscillation is likely due to the imperfect simulation of contact with the environment which required the use of very compliant surface stiffness values. As described in Section 3.5 contact was simulated by creating a closed kinematic chain between the end-effector and the contact surface using a series of SimMechanics joints. The kinematic chain included a prismatic joint that would be actuated with a force acting against the end-effector when the contact surface was reached. The simulations would fail if actual surface stiffness values representing the experimental drawing surface were used, thus requiring the use of artificially compliant surface stiffness values.

The trajectory tracking error in the X and Y directions of each of the force controllers is shown in Figure 5.8. The error in the Y direction is the same in shape and magnitude for both controllers. While the error in the

6 Experimental Setup

The CataLyst-5 Robot System is introduced with various parameters and limitations listed. The basic controller and power supply are introduced with an explanation of various control options and safety systems that protect the robot arm. Modifications that were made to the basic controller, along with the communication channels that are used, to allow for open architecture control are described. The QuaRC software that provides interface between controllers created in Simulink and the physical hardware is presented along with various hardware in the loop blocks and their functions. The force/torque sensor that is installed between the robot arm and the end-effector to provide the feedback required for position/force control is introduced with the parameters and limitations listed. Lastly, the robot contact environment that was devised for experimentally testing the position/force control algorithms is explained.

6.1 CataLyst-5 Robot System

Designed by Thermo-Fisher Scientific for micro-plate handling in a laboratory environment, the CataLyst-5 robot system as described in [60] and shown in Figure 6.1 consists of the serial arm robot, a C500C controller and the umbilical cables and power cords in the simplest configuration. A number of optional features can enhance the functionality of the robot system, including a teach pendant to allow for easier configuration, a homing bracket to ensure precise positioning in the workspace and a linear track system that expands the robots workspace considerably. Various end effectors can be fitted to the robot arm depending upon the application.

The serial arm robot is a five degrees of freedom manipulator that can be considered an articulated (RRR) manipulator as defined in [15], but with limited mobility at the wrist. The wrist of the Cat-5 can move in pitch (joint 4) and roll (joint 5), but not yaw. The joint numbering and orientation are

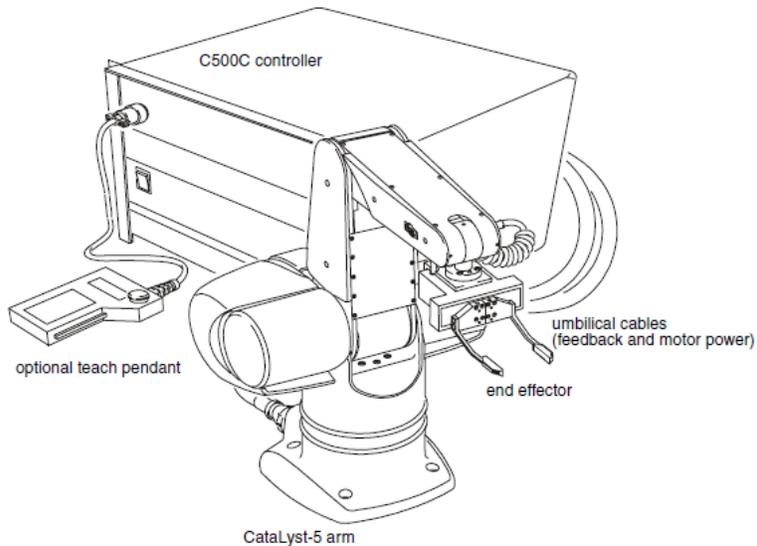


Figure 6.1: CataLyst-5 Robot System (from [60])

shown in Figure 6.2.

Each of the joints are actuated, through harmonic drive transmissions, by DC motors with rated voltages of 34V and a current-torque constant of 0.07NmA. Each of the joints are fitted with encoders that continuously measure the angular position of the motors and brakes that hold the robot in position between movements and when the robot is shut down. The transmission ratio, encoder calibration and a number of other parameters of the CataLyst-5 robot system can be found in Table 6.1

Table 6.1: CataLyst-5 System Parameters (adapted from [64] and [60])

Joint	Gear Ratio	Encoder Calibration $^{\circ}/\text{count}$	Maximum Speed $^{\circ}/s$	Default Acceleration $^{\circ}/s^2$	Continuous Stall Torque Rating Nm
1	72 : 1	0.001250	210	500	9.6
2	72 : 1	0.001250	210	500	9.6
3	72 : 1	0.001300	210	500	9.6
4	19.6 : 1	0.005625	551	1836	3.4
5	9.8 : 1	0.011250	1102	3673	1.7

Specifications determined for 1kg payload at tool flange.

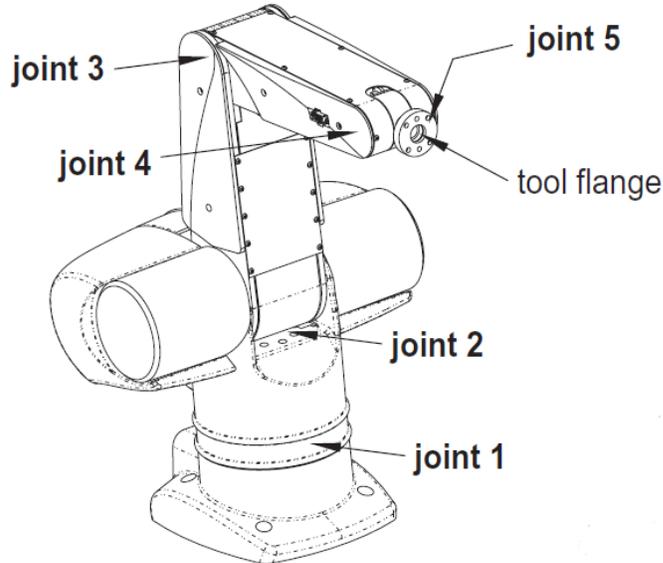


Figure 6.2: Catalyst-5 Robot Arm (from [60])

6.2 C500C Controller and Power Supply

The basic controller and power supply for the CataLyst-5 robot system (C500C) provides power, safe operation and basic motion control of the robot arm [60]. The C500C actuates the joints via power amplifiers, receives the position feedback from the encoders and shuts down the robot if a potentially damaging condition is detected. The C500C also has the ability to execute pre-programmed trajectories and operate independently of a computer terminal by using the CRS Robot Operating System (CROS) that provides low-level system functionality. Simple position control based tasks are downloaded onto the CROS using a support computer via the ROBCOMM application with control sequences developed in the RAPL-3 programming language.

Ports on the back of the C500C provide connections for both the power supply and feedback umbilicals that provide the link with the robot arm. Serial and parallel ports on the back of the controller provide an interface for force torque sensors, end effectors and other accessories that may be used with the robot system. The back of the C500C also provides the connection for the emergency stop (or E-Stop) circuit that can be used to hard wire a shut down trip specific to the application where the robot is installed. For the

installation at RMC, the E-Stop circuit is wired to the plexiglass door that allows access to the safety enclosure surrounding the robot so that the robot cannot be moved when the door is open.

Connection ports on the front of the panel of the C500C provide the serial cable interface with the support computer and the connection for the teach pendant. The teach pendant, as shown in Figure 6.3 allows for basic positioning of the robot arm using direct angle control of the joints, Cartesian or radial coordinate based control of the end effector position (and orientation) and position control of the end effector based on an end effector based reference frame. The teach pendant can also be used to program and execute position based applications into the CROS for installations without a computer terminal. Operation of the robot using the teach pendant requires the proper positioning of the live man switch, which ensures safety of the operator from either electric shock, or pinch hazards with the arm. The teach pendant is also fitted with an E-Stop button that will shut down power to the robot arm when actuated.

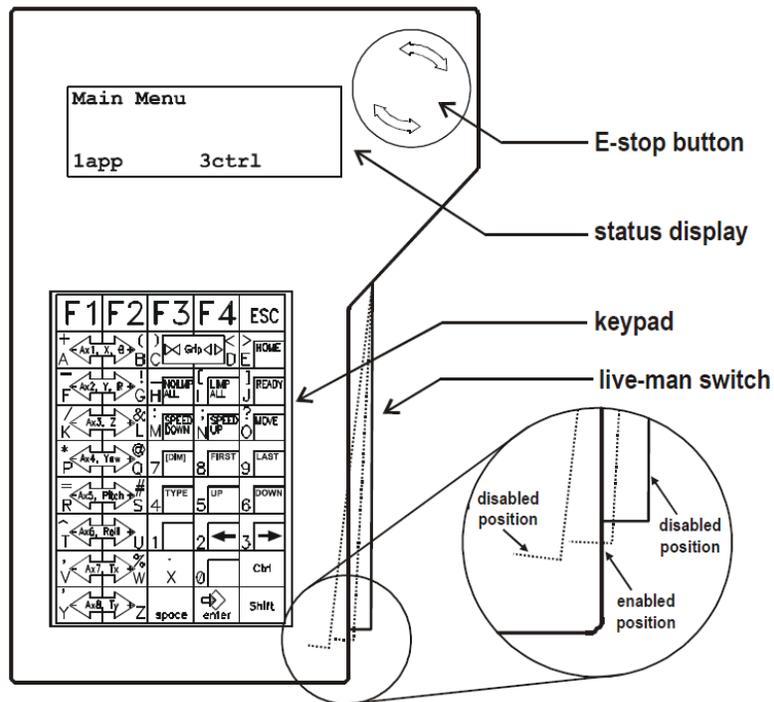


Figure 6.3: Catalyst-5 Teach Pendant (from [60])

6.3 Open Architecture Control

The controller installed with the RMC CataLyst-5 is a C500D variant that has been modified by Quanser to allow for open architecture control of the robot arm for research and education purposes. As shown in Figure 6.4 the C500D open architecture modification allows for various user designed control algorithms to be applied (via the solid black lines) to the robot arm, bypassing the CROS.

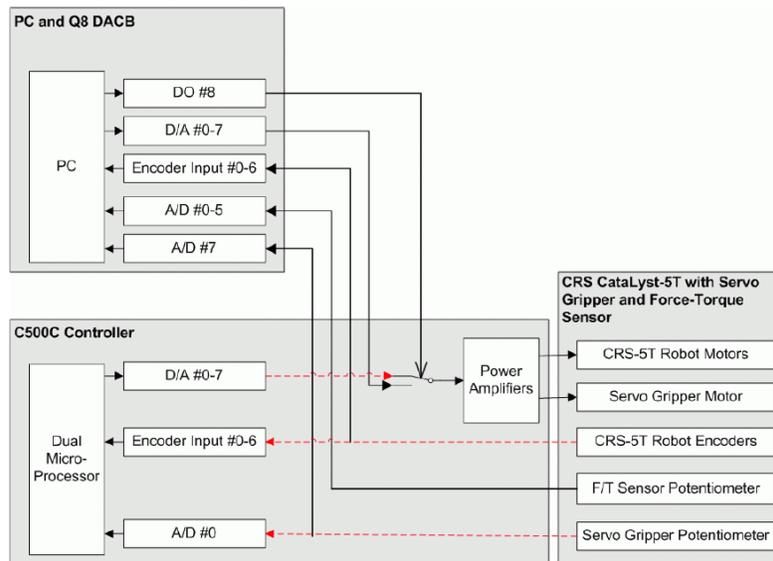


Figure 6.4: C500D Open Architecture Controller (from [60])

Open architecture control is achieved using a computer with a Quanser-8 (Q8) data acquisition card that connects to the modified panel on the back of the C500D controller using 3 ribbon cables. The Q8 card provides communications channels for the following digital and analog inputs and outputs:

- DO#8 - digital output signal switching between closed and open architecture control;
- D/A#0 – 4 - analog control currents to the motors of joint 1 (#0) through joint 5 (#4) respectively
- D/A#5 - analog control current to the optional linear track motor
- D/A#7 - analog control current to the servo gripper motor
- Encoder Input #0 – 4 - encoder position signal for joint 1 (#0) through joint 5 (#4) respectively

- Encoder Input #5 - encoder position signal for optional linear track
- A/D#0 – 2 - analog signal input of the force along the X (#0), Y (#1) and Z(#2) axes
- A/D#3 – 5 - analog signal input of the torque along the X (#3), Y(#4) and Z(#5) axes
- A/D#7 - analog input signal of the position of the servo gripper from the potentiometer

When open architecture control is initiated and a signal is sent from the DO#8 output, the control current into the power amplifiers comes from the Q8 card vice the closed architecture control in the C500D to ensure only one controller can actuate the robot at a time. The feedback signals from the motor encoders and the gripper potentiometer are sent to both the Q8 card and the closed architecture controller regardless of the mode of operation. This ensures that the built in safeties that protect the robot arm from runaway conditions and collision are still available. The signals from the force/torque sensor (if fitted) can only be sent to the Q8 card as the closed architecture controller does not have the necessary analog input channels.

6.4 Quanser Real-Time Control (QuaRC) Software

Quanser's real-time control software (QuaRC) is used to integrate controllers created in Simulink with various types of hardware to be run in real time. The software generates C source code from the Simulink model to suit the particular target platform based on the blocks used from the QuaRC library, then automatically compiles it and downloads the code to the target. QuaRC also allows for real-time communication between the Simulink model and the target hardware which allows for monitoring of the generated signals from the hardware while simultaneously tuning the parameters of the controller [65].

For the CataLyst-5 installation at RMC, QuaRC is used to interface directly with the Q8 card using Hardware-in-the-Loop (HIL) blocks that are added to the Simulink controller models. Four types of HIL blocks are available for use:

- HIL Initialize Block
- Immediate I/O HIL Block
- Timebase HIL Block
- Buffered I/O HIL Block

The HIL initialize block is required for any model that requires hardware access. The block both initializes the HIL board and allows the user to assign

a name for the board and identify the type, Q8 for this application. The name assigned for the board is used with the other HIL blocks to ensure the appropriate signals flow between the Simulink model and the hardware. The HIL initialize block also allows the user to configure the digital lines of a card as either inputs or outputs and define the range of the analog inputs and outputs, where applicable.

The immediate I/O HIL blocks are used to read or write to specified channels every time the block is executed. A single HIL read block can be configured to read analog, encoder and digital channels, or separate specific blocks can be used for the analog, encoder and digital signals from the hardware. Similarly, a single HIL write block can be used to send both analog and digital signals to the hardware, or separate blocks can be used, as shown in Figure 6.5. The immediate I/O HIL write blocks are used in sending joint motor control currents and gripper control signals in real-time control applications. Immediate HIL read blocks are used to collect signals from the force/torque sensor and to determine the position of the gripper.

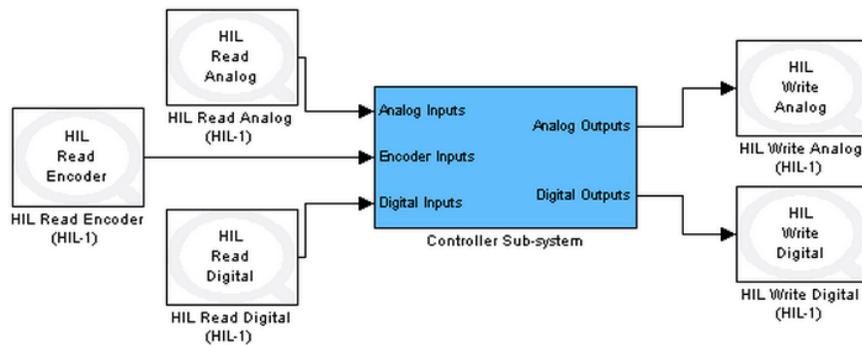


Figure 6.5: Immediate I/O HIL Blocks (from [65])

The timebase HIL blocks are used to read or write to specific channels using the sampling rate of the model. When the block is used, the acquisition timing is based on the timebase of the data acquisition card vice the QuaRC model, resulting in better performance due to a more efficient hardware timebase. Only one timebase block can be used per Simulink model as different timebases could result in incorrect control execution and system instability. Similar to the immediate HIL block, a single block can be used for multiple and different types of read or write signals, which is the recommended configuration to avoid multiple timebases. A timebase HIL block is used to gather the joint position information from the motor encoders.

The buffered I/O HIL block is used to collect buffered data by specifying the number of samples and the sampling rate. The collected data is then output from the block once all samples have been collected, as such this block is better suited to data acquisition than control. [65]

6.5 Manipulator Contact Environment

In order to test the effectiveness of the position/force controllers, an experimental contact environment, as shown in Figure 6.6, was devised involving a chalkboard surface upon which the robot would be required to apply the appropriate force to draw various shapes (lines, stars, circles and spirals) using chalk. The drawing surface which the robot is programmed for is parallel with the X-Y plane of the base frame, but the actual Z position of the surface varies slightly over the drawing workspace, requiring the control algorithm to adjust the position in the Z-axis based on the detected force.

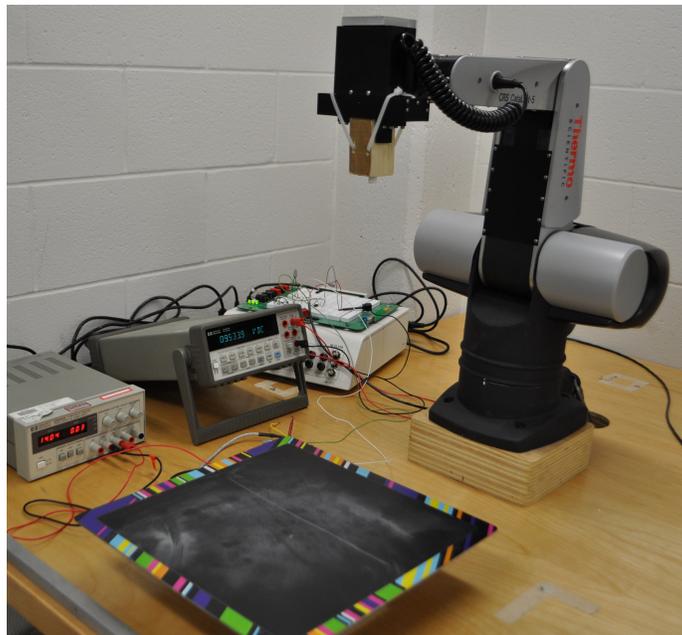


Figure 6.6: Experimental Setup with Drawing Surface

Initial intention with this research was to use the force/torque sensor fitted to the CataLyst-5 and described in Appendix G to measure the force applied to the environment by the end-effector. In order to calibrate the force/torque

sensor and monitor the force being applied by the robot arm, the chalk board drawing surface was mounted on an Omega Engineering LCA 10 triple beam platform load cell (specifications in Appendix I), that was re-purposed from previous research. As no calibration info was available for the load cell, the relationship between the mV output of the Wheatstone bridge circuit to the applied force was determined experimentally by calibrating it using known masses and assuming a gravitational constant of 9.81m/s^2 . Using a supply voltage to the Wheatstone bridge of 14V , the force measured by the load cell in N is given by the following equation where U is the measured mV output from the Wheatstone bridge.

$$F_{cell} = -1.4402U + 71.4 \quad (6.1)$$

When it was determined that the controller for the force/torque sensor was unserviceable and could not be used to measure the force applied by the end-effector on the contact surface, the experimental setup was reconfigured to use the output from the load cell as the measured force for the force controller. This was deemed to be an acceptable solution as the original intent of this research was to control the force along the Z-axis.

When the output from the load cell was input directly into the C500D panel, large gains were required to convert the small 0.00062V/N signal to an appropriate force value. These large gains greatly amplified the signal noise that is present due to the ribbon cables that connect the C500D to the control computer. This amplified noise made the signal unusable for control purposes. In order to obtain a usable measured force signal, the output from the load cell was amplified, external to the Simulink model, using a SN741 operational amplifier in the non-inverting configuration shown in Figure 6.7.

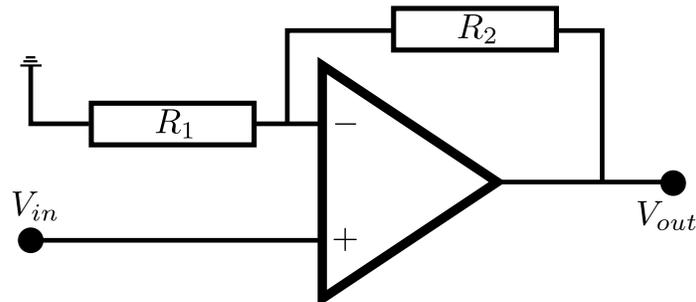


Figure 6.7: Non-Inverting Operational Amplifier

The values of the resistors used with the op amp determine the level of amplification. Two different resistor configurations were used to determine an

appropriate signal amplification while staying within the $\pm 10\text{V}$ analog input limits set out in [63]. The first involved two $62.5\text{k}\Omega$ resistors in series in the R_2 position and one $0.97\text{k}\Omega$ resistor in the R_1 position. This resulted in the inverse linear relations shown in Figure 6.8 with a maximum input voltage of 6.5V for 0N load and a resolution of 0.09V/N . Three $62.5\text{k}\Omega$ resistors in series in the R_2 position resulted in a similar inverse linear relationship, but with a maximum input voltage of 9.7V for 0N and an improved resolution of 0.13V/N . As it provided improved resolution, the configuration with 3 resistors in series was used for all experimental work.

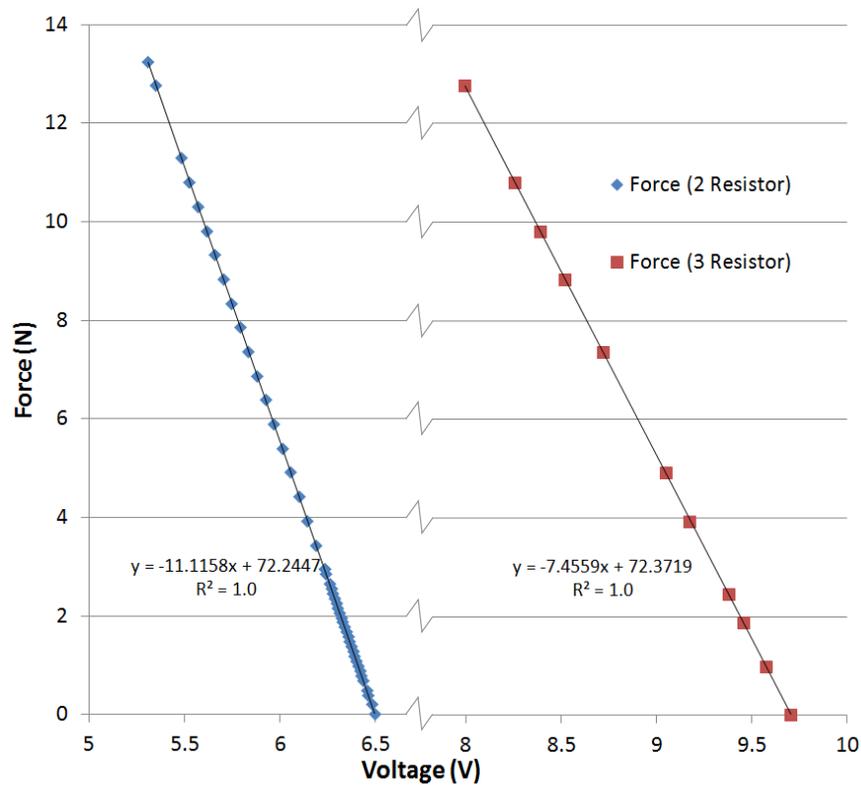


Figure 6.8: Operational Amplifier Calibration

An instrumentation amplifier was considered as a means of further expanding the resolution of the signal, but the op amp circuit was determined to be sufficient to eliminate the signal noise and provide an adequate control signal. With the load cell calibrated, the robot arm was operated using the teach pendant to determine that 5N of force is required to draw a sufficiently

thick and full chalk line. This 5N value will be the desired force used with the force controller.

6.6 Controller Implementation

To implement the controllers as designed and tested in simulation, a number of modifications were required to ensure appropriate communications with the C500D and the CataLyst-5. Controllers that were designed by Quanser, described in [63], and known to properly interface with the robot installed at RMC, were used as a starting point when modifying the controllers described in Chapter 4 for implementation. The HIL Initialize, HIL Read Encoder Timebase and HIL Write Analog blocks were copied from the Quanser models. The configuration of the initialize block was confirmed as relating to the Q8 DAC, while the read and write blocks were modified to eliminate the optional track position/actuation signals.

As it affects some of the background communications between the control computer and the C500D, that are not explained in [63], the CRS Takeover block as designed by Quanser was used directly. Along with the background communications, the blocks primary role is switching the C500D between closed and open-architecture control modes. When initially implementing the controllers on the robot, Simulink model errors were encountered and were determined to be related to the model configuration parameters, that again were not well documented. To remedy these issues, new files for each controller were created by copying a working version of the controller to a copy of the Quanser Simulink files but with all non-QuaRC blocks removed. These new files eliminated the model errors and allowed for the necessary communications between Matlab and the CataLyst-5.

As the signals input from the timebase HIL block consist of encoder counts vice joint angles an appropriate calibration equation for each joint is required. Each of the incremental encoders provides the motor (vice joint) position, at a rate of 4000 counts per revolution. As none of the available documentation for the CataLyst-5 detailed the calibration of the encoders, it had to be determined experimentally. As the joints are gear driven and that the joints farther out the kinematic chain are driven by motors located in the shoulder assembly through chains, made the encoder/joint angle relationship difficult to determine. As example, the angle for joint 3 is dependent on the encoder count for motors 2 and 3. The final equations relating encoder counts to joint angles, as defined by the DH axes shown in Figure 3.1, are given below. These equations were used in a Matlab function block to convert the encoder counts

to joint angles in real time.

$$\begin{aligned}
\theta_1 &= -counts_1 K_{enc_1} \\
\theta_2 &= \frac{\pi}{2} - counts_2 K_{enc_2} \\
\theta_3 &= -\frac{\pi}{2} + (counts_2 + counts_3) K_{enc_3} \\
\theta_4 &= \frac{\pi}{2} - counts_4 K_{enc_4} - \theta_2 - \theta_3 \\
\theta_5 &= (counts_4 + counts_5) K_{enc_5} \\
&\text{where} \\
K_{enc_i} &= \frac{2\pi}{4000} \frac{1}{K_{g_i}}
\end{aligned} \tag{6.2}$$

The output from the controllers as used in simulation with the SimMechanics model is joint torque. As the input for the C500D in open architecture mode is a control current, a conversion was required. The conversion is given by Eq. 6.3, for the i th joint, that makes use of the motor torque constant and the gear ratio for each of the joints. Eq. 6.3 was used to create a subsystem that also included current saturation blocks set at 3.5A to ensure the motors in the robot are not damaged if a problem arises during experimentation. When the controllers were first used with the robot, these saturation blocks were set to much lower values to further protect the robot until the controllers were proven to function as expected.

$$I_i = \frac{1}{K_t} \frac{1}{K_{g_i}} \tau_i \tag{6.3}$$

With the torque generated by the controllers converted to an appropriate control current, the last step prior to passing this signal to the C500D to actuate the robot is to ensure an appropriate sign is used. Based on the configuration of the robot, the control signals for joints 3 and 5 are inverted to ensure the motors is actuated in the correct direction.

The amplified voltage signal from the load cell is input to the control system using a HIL read analog block. The signal is passed through a second order low pass filter using a cut-off frequency of 1Hz. As the voltage signal was found to vary over time, a Mean block was used to average out the signal. Finally the voltage signal is converted into a force using Equation 6.1. The calculated applied force is used directly by the force controller and also passed through a 1N Deadzone block followed by a latching switch that pro-

vides indication of contact and the transition between purely position and position/force control.

6.7 Implementation Issues

A number of problems were encountered while implementing the controllers on the CataLyst-5 robot. First, Timeseries databases were used to store the trajectory information that would be used by the controllers. When it was determined that Timeseries .mat files were not compatible with QuaRC, all trajectory files were recreated as Arrays. It was also determined based on information within [65] that the trajectory files need to use a fixed time step size as QuaRC will interpolate between data points as necessary, but only if the step sizes are consistent.

The first two controllers implemented on the robot were the PD Independent and Joint Linearized methods. Using the optimized gains described in Section 4.4, and found in Tables 4.3 and 4.4, with the robot resulted in chattering from some of the motors while moving through the desired trajectory. The gains were reduced to initial values described in Sections 4.2 and 4.3, then some basic gains tuning was conducted to try to reduce the overall end-effector orientation error found using these controllers. As these controllers would not be used with force control, the gains were not optimized with the actual robot.

Building and compiling the significantly more complex Cartesian Space Linearized controller for use with the CataLyst-5 provided some problems initially. The computer that was set up for use with the robot required a 32-bit version of Windows 7 in order to be able to run ROBCOMM (required to initially home the robot). This limited the computer to 4 GB of RAM which proved to be insufficient for the more complex controllers. To solve this two computers were configured, one running ROBCOMM and the other running QuaRC. This requires regular switching back and forth as each controller is trialed in QuaRC and the robot reset with ROBCOMM. A KVM switch was set up between these computers and a 9-pin serial cable switch was sourced to ensure appropriate background communications with the C500D. The process for running Simulink/QuaRC based controllers with the RMC CataLyst-5 two computer setup is summarized in Appendix J.

The model complexity and large file sizes led to some issues compiling the models due to long directory/file names. When running the complex controllers it was found that copies of the controllers placed directly in a folder on the C: drive alleviated this problems and allowed for more reasonable build

times. Operating the controllers from the C: drive was also found to remedy a problem where data from the experimentation would not be properly exported by the To File blocks that were set up to capture results. These blocks were critical for troubleshooting the controllers as Simulink scope blocks do not function properly with QuaRC, only capturing the most recent 5s portion of the signal they are attached to.

With the sufficient computing capacity and proper file naming conventions determined, the Cartesian Space Linearized controller could be implemented on the CataLyst-5. When the controller was used with the initial gains determined in Section 4.5 of $k_p = 400$ and $k_d = 40$, joints 4 and 5 showed significant error with the controller unable to maintain the desired orientation. As maintaining the desired end-effector orientation was critical to implement force control as designed, significant work went into modifying the controller including increasing the gains significantly and adding integral control action to the first three control channels that provide input based on orientation error.

Lastly, a number of problems were found when trying to obtain a consistent voltage signal representing the force measured by the load cell. Due to the large amplification of the signal, small voltage fluctuations from the building's power supply were found to greatly affect the force detected by the controller. This led to some unexpected consequences including switching to the force controller prior to contact being made resulting in undesirable system response. The voltage variation, partially attributed to the C500D, required that a small offset be added in the Simulink model based on the output voltage from the OpAmp as measured by the voltmeter connected to it. To further reduce the affects of voltage variation, an uninterruptible power supply was borrowed from the Electrical and Computer Engineering Department and used to power the load cell voltage source and the OpAmp circuit. This power supply was found to greatly reduce, though not eliminate, the voltage fluctuations. The reduced fluctuations were managed by increasing the deadband in the force detection switching circuit described above. Also the force controller was configured such that the desired applied force would only be input once contact was detected, thereby eliminating any built up control signal coming from that portion of the controller.

6.8 Conclusion

The experimental setup that was used in this research was presented in detail, including an introduction to the CataLyst-5 robot system, its associated power supply and controller and the open architecture control software provided by

Quanser for use in conducting controls research. The additional hardware that was used in conjunction with the robot to implement force control was explained, including the ATI force torque sensor and the contact environment that was used, involving a small chalk board attached to an Omega Engineering load cell. Lastly, the work that was undertaken to implement open architecture force control with the CataLyst-5 was presented including summary of the challenges that were overcome in conducting the experimentation detailed in the following chapter.

7 Experimental Results

The three position controllers that were developed for the CataLyst-5 were used to control the robot in following the shape positions described in Section 4.1, with primary focus on the five point star. The accuracy of each of these controllers in following the desired trajectory are compared, along with the actuation torques from each controller as an indicator of the control action required.

The two force controllers, as described in Chapter 4, were implemented on the robot with various experimental results collected and discussed. As it could not be configured to maintain contact with the drawing surface, the Hybrid controller was modified with additional experimental results collected. The control signals from the original and modified Hybrid controller and the signal from the Position Based controller are compared and discussed. Next, the error in applied force between the force controllers is investigated. Lastly, the error in end-effector position when following the desired trajectory for each of the force controllers when in contact with the drawing surface is presented and discussed.

Using the same trajectories as in simulation, the position and position/force experimentation results were collected for the case involving the robot arm starting from the ready position, moving to contact the drawing surface (or a designated Z-axis for the position controllers), tracing the star shape then moving back to the ready position. Once again, the order in which the lines for the five point star are drawn is shown in Figure 5.1, where 1 indicates that start (and end) point with the subsequent point that the end-effector moves to in numerical order. All experimental data was gathered with the robot following trajectories that were generated at 50% of the maximum joint and/or Cartesian speeds.

7.1 Comparison of Position Controllers

As discussed in Section 6.7, when the two joint space position controllers were initially implemented on the CataLyst-5 using the optimized gains determined in Section 4.4, excessive chatter at the robot motors and joints level was noticed. Resetting the gains to those determined in Sections 4.2 and 4.3 and conducting some basic gains tuning to improve the end-effector orientation errors resulted in the gains found in Table 7.1 which were used for all experimental data that was collected.

Table 7.1: Joint Space Controller Gains - CataLyst-5

Joint	PD Independent		Joint Linearized	
	k_p	k_d	k_p	k_d
1	70.3	7.03	400	40
2	61.5	6.15	400	40
3	15.4	1.54	400	40
4	5	0.6	500	50
5	2	0.3	500	50

Comparing the joint errors between the two joint space based controllers, as shown in Figure 7.1, it can be seen that the two controllers have a similar error profile when following the star trajectory. Of interest, the simpler PD Independent controller has joint errors that are approximately half the magnitude of those of the Joint Linearized method. Both controllers show the largest error in joint 2 which can be attributed to the effects of gravity causing the joint to sag from the desired position when the arm is extended. This error due to gravity is also observed in joint 3. The cumulative errors in joints 2 through 4 contribute to these controllers' inability to maintain the desired end-effector orientation in pitch. Despite having no external forces applied to the end-effector, joint 5 shows a notable error which contributes to the orientation issues in roll. It is believed that this error is caused by friction within the gearing and chain mechanism that actuates joint 5. As there is no integral action with these controllers, this error is not corrected.

Maintaining the desired end-effector orientation was also an initial deficiency for the Cartesian Space Linearized controller and because proper orientation was critical in using this controller as a foundation for force control, significant work was done to improve this position control method when applied to the actual robot. As discussed in Section 6.7, gains for all of the control axes were increased, with specific attention paid to the first three (of

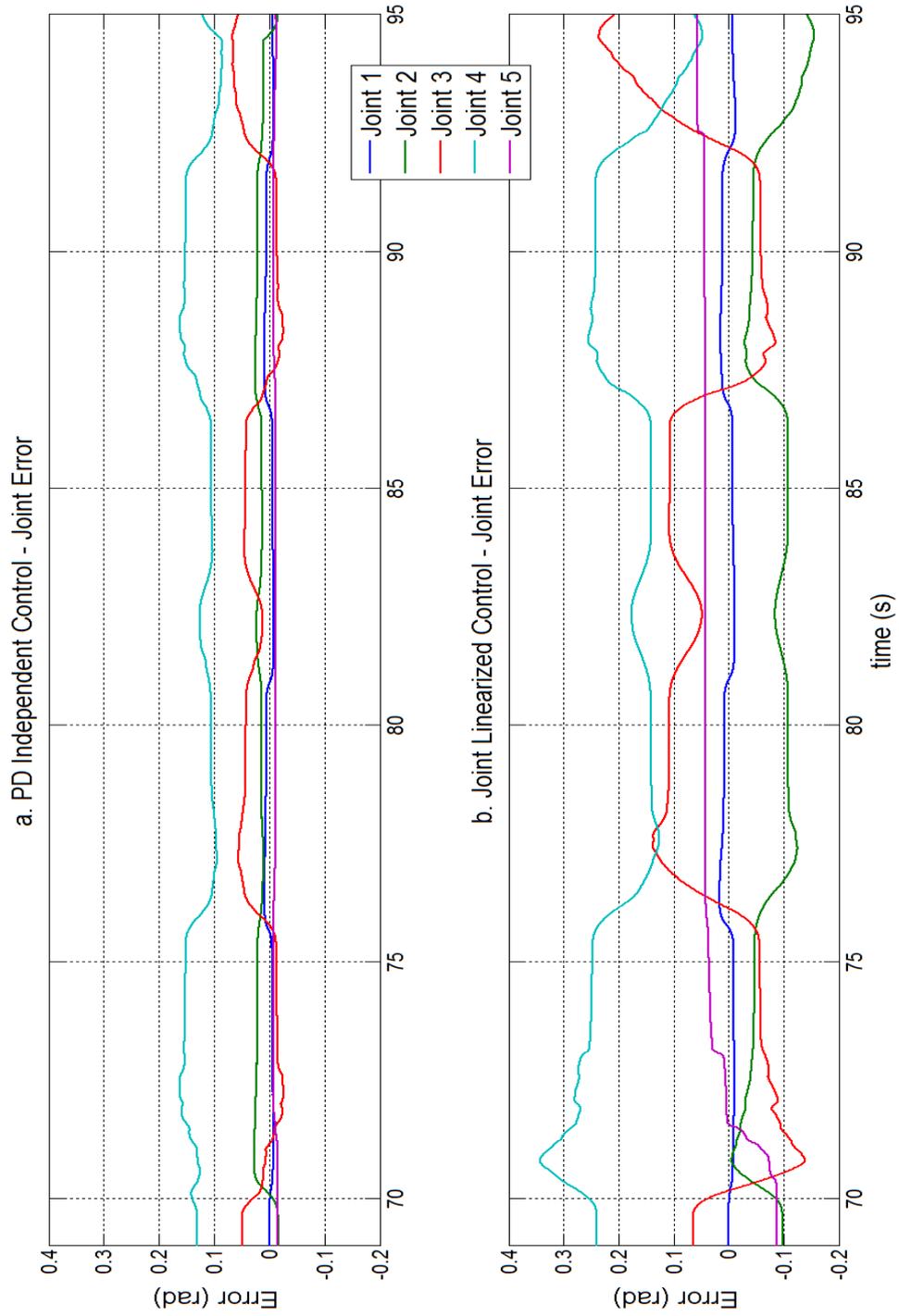


Figure 7.1: Joint Errors - Joint Space Controllers

six) channels that provide a control signal based on errors in the end-effector orientation. That the system can be considered linear, the initial gains for all channels were calculated using the following formulas for linear critically damped second order systems.

$$k_p = \omega_n^2$$

$$k_d = 2\omega_n$$

The gains for all channels were then systematically increased until acceptable control of the position of the end-effector was achieved. The gains for the first two channels were further increased until the desired orientation could be maintained, because they directly control end-effector roll (r) and pitch (β). In addition to increasing the gains, integral control action was added to the first three control channels to help eliminate any steady-state error in the end-effector orientation. The final gains values used with the Cartesian Space controller (and both force controllers) can be found in Table 7.2.

Table 7.2: Cartesian Space Controller Gains - CataLyst-5

	k_p	k_d	k_i
r	6000	80	5
β	6000	80	5
γ	1789	60	5
X	1789	60	
Y	1789	60	
Z	1789	60	

With the larger gains and the integral action added, the Cartesian Space controller, as shown in Figure 7.2a. does a notably better job of maintaining the desired end-effector orientation when compared with the previously discussed joint space controllers. Maximum error in roll (r) when following the star trajectory is approximately 0.01 rad which is comparable to the joint 5 error for the PD Independent controller, but smaller than the 0.04 rad error for the Joint Space Linearized controller. The Cartesian Space controller significantly improved the sag of the end-effector in pitch (β) compared to the other controllers (joint 4), with a negative error value indicating the orientation was held in a slightly positive direction against the effects of gravity. As the error in yaw for the Cartesian controller can be compared directly to the joint 1 error for the joint space controllers it is observed that the more complex controller provides no significant improvement.

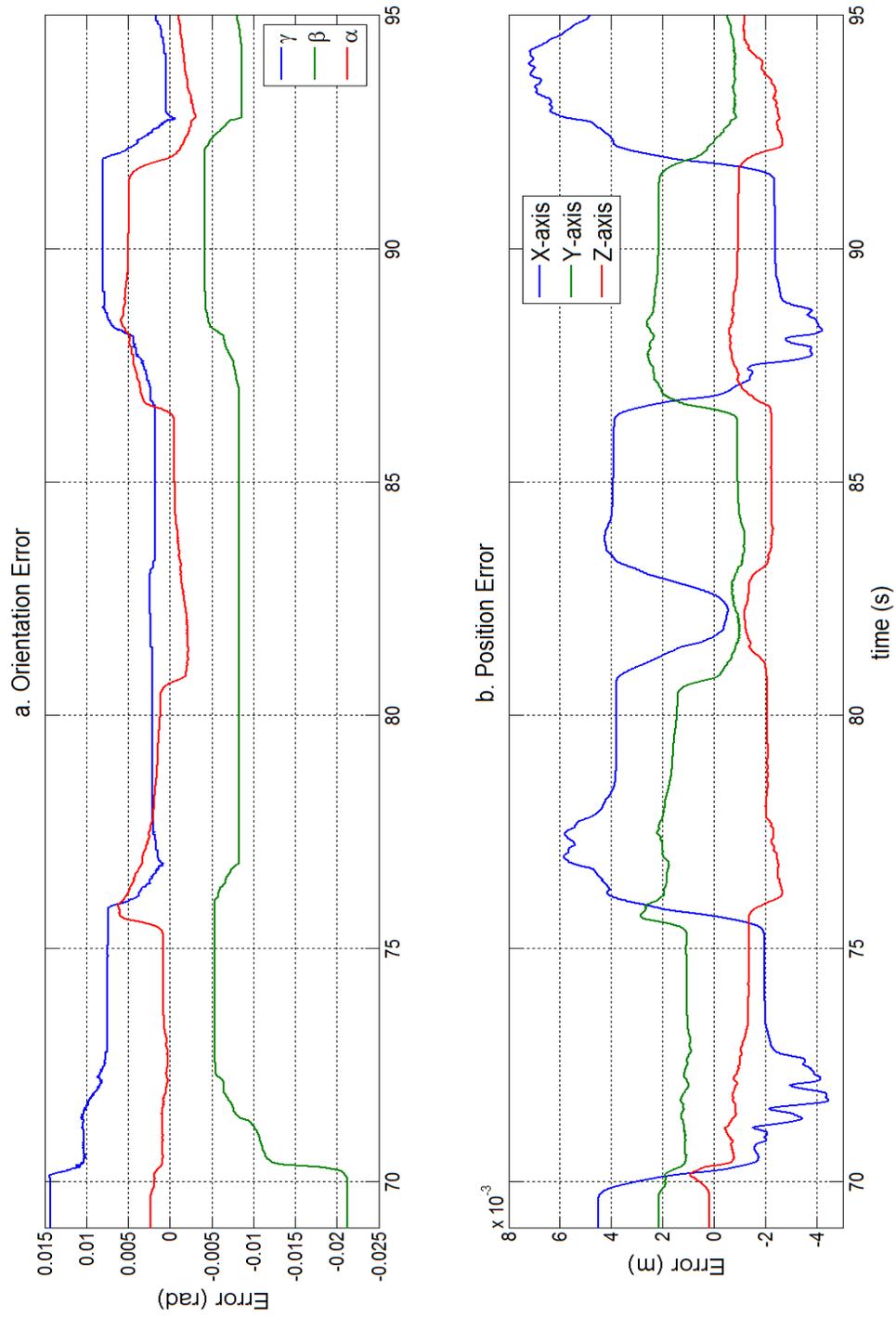


Figure 7.2: Cartesian Space Linearized Controller Errors

The Cartesian Space controller provides acceptable accuracy of the position of the end-effector throughout the star trajectory as shown in Figure 7.2b. with a maximum error in the X direction of approximately 7mm (3.8%)¹ with the average closer to 4mm (2.2%) and the maximum error in the Y direction of 2.5mm (2.5%), though smaller for the majority of the trajectory. The shape of the errors in the X and Y direction indicate that the larger errors occur when the lines of the star are being traced and movement is taking place in both the X and Y axes. The errors could thus be attributed to a slight lag between actual and desired positions. The maximum error in the Z direction, in which the end-effector position is held constant, is -2mm which is due to the effects of gravity on the robot. If required for more precise applications, further refinement of the gains for this controller could help to reduce these errors.

Comparing the control currents from the three position controllers, in Figure 7.3, shows similar shapes and magnitudes from each. Currents from the two joint space controllers are nearly identical with no notable difference between the two. Comparison of the control currents from the Cartesian space controller with the two joint space controllers shows that the latter is able to better control end-effector orientation with more pronounced and slightly faster response for joints 2 and 3. As none of the control currents are near motor saturation, validates that any would be suitable for use for position only control. That the Cartesian Space controller provides greater accuracy for the same levels of applied current justifies the higher complexity of this method.

7.2 Comparison of Force Controllers

As was done with the position controllers, the gains used in the force control portion of each of the position/force controllers were adjusted based on experimental results in an attempt to reduce the error in applied force. The majority of the effort in gains tuning was done with the Hybrid controller; when applied to the actual robot there were significant difficulties maintaining contact with the drawing surface. Too small of a force controller gain would result in an insufficient control signal in the Z direction to maintain contact with the chalk board while the star trajectory was followed. Too large of values for these gains would result in the end-effector rapidly rising off the drawing surface when force control, responding to an initial excessive applied

¹Percentage of total distance end-effector travels in the X-direction when following star trajectory

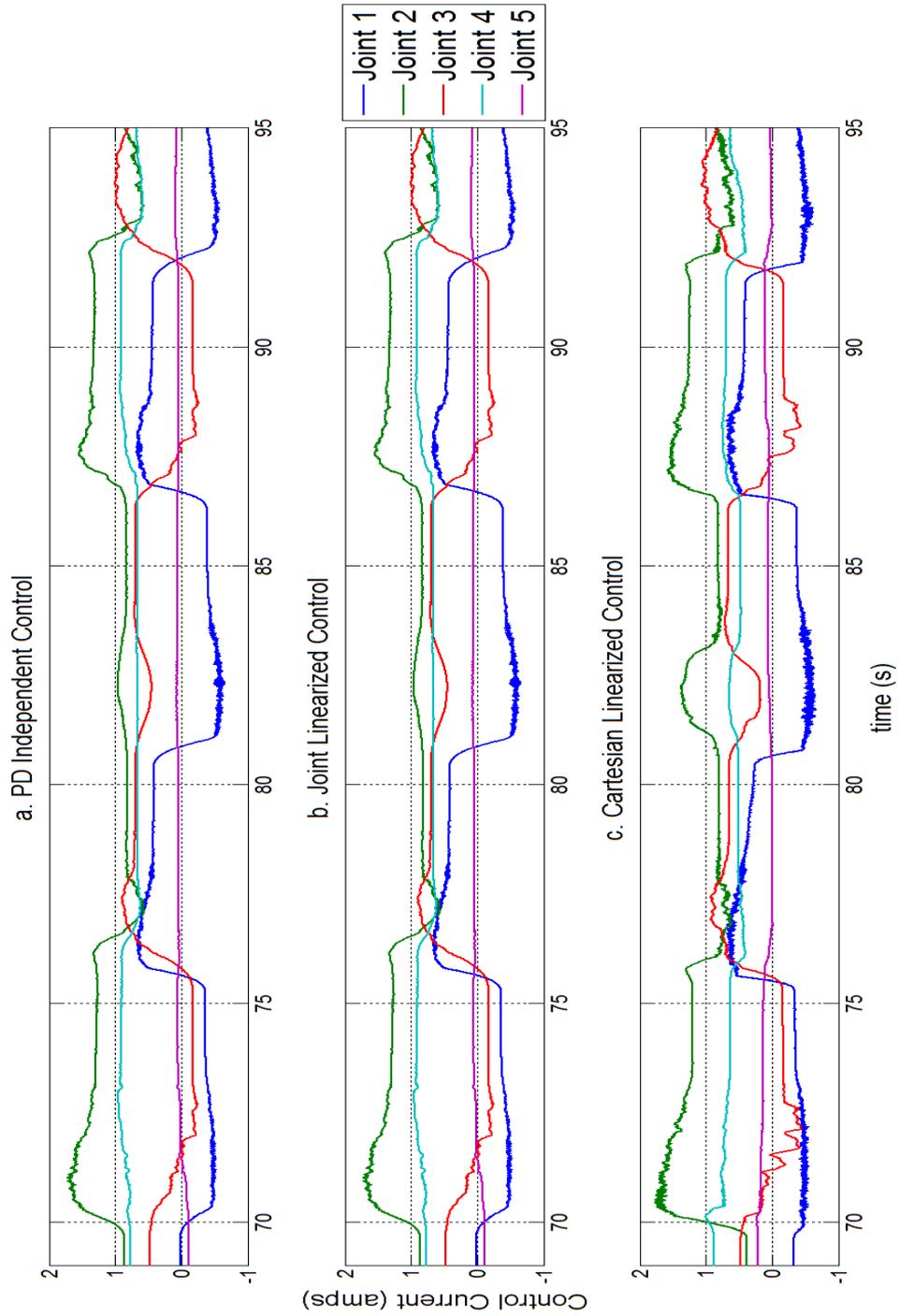


Figure 7.3: Control Currents - Position Controllers

force during transition, was switched on. Significant troubleshooting and trial and error led to the gains found in Table 7.3. These gain values are notably smaller than those determined for the position controllers, though they were found to be the best compromise to allow the Modified Hybrid controller (described below) to maintain contact with the drawing surface.

Table 7.3: Force Controller Gains - CataLyst-5

	Hybrid	Position Based
k_{pf}	16	0.01
k_{if}	1.5	0.001
k_{df}	0.5	0.0001

Adjusting the gains for the Position Based controller proved to be less involved as this method takes advantage of the stability and fast control action of the Cartesian space position controller. Similar loss of contact with the drawing surface was observed with this controller when too small, or too large of gain values were used, though the departure from the drawing surface was always less pronounced. The gain values that were used with this controller, when gathering the experimental data, can be found in Table 7.3. The very small gains values can be attributed to the small control effort required to achieve minor adjustments in the desired Z position of the end-effector. The gains for the position portion of both force control methods are those previously determined for the Cartesian space controller as found in Table 7.2.

Figure 7.4 shows the control signal in each of the Cartesian directions from the Hybrid controller, as described in Section 4.6, when applied to the actual robot. The plot shows that contact is made and control of the end-effector position along the Z-axis switched over to the force controller at the sharp inflection just prior to 45s (circled). Once the force controller is activated, it can be observed that the control signal along the Z-axis starts to oscillate positive and negative with increasingly larger amplitudes. The oscillation stops when the E-stop is pressed on the teach pendant and the control signals stop when the stop is pressed in Simulink.

The results presented for the Hybrid controller in Figure 7.4 is a representative example of the failed attempts to find gains that would make this controller both maintain contact with the drawing surface and apply a suitable force to draw a solid line without breaking the chalk. Once it was determined that it was not possible for the PID force controller to provide the fast acting and large control signals required to maintain the Z position in a near constant position while tracing the desired trajectory, a modified version of the

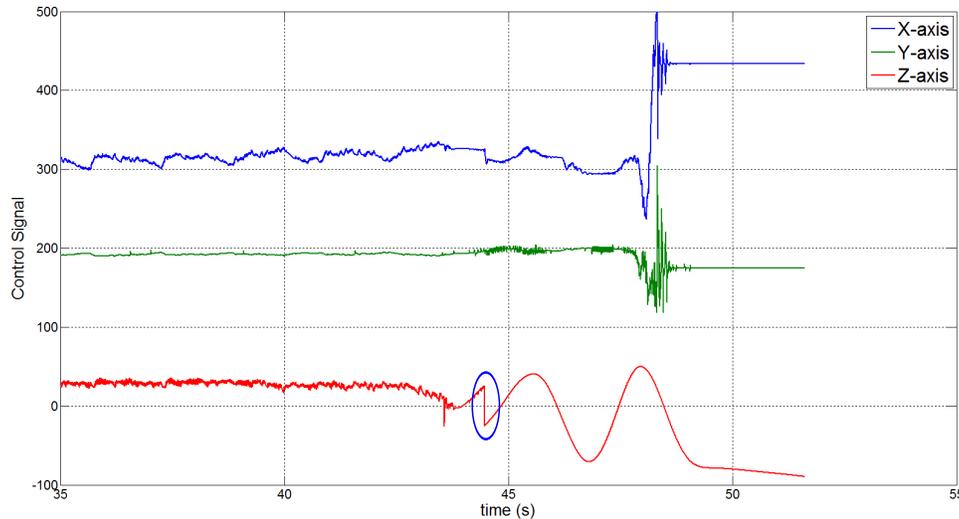


Figure 7.4: Control Signal - Hybrid Position/Force Controller

controller was developed.

The Modified Hybrid controller combines the Z-axis control signal from both the position and force controllers, taking 50% of the former and adding the latter once contact with the drawing surface is made. This modified configuration was found during the extensive troubleshooting of the Hybrid controller, which found that the inherent stability of the Cartesian space controller would ensure rapid loss of contact with the surface would not occur with the force controller being activated. Only 50% of the position control signal is used as this allows the 'weaker' force controller to be able to adjust the Z position of the end-effector slightly according to applied force error, counter to the 'stronger' position controller trying to maintain a set position. This combined control signal along the Z-axis can be seen in Figure 7.5a. which shows the control signal along all three axes when the Modified Hybrid controller is in contact with the drawing surface and following the star trajectory.

The control signals from the modified hybrid controller when compared with those from the Position Based controller, Figure 7.5b., appear to be similar in shape, but with some differences in magnitude for various portions of the trajectory. The differences in control signals along the X and Y axes can be attributed to variations in the force applied by the robot affecting the friction between the chalk and drawing surfaces. The most notable difference between the controllers' X and Y axes signals is that the Y axis signal from the Position Based method has a notably lower numerical value when the

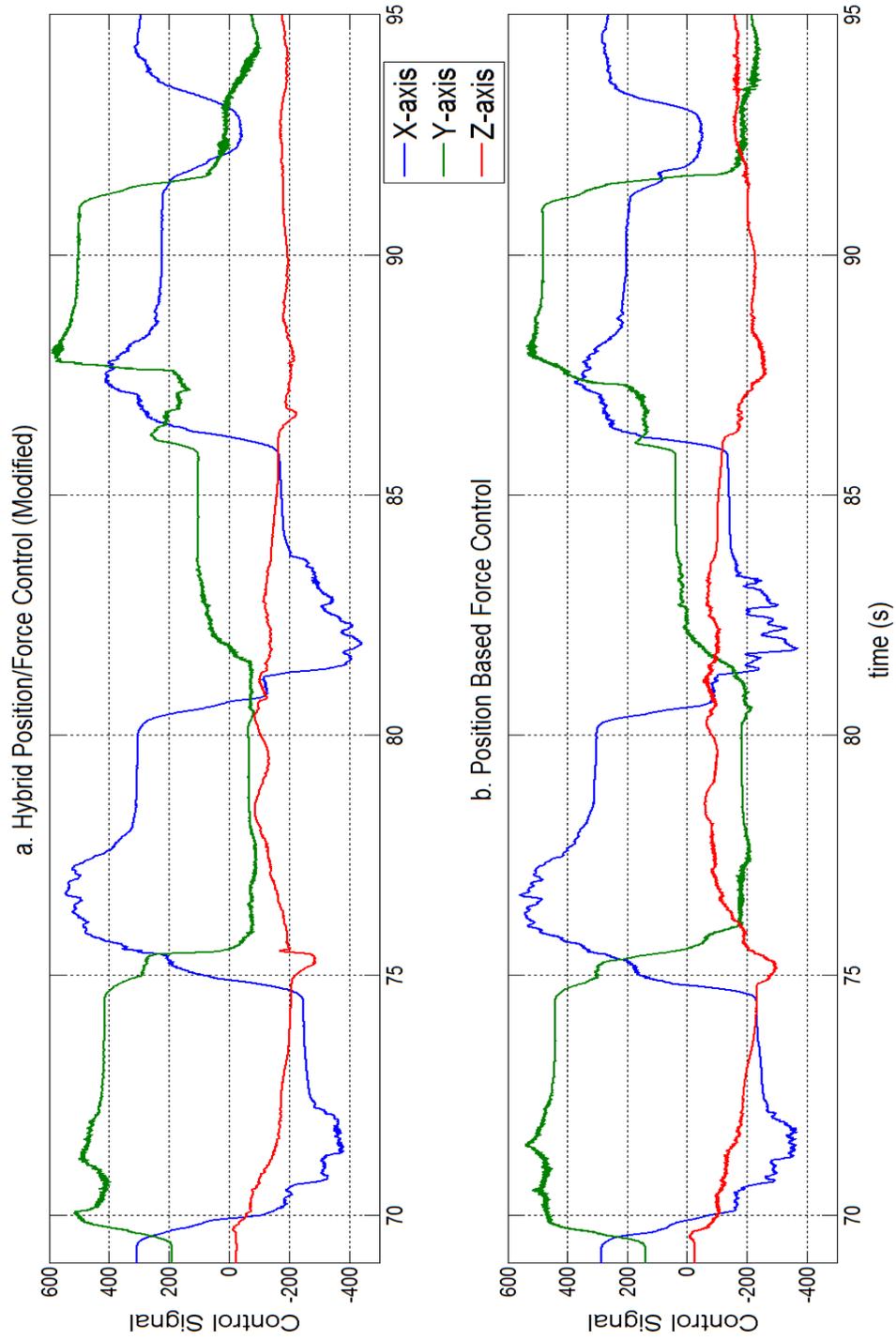


Figure 7.5: Control Signal - Force Controllers

trajectory has the end-effector moving in the negative Y direction (76 to 82s, 83 to 87s and 92 to 95s). This difference directly contributes to the smaller position errors for this controller, which will be discussed later, but is difficult to explain as both methods use the same controller for the X and Y position signal.

The difference in control signals along the Z axis of each of the controllers is small but noticeable. The signal from the Position Based controller holds slightly larger values in key areas where the Modified Hybrid controller loses contact with the surface, as highlighted in Figure 7.6. The two flat sections of the Modified Hybrid applied force error plot indicate brief periods where the end-effector lost contact with the drawing surface. Both periods of lost contact occur when the end-effector is moving away from the base, requiring a positive rotation of joints 3 and to a lesser extent joint 2, which directly leads to lifting chalk from the surface if not adequately compensated for by the force controller.

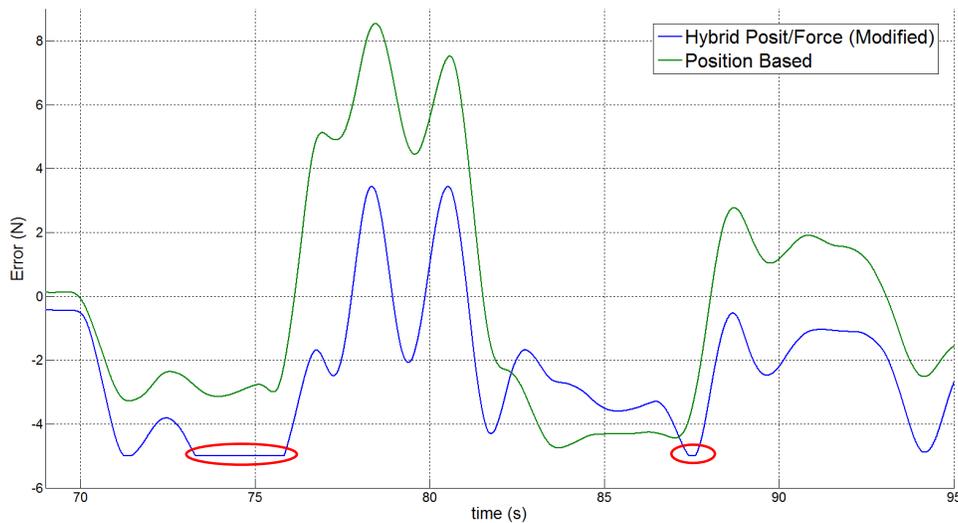


Figure 7.6: Applied Force Error - Position/Force Controllers

The larger control signals and slightly faster response from the Position Based controller, while ensuring contact is maintained, do lead to greater positive applied force errors than the Modified Hybrid method. While this ensured a solid chalk line was traced throughout the star trajectory, it did result in the chalk breaking on occasion. The large variation in force errors for both of the controllers indicate that neither is able to respond with sufficient speed

along the Z-axis to maintain a uniform force for the duration of contact with the drawing surface. While larger gains could theoretically improve the speed of response, experimentally it was found that too large of gains would only result in greater and in some cases more violent loss of contact. Investigation into more advanced control methods for the force control loop of each of these controllers could result in a more even applied force on the contact surface. Application of more advanced control methods could also make the original Hybrid controller functional.

Comparison of the position errors between the Modified Hybrid and Position Based controllers shows that the latter provides notably better trajectory following performance along both the X and Y axes as shown in Figure 7.7. The difference could be attributed to the force control loop working internal to the position controller with the Position Based method, vice attempting to work against the position control signal with the Modified Hybrid method. Though it applies greater than desired force, the Position Based Force controller proved to be the better of the two methods as it was able to maintain contact with the drawing surface for the duration of the star trajectory and followed the trajectory more accurately than the Modified Hybrid method.

7.3 Conclusion

The experimental results collected when controlling the physical CataLyst-5, following the star trajectory, with the three position and two force control methods that were developed are presented. It was explained that excessive chatter from the robot's motors and joints required base level gains to be used with the two joint space level position controllers. The smaller gains led to larger errors from these controllers. The adjustment of the gains for the Cartesian Space Linearized controller and the subsequent accurate position following results were discussed. Efforts made to find suitable gains to allow the Hybrid controller to maintain contact with the drawing surface were detailed followed by an explanation of the modified Hybrid control method that was developed as a partial remedy to the deficiency. Lastly, the greater accuracy of the Position Based Force controller compared to the Modified Hybrid in both maintaining the desired contact force while simultaneously following the star trajectory was discussed.

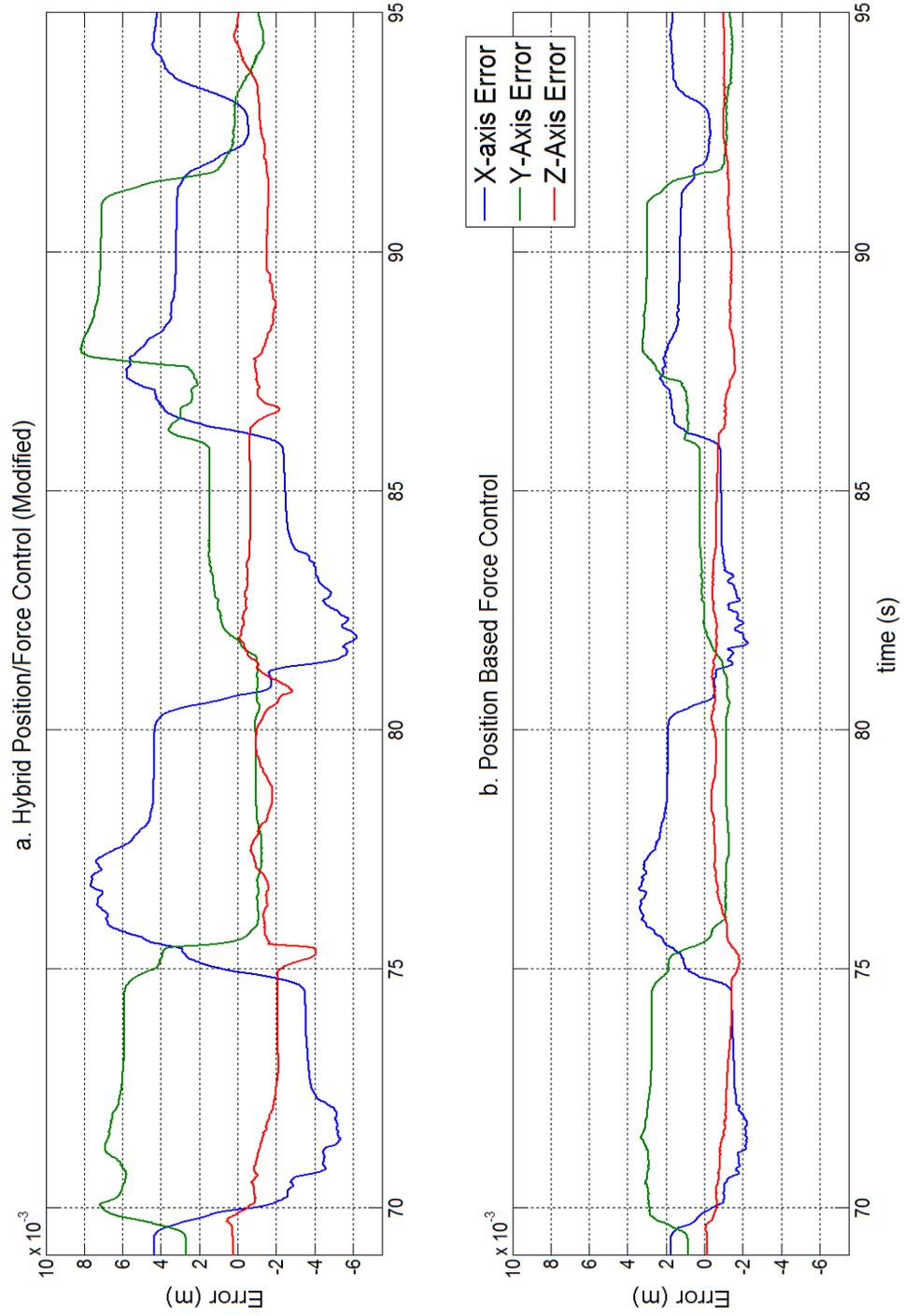


Figure 7.7: Position Error - Position/Force Controllers

8 Conclusion

In this thesis, research was conducted on modelling and control system design for use with the CataLyst-5 five degrees of freedom articulated manipulator to allow it to come into contact with the environment and apply a controlled force. The modeling consisted of the development of detailed solutions to the direct and inverse kinematic problems for the robot as well as simplified task specific solutions that improved simulation and experimental operations. The detailed robot arm dynamics were determined using the Lagrange method with the mathematical equations generated, used within Simulink to create a model of the CataLyst-5 for use when simulating the designed controllers. As validation and comparison with the mathematical equation based model, a second model was created using SimMechanics. With both models proving equivalent, the second method was selected for all simulation work as it more readily allowed the simulation of contact with the environment. The mathematical equations representing the robot arm dynamics were used extensively within the control methods that were developed as part of this research.

Three different control methods were developed for controlling the position of the CataLyst-5, each increasingly more complex and as might be expected more accurate. As it can be considered the industry standard, the first position control method that was developed was a simple independent joint space PD controller that made use of the high gear ratios used within the robot's joints to treat each joint as a separate linear control problem, with the non-linearities and coupling effects due to movements of the other joints being considered as external disturbances. This controller provided fast and relatively accurate position control of the robot arm, with the majority of the observed inaccuracy due to the effects of gravity which could be compensated for with minor modifications to the controller.

The second method developed for position control was the joint space feedback linearized controller (also known as computed torque control) that made use of the previously determined dynamics of the robot arm to linearize and decouple the system. Once linearized, the robot could be readily controlled us-

ing simple proportional and derivative action with notably improved accuracy and any errors attributed to a slight lag in ordered and actual position.

Detailed comparison in simulation between the first two position controllers determined that the PD Independent controller could provide the most rapid movements of individual joints as it required slightly less actuation currents than the Joint Space Linearized controller. Extensive work to optimize the gains and maximum joint speeds for each of these controllers resulted in values that could not be used with the actual robot experimentally, requiring more modest gains to be used experimentally. Comparison of the experimental results of the first two position control methods resulted in smaller errors for the PD Independent controller when using smaller control gains. Additional time spent tuning the gains for each of these controllers, when applied to the actual robot, would likely result in the linearized controller providing a better performance.

The third and most complex position controller that was developed and applied both in simulation and experimentally was the Cartesian Space Linearized controller. This method made use of the robot dynamics in Cartesian space to both linearize and decouple the robot while also allowing control effort be determined as if a vector of forces and torques could be applied at the end-effector to achieve the desired movement. Substantially more mathematically complex, requiring the inverse of the manipulator's non-square Jacobian, this controller provided the most accurate position control of the three methods in both simulation and experimentation. The longer simulation and controller build times with this method made troubleshooting and optimization challenging, but the accurate control results greatly justified the effort.

The Cartesian Space Linearized position controller also provided the foundation for the two force controllers that were developed for and used with the CataLyst-5. The first position/force controller that was developed was a traditional Hybrid Position/Force controller that would selectively control the applied force (and moment) along selected axes while controlling position (and orientation) along the remaining axes. Despite the controller being proven in simulation, it could not be effectively applied with the actual robot due to the significant control action along the Z-axis of the end-effector required to maintain a steady position in this direction. In order to gather some basic experimental data using hybrid control, a modified method was developed that made use of 50% of the control signal from the position controller along the Z-axis combined with the full control signal from the force controller. This Modified Hybrid controller was better able to maintain contact with the chalk board drawing surface that was used experimentally, though with a few brief periods where contact was lost.

The second position/force controller that was developed involved modifying the desired position of the end-effector along the Z-axis based on errors in the force applied to the drawing surface. The Position Based Force controller made use of the fast and stable control provided by the Cartesian Space Linearized controller to provide more accurate force control than the traditional hybrid control method in simulation. Experimentally the position based method maintained contact with the drawing surface throughout the desired trajectory, but did apply a larger than desired force during portions of the trajectory.

The difficulty both controllers encountered in maintaining contact with the drawing surface and/or accurately controlling the contact force was determined to be due to the physical construction of the CataLyst-5. The articulated robot arm employs joints 2 through 4 to both control the vertical and radial position of the end-effector, often with conflicting results requiring significant control action along the Z-axis just to maintain a constant Z position while the end-effector is moved in the X and Y directions. Great difficulty was encountered in creating force control loops with sufficiently quick response times to provide the necessary level of control.

As it made use of the large gains and stable operation of the Cartesian Space Linearized controller, the Position Based Force controller proved to be the best method in overcoming these manipulator deficiencies to control force along a single axis while following a desired trajectory in the remaining axes. Further development of this method could see its application with the simpler joint space controllers. More advanced control methods used for force control loop could greatly improve the ability of each of the position/force controllers to maintain contact while also reducing the error in applied force.

9 Recommendations for Future Work

Building on what has been completed in this research, a number of areas for additional work could be considered. As they were not intended for use as position/force controllers, limited experimental work was completed on the PD Independent and Joint Space Linearized controllers, resulting in unacceptable trajectory following errors for both when conducting free motion movements. A number of improvements could be made to both of these controllers. Feed-forward gravity compensation could be developed for use with the PD Independent controller which would have the effect of limiting or removing the steady state error due to gravity induced sag in joints 2 through 4. The addition of integral control action to this controller could also help to counter this error, though careful selection of the gains would be required to ensure the controller remained stable.

Similarly, integral action could be added to the Joint Space Linearized controller to remove the smaller errors that were found when using this controller. The easiest, though potentially most time consuming method of improving the performance of these controllers would come from optimizing the gains used in experimentation. The optimal gains selected in simulation proved too aggressive for use with the physical robot, though time spend studying individual joints experimentally could greatly improve the controller's performance.

Sufficient improvement in the trajectory following performance of the two more simple position controllers could allow them to be used as a basis for a position based force control strategy similar to what was developed in Section 4.7. As both of these controllers work in joint space, using them as a position based force controller would require trajectory generation in real time, adding a level of model complexity that may negate any benefits of using the simpler position control methods.

The primary deficiency with both of the force control methods investigated

was that the speed and magnitude of the force control signal was inadequate to either maintain contact or adequately control the applied force. Research into a more rapid method of providing the force control signal could allow for the controllers designed in this thesis to be greatly improved without significant modification.

For this research, force was only controlled along one end-effector degree of freedom. Future research could look at control the applied force (and torque) along any of the end-effector axes. A more complex controller might also involve controlling the end-effector orientation so as to be perpendicular to the contact surface. To do further research in this direction, a working force/torque sensor would have to be procured or the one described in this research repaired. Also, the contact model created in SimMechanics would have to be expanded to include either a varying contact surface shape in the Z-axis, or contact along the other axes. Similarly, a new physical contact environment would have to be created to trial the more advanced position/force controllers.

The only version of the Hybrid Position/Force controller that could be configured to maintain reasonable contact with the drawing surface was the modified version that combined the position and force control signal along the Z-axis. Further research could be conducted into a similar adapted version of Hybrid control that makes use of advanced methods like fuzzy control to take varying portions of both the position and force control signal and combine them to more accurately control the applied force along the desired axis.

Research into using other sensing methods to allow for more controlled transition could improve upon the overall position/force control using the CataLyst-5. Even when the contact surface was approached at a very slow speed, the initial contact force would often exceed the desired force by over 100% before the contact detection would cease the position based movement. Robot vision monitoring the position of the end-effector with respect to the contact surface, or a proximity sensor mounted on the end-effector could be used to help reduce the excessive initial contact force when transition from position to force control.

Bibliography

- [1] F. Caccavale, C. Natale, B. Siciliano, and L. Villani. Integration for the next generation: embedding force control into industrial robots. *Robotics Automation Magazine, IEEE*, 12(3):53–64, Sept 2005.
- [2] Bruno Siciliano and Luigi Villani. Robotics: Vision, manipulation and sensors. In Takeo Kanade, editor, *Robot Force Control*, volume 540 of *The Springer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, 1999.
- [3] R.P. Paul. Problems and research issues associated with the hybrid control of force and displacement. In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, volume 4, pages 1966–1971, March 1987.
- [4] Joris De Schutter, Herman Bruyninckx, Wen-Hong Zhu, and MarkW. Spong. Force control: A bird’s eye view. In Bruno Siciliano and KimonP. Valavanis, editors, *Control Problems in Robotics and Automation*, volume 230 of *Lecture Notes in Control and Information Sciences*, pages 1–17. Springer Berlin Heidelberg, 1998.
- [5] T. Yoshikawa. Force control of robot manipulators. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 1, pages 220–226 vol.1, 2000.
- [6] Jianjun Wang, G. Zhang, Hui Zhang, and T. Fuhlbrigge. Force control technologies for new robotic applications. In *Technologies for Practical Robot Applications, 2008. TePRA 2008. IEEE International Conference on*, pages 143–149, Nov 2008.
- [7] N. Hogan. On the stability of manipulators performing contact tasks. *Robotics and Automation, IEEE Journal of*, 4(6):677–686, Dec 1988.
- [8] Ray C. Goertz. Fundamentals of general purpose remote manipulators. *Nucleonics*, 10:36–42, 1952.

-
- [9] J. Maples and J. Becker. Experiments in force control of robotic manipulators. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 695–702, Apr 1986.
- [10] Matthew T. Mason. Compliance and force control for computer controlled manipulators. *Systems, Man and Cybernetics, IEEE Transactions on*, 11(6):418–432, June 1981.
- [11] Simeon P. Patarinski and Roumen G. Botev. Robot force control: A review. *Mechatronics*, 3(4):377 – 398, 1993.
- [12] RA Rothchild and RW Mann. An emg controlled, force sensing, proportional rate, elbow prosthesis. In *Proc. Symp. on Biomedical Engineering*, 1966.
- [13] Daniel E. Whitney. Historical perspective and state of the art in robot force control. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 262–268, Mar 1985.
- [14] Lorenzo Sciavicco and Bruno Siciliano. *Modeling and Control of Robot Manipulators*. The McGraw-Hill Companies, Inc., 1996.
- [15] Mark Spong, Seth Hutchinson, and M Vidyasgar. *Robot Modelling and Control*. John Wiley & Sons, Inc., 2006.
- [16] Marc Arsenault. Robot mechanics. Graduate Course Notes, January 2013.
- [17] R. Volpe and P. Khosla. An experimental evaluation and comparison of explicit force control strategies for robotic manipulators. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 1387–1393 vol.2, May 1992.
- [18] Dimitry M. Gorinevsky, Alexander M. Formalsky, and Anatoly Y. Schneider. *Force Control of Robotics Systems*. CRC Press LLC, 1997.
- [19] T. Tarn, Yunging Wu, Ning Xi, and A Isidori. Force regulation and contact transition control. *Control Systems, IEEE*, 16(1):32–40, Feb 1996.
- [20] Chae H. An and J.M. Hollerbach. Dynamic stability issues in force control of manipulators. In *American Control Conference, 1987*, pages 821–827, June 1987.
- [21] R. Volpe and P. Khosla. Experimental verification of a strategy for impact control. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 1854–1860 vol.2, Apr 1991.

-
- [22] J.K. Mills and D.M. Lokhorst. Stability and control of robotic manipulators during contact/noncontact task transition. *Robotics and Automation, IEEE Transactions on*, 9(3):335–345, Jun 1993.
- [23] J.K. Salisbury. Active stiffness control of a manipulator in cartesian coordinates. In *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*, volume 19, pages 95–100, Dec 1980.
- [24] Miomir K. Vukobratovic and Veljko Potkonjak. Dynamics of contact tasks in robotics. part i: General model of robot interacting with environment. *Mechanism and Machine Theory*, 34(6):923 – 942, 1999.
- [25] John J. Craig. *Introduction to Robotics, Mechanics and Control*. Pearson / Prentice Hall, second edition, 2005.
- [26] S.D. Eppinger and W.P. Seering. Introduction to dynamic models for robot force control. *Control Systems Magazine, IEEE*, 7(2):48–52, April 1987.
- [27] K. Khayati, P. Bigras, and L-A Dessaint. A multistage position/force control for constrained robotic systems with friction: Joint-space decomposition, linearization, and multiobjective observer/controller synthesis using lmi formalism. *Industrial Electronics, IEEE Transactions on*, 53(5):1698–1712, Oct 2006.
- [28] D.P. Stoten. *Model Reference Adaptive Control of Manipulators*. John Wiley & Sons, Inc., 1990.
- [29] Daniel E. Whitney. Force feedback control of manipulator fine motions. *Journal of Dynamic Systems, Measurement, and Control*, 99(2):91–97, June 1977.
- [30] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, June 1981.
- [31] Frank Lewis, Darren Dawson, and Chaouki Abdallah. *Robot Manipulator Control, Theory and Practice*. Marcel Dekker, Inc, second edition edition, 2004.
- [32] N. Hogan. Impedance control: An approach to manipulation. In *American Control Conference, 1984*, pages 304–313, June 1984.
- [33] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *Robotics and Automation, IEEE Journal of*, 3(1):43–53, February 1987.

-
- [34] T. Yoshikawa. Dynamic hybrid position/force control of robot manipulators—description of hand constraints and calculation of joint driving force. *Robotics and Automation, IEEE Journal of*, 3(5):386–392, October 1987.
- [35] S. Chiaverini and L. Sciavicco. The parallel approach to force/position control of robotic manipulators. *Robotics and Automation, IEEE Transactions on*, 9(4):361–373, Aug 1993.
- [36] Ganwen Zeng and Ahmad Hemami. An overview of robot force control. *Robotica*, 15:473–482, 9 1997.
- [37] C. Natale, R. Koeppel, and G. Hirzinger. A systematic design procedure of force controllers for industrial robots. *Mechatronics, IEEE/ASME Transactions on*, 5(2):122–131, Jun 2000.
- [38] R. Anderson and M.W. Spong. Hybrid impedance control of robotic manipulators. In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, volume 4, pages 1073–1080, Mar 1987.
- [39] Yuan-Fang Zheng and Hooshang Hemami. Mathematical modeling of a robot collision with its environment. *Journal of Robotic Systems*, 2(3):289–307, 1985.
- [40] Harry West and H. Asada. A method for the design of hybrid position/force controllers for manipulators constrained by contact with the environment. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 251–259, Mar 1985.
- [41] O. Khatib and J. Burdick. Motion and force control of robot manipulators. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 1381–1386, Apr 1986.
- [42] R.E. Goddard, Y.F. Zheng, and H. Hemami. Dynamic hybrid velocity/force control of robot compliant motion over globally unknown objects. *Robotics and Automation, IEEE Transactions on*, 8(1):132–138, Feb 1992.
- [43] E. Bassi, F. Benzi, L.M. Capisani, D. Cuppone, and A Ferrara. Hybrid position/force sliding mode control of a class of robotic manipulators. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 2966–2971, Dec 2009.
- [44] J. Velagic, A Adzemovic, and J. Ibrahimagic. Switching force/position fuzzy control of robotic manipulator. In *Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings. 2003 IEEE/ASME International Conference on*, volume 1, pages 484–489 vol.1, July 2003.

-
- [45] Il-Hong Suh, Jong Hyuck Hong, Sang-Rok Oh, and Kwang-Bae Kim. Fuzzy rule based position/force control of industrial manipulators. In *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/R SJ International Workshop on*, pages 1617–1622 vol.3, Nov 1991.
- [46] Shih-Tin Lin and Ang-Kiong Huang. Hierarchical fuzzy force control for industrial robots. *Industrial Electronics, IEEE Transactions on*, 45(4):646–653, Aug 1998.
- [47] Ming-Shaung Ju, C.-C.K. Lin, Dong-Huang Lin, I-S. Hwang, and Shu-Min Chen. A rehabilitation robot with force-position hybrid fuzzy controller: hybrid fuzzy control of rehabilitation robot. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 13(3):349–358, Sept 2005.
- [48] Kazuo Kiguchi and T. Fukuda. Fuzzy neural friction compensation method of robot manipulation during position/force control. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 1, pages 372–377 vol.1, Apr 1996.
- [49] D. Jeon and M. Tomizuka. Learning hybrid force and position control of robot manipulators. *Robotics and Automation, IEEE Transactions on*, 9(4):423–431, Aug 1993.
- [50] P. Lucibello. A learning algorithm for hybrid force control of robot arms. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 654–658 vol.1, May 1993.
- [51] Feng-Yih Hsu and Li-Chen Fu. Adaptive fuzzy hybrid force/position control for robot manipulators following contours of an uncertain object. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 3, pages 2232–2237 vol.3, Apr 1996.
- [52] S.W. Kim, E.T. Kim, and M. Park. A new adaptive fuzzy controller using the parallel structure of fuzzy controller and its application. *Fuzzy Sets and Systems*, 81(2):205 – 226, 1996.
- [53] Miomir K. Vukobratovic and Yuri Ekalo. New approach to control of robotic manipulators interacting with dynamic environment. *Robotica*, 14:31–39, 1 1996.
- [54] N. Houshangi. Constrained motion control of a robot manipulator. In *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, volume 4, pages 3495–3500 vol.4, Oct 1998.

-
- [55] Wim Meeussen, Ernesto Staffetti, Herman Bruyninckx, Jing Xiao, and Joris De Schutter. Integration of planning and execution in force controlled compliant motion. *Robotics and Autonomous Systems*, 56(5):437 – 450, 2008.
- [56] Kazuo Kiguchi and T. Fukuda. Position/force control of robot manipulators for geometrically unknown objects using fuzzy neural networks. *Industrial Electronics, IEEE Transactions on*, 47(3):641–649, Jun 2000.
- [57] Feng-Yi Hsu and Li-Chen Fu. Intelligent robot deburring using adaptive fuzzy hybrid position/force control. *Robotics and Automation, IEEE Transactions on*, 16(4):325–335, Aug 2000.
- [58] Lei Wang, Yongping Hao, and Fei Wang. The experiment research of force control based on intelligent prediction in unknown environment. In *Technology and Innovation Conference, 2006. ITIC 2006. International*, pages 2047–2052, Nov 2006.
- [59] Jorge Angeles. *Fundamentals of Robotic Mechanical Systems*. Springer, 2007.
- [60] Thermo CRS. *CataLyst-5 Robot System User Guide*, umi-cat5-400 edition.
- [61] Jacques Denavit. A kinematic notation for lower-pair mechanisms based on matrices. *Trans. of the ASME. Journal of Applied Mechanics*, 22:215–221, 1955.
- [62] Specifications for architects - surface thermosetting acrylic steel.
- [63] Quanser. Open-architecture control manual.
- [64] Quanser/Thermo Electron. *CRS CataLyst-5T Robot System - Open-Architecture Control Manual*, revision: 1.2 edition, 2014.
- [65] Quanser. Quanser real-time control software (quarc) manual. http://www.quarcservice.com/ReleaseNotes/files/release_notes_index.html, 2014. [Online; accessed 16 Dec 2014].
- [66] Michael R. Strawson. Impedance control of a manipulator using a fuzzy model reference learning controller. Master’s thesis, Royal Military College of Canada, 2006.
- [67] Assurance Technologies Inc. *Installation and Operations Manual for F/T*, pn 9610-05-1001-07 edition.

Appendices

A Geometric Solution to the DKP and IKP

A.1 Solution to the DKP

Referring to Figure A.1, the vector position of P can be found by working along the serial arm chain and consecutively finding the vector positions, from the base reference frame (\mathcal{F}_1), of reference points A, B, C, D, and finally P, which aside from D coincide with the origin of the reference frames assigned in Section 3.1. Figure A.1 also shows a useful property of the Catalyst-5 robot, that it can be considered planar from point A to point D. As a planar manipulator, the reference points between A and D can be fully defined in cylindrical coordinate system that corresponds to a radial component given by R_i , an angular coordinate given by θ_1 a vertical component along the Z-axis.

By observation the location of A does not vary by joint angle and is found a length b_1 along the Z_1 -axis giving the vector location of point A in Cartesian coordinates as:

$$\vec{A} = \begin{bmatrix} 0 \\ 0 \\ b_1 \end{bmatrix} \quad (\text{A.1})$$

The radial distance (R) of point B from the Z-axis is found by multiplying the link length (a_2) by the cosine of θ_2 . This radial distance is then multiplied by the cosine of θ_1 to find the X component and sine of θ_1 to find the Y component as shown in Figure A.2, which shows a top view of the planar portion of the mechanism and specifically the radial position of the reference points. The Z component of the position vector of point is found by multiplying the link length (a_2) by the sine of θ_2 and adding a length b_1 to account for

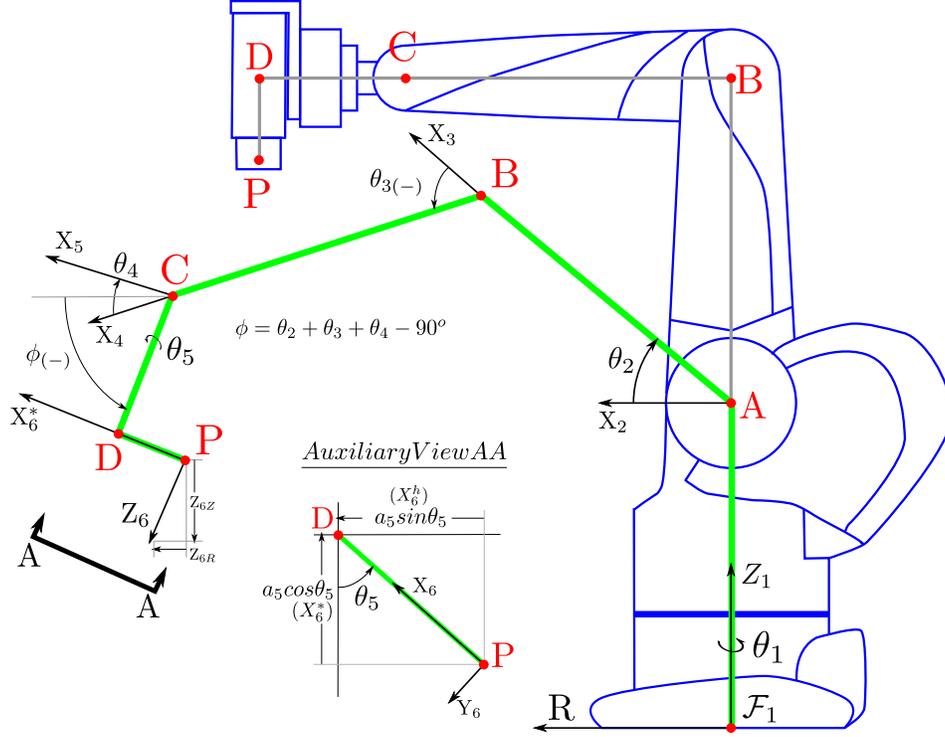


Figure A.1: Planar Mechanism View for DKP/IKP Solution

the position relative to point A. The vector location of point B in Cartesian coordinates is given by:

$$\vec{B} = \begin{bmatrix} a_2 \cos \theta_2 \cos \theta_1 \\ a_2 \cos \theta_2 \sin \theta_1 \\ b_1 + a_2 \sin \theta_2 \end{bmatrix} \quad (\text{A.2})$$

Similarly, the radial distance (R) of points C and D from the Z-axis is found by multiplying an equivalent link length (b_5 for point D) by the cosine of the angle between the link and a horizontal plane parallel to the X-Y plane. For point D, the angle between the link and the horizontal is defined by ϕ shown in Figure A.1 and determined from Eq. A.5. The radial distance is then multiplied by the cosine of θ_1 to find the X component and sine of θ_1 to find the Y component. The Z component of the position vectors of points C and D is found by multiplying the equivalent link length by the sine of the angle between the link and a horizontal plane parallel to the X-Y plane and

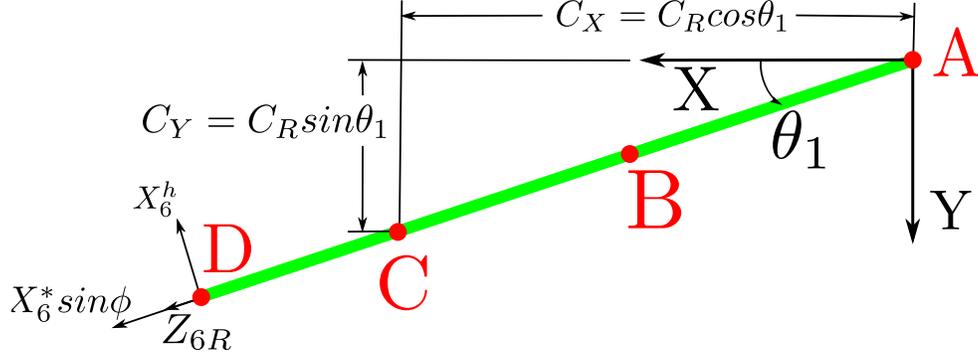


Figure A.2: Planar Robot Top View

adding a length b_1 to account for the position relative to point A. The vector location of points C and D in Cartesian coordinates are given by:

$$\vec{C} = \begin{bmatrix} (a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3)) \cos \theta_1 \\ (a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3)) \sin \theta_1 \\ b_1 + a_2 \sin \theta_2 + a_3 \sin(\theta_2 + \theta_3) \end{bmatrix} \quad (\text{A.3})$$

$$\vec{D} = \begin{bmatrix} (a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3) + b_5 \cos \phi) \cos \theta_1 \\ (a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3) + b_5 \cos \phi) \sin \theta_1 \\ b_1 + a_2 \sin \theta_2 + a_3 \sin(\theta_2 + \theta_3) + b_5 \sin \phi \end{bmatrix} \quad (\text{A.4})$$

$$\phi = \theta_2 + \theta_3 + \theta_4 - 90^\circ \quad (\text{A.5})$$

The vector position of point P can be found by building on the location of point D and noting a vector from P to D can be broken into 2 components as per the auxiliary view of Figure A.1. The X_6^* component ($a_5 \cos \theta_5$) contributes to the radial distance of P from the Z-axis when multiplied by $\sin \phi$ and the vertical distance from the X-Y plane when multiplied by $\cos \phi$. The X_6^h component of the PD vector ($a_5 \sin \theta_5$) is always perpendicular to the Z-axis and contributes to the X and Y values of P depending on the value of θ_1 as is observed in Figure A.2. The vector location of point P and the position portion of the DKP solution is:

$$\vec{P} = \begin{bmatrix} [a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3) + b_5 \cos \phi + |a_5| \cos \theta_5 \sin \phi] \cos \theta_1 - |a_5| \sin \theta_5 \sin \theta_1 \\ [a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3) + b_5 \cos \phi + |a_5| \cos \theta_5 \sin \phi] \sin \theta_1 + |a_5| \sin \theta_5 \cos \theta_1 \\ b_1 + a_2 \sin \theta_2 + a_3 \sin(\theta_2 + \theta_3) + b_5 \sin \phi - |a_5| \cos \theta_5 \cos \phi \end{bmatrix} \quad (\text{A.6})$$

This solution was compared numerically to the more complex version found at Eq. 3.4 and found to provide the same solution for all possible joint angles.

In determining a more compact DKP solution to the end-effector's orientation it is important to note, as shown in Eq. A.7, that the rotation matrix provides the base frame unit vector coordinates of the end-effector reference frame axes. This allows some of the components of Q to be readily determined by observation. First, the components of $|\vec{Z}_6|$ don't vary with θ_5 , only with ϕ and θ_1 . As can be seen in Figure A.1, $Z_{6Z} = |\vec{Z}_6| \sin \phi$ and $Z_{6R} = |\vec{Z}_6| \cos \phi$. As can be observed in Figure A.2, the Z_{6R} component when multiplied by $\cos \theta_1$ gives Z_{6X} and when multiplied by $\sin \theta_1$ gives Z_{6Y} .

$$Q = \begin{bmatrix} X_{6X} & Y_{6X} & Z_{6X} \\ X_{6Y} & Y_{6Y} & Z_{6Y} \\ X_{6Z} & Y_{6Z} & Z_{6Z} \end{bmatrix} \quad (\text{A.7})$$

As previously described and as shown in the auxiliary view of Figure A.1, \vec{X}_6 (and similarly \vec{Y}_6) can be broken down into components based on θ_5 . These components are X_6^* as shown on the main diagram and X_6^h that is always parallel to the X-Y plane regardless of the value of θ_5 and thus does not contribute to X_{6Z} . This leads to the observation that $X_{6Z} = X_6^* \cos \phi = |\vec{X}_6| \cos \theta_5 \cos \phi$. The two horizontal component of X_6^* ($X_6^* \sin \phi$ and X_6^h) contribute to the X and Y elements of \vec{X}_6 based on the value of θ_1 as can be observed in Figure A.2. The X and Y components of \vec{X}_6 are:

$$\begin{aligned} X_{6X} &= -|\vec{X}_6| \cos \theta_5 \sin \phi \cos \theta_1 + |\vec{X}_6| \sin \theta_5 \sin \theta_1 \\ X_{6Y} &= -|\vec{X}_6| \cos \theta_5 \sin \phi \sin \theta_1 - |\vec{X}_6| \sin \theta_5 \cos \theta_1 \end{aligned} \quad (\text{A.8})$$

The three components of the unit vector providing the orientation of \vec{Y}_6 are found in a similar manner to \vec{X}_6 , but with care taken regarding differing signs of Y_6^* and Y_6^h . The components of \vec{Y}_6 are:

$$\begin{aligned} Y_{6X} &= -|\vec{Y}_6| \sin \theta_5 \sin \phi \cos \theta_1 + |\vec{Y}_6| \cos \theta_5 \sin \theta_1 \\ Y_{6Y} &= |\vec{Y}_6| \sin \theta_5 \sin \phi \sin \theta_1 - |\vec{Y}_6| \cos \theta_5 \cos \theta_1 \\ Y_{6Z} &= -|\vec{Y}_6| \sin \theta_5 \cos \phi \end{aligned} \quad (\text{A.9})$$

That the magnitudes of \vec{X}_6 , \vec{Y}_6 , and \vec{Z}_6 are all unity, the above equations can be simplified. Placing them all in rotation matrix form as per Eq. A.7,

the more compact form of the rotation matrix is:

$$Q = \begin{bmatrix} -\cos \theta_5 \sin \phi \cos \theta_1 + \sin \theta_5 \sin \theta_1 & -\sin \theta_5 \sin \phi \cos \theta_1 + \cos \theta_5 \sin \theta_1 & \cos \phi \cos \theta_1 \\ -\cos \theta_5 \sin \phi \sin \theta_1 - \sin \theta_5 \cos \theta_1 & \sin \theta_5 \sin \phi \sin \theta_1 - \cos \theta_5 \cos \theta_1 & \cos \phi \sin \theta_1 \\ \cos \theta_5 \cos \phi & -\sin \theta_5 \cos \phi & \sin \phi \end{bmatrix} \quad (\text{A.10})$$

As with the position vector equation, this rotation matrix was compared numerically to the more complex version found at Eq. 3.3 and found to provide the same solution for all possible joint angles.

A.2 Solution to the IKP

In finding a solution to the IKP of the Catalyst-5 robot, it again becomes useful to note that the serial arm robot is planar from joint nodes A through D on Figure 3.1. As input to the IKP solution, both the end-effector position vector (\vec{P}) and the orientation matrix of the end-effector (Q) are provided, and the intent is to find a joint angle vector $\vec{\theta}$ that allows for this configuration. Starting from \vec{P} and using elements of Q , as defined in Eq. A.7, it becomes possible and relevant to find the position vectors of \vec{D} and \vec{C} respectively. From the auxiliary view in Figure A.1 and noting that the distance between points P and D is a_5 , the position vector \vec{D} can be found from Eq. A.11.

$$\vec{D} = \vec{P} + a_5 \vec{X}_6 = \begin{bmatrix} P_X + a_5 X_{6X} \\ P_Y + a_5 X_{6Y} \\ P_Z + a_5 X_{6Z} \end{bmatrix} \quad (\text{A.11})$$

The position vector of \vec{C} can be found in a similar manner. Noting that a vector from points C to point D is parallel to \vec{Z}_6 for all values of $\vec{\theta}$ and that the distance between these points is b_5 , \vec{C} can be found from Eq. A.12.

$$\vec{C} = \vec{D} - b_5 \vec{Z}_6 = \begin{bmatrix} D_X - b_5 Z_{6X} \\ D_Y - b_5 Z_{6Y} \\ D_Z - b_5 Z_{6Z} \end{bmatrix} \quad (\text{A.12})$$

With the position of point C known, the first joint angle can be determined from Eq. A.13. In creating a solution to the IKP certain joint angle limits had to be defined to ensure an appropriate solution. Using Eq. A.13 to determine θ_1 requires that joint 2 to be limited such that $\theta_2 < \pi/2$. Values of $\theta_2 \geq \pi/2$ would result in no solution, or an incorrect solution for θ_1 . While not ideal,

this artificial limitation does not significantly limit possible solutions to the IKP as joint 2 is physically limited to $\theta_2 \leq 109^\circ$ as per Table 3.1.

$$\theta_1 = \text{atan2}(C_Y, C_X) \quad (\text{A.13})$$

With the position vectors of points C and D known, it is now possible to determine the radial position of both points. As shown in Figure A.2, the radial positions can be found from Eqs. A.14, which is undefined for values of $\theta_1 = \pm\frac{\pi}{2}$. When this is the case, the radial positions can be found from Eqs. A.15. Using this IKP solution requires an algorithm to choose the appropriate formula based on the value of θ_1 .

$$C_R = \frac{C_X}{\cos\theta_1}, \quad D_R = \frac{D_X}{\cos\theta_1} \quad (\text{A.14})$$

$$\text{for } \theta_1 = \pm\frac{\pi}{2}, \quad C_R = \frac{C_Y}{\sin\theta_1}, \quad D_R = \frac{D_Y}{\sin\theta_1} \quad (\text{A.15})$$

The value of ϕ can now be determined based on the differences of radial and vertical positions of points C and D. From Figure A.1 it can be observed that ϕ can be found using Eq. A.16.

$$\phi = \text{atan2}(D_Z - C_Z, D_R - C_R) \quad (\text{A.16})$$

With the value of ϕ known, θ_5 can now be determined using X_{6Z} and Y_{6Z} taken from Eq. A.7, the solution to the DKP equations, as follows in Eq. A.17.

$$\begin{aligned} X_{6Z} &= \cos\theta_5 \cos\phi \Rightarrow \cos\theta_5 = \frac{X_{6Z}}{\cos\phi} \\ Y_{6Z} &= -\sin\theta_5 \cos\phi \Rightarrow \sin\theta_5 = -\frac{Y_{6Z}}{\cos\phi} \\ \theta_5 &= \text{atan2}\left(-\frac{Y_{6Z}}{\cos\phi}, \frac{X_{6Z}}{\cos\phi}\right) \end{aligned} \quad (\text{A.17})$$

As Eq. A.17 is undefined for values of $\phi = \pm\frac{\pi}{2}$, an alternative solution is needed along with an algorithm to choose the correct equation based on the value of ϕ . The derivation of these alternative solutions is shown below with Eq. A.18 giving the value for θ_5 when $\phi = \frac{\pi}{2}$ and using a similar derivation Eq. A.19 provides the solution when $\phi = -\frac{\pi}{2}$.

$$\begin{aligned}
X_{6X} &= -\cos \theta_5 \sin \phi \cos \theta_1 + \sin \theta_5 \sin \theta_1 \\
&= -\cos \theta_5 \cos \theta_1 + \sin \theta_5 \sin \theta_1 \\
&= -(\cos \theta_5 \cos \theta_1 - \sin \theta_5 \sin \theta_1) \\
&= -\cos(\theta_5 + \theta_1)
\end{aligned}$$

$$\begin{aligned}
X_{6Y} &= -\cos \theta_5 \sin \phi \sin \theta_1 - \sin \theta_5 \cos \theta_1 \\
&= -(\cos \theta_5 \sin \theta_1 + \sin \theta_5 \cos \theta_1) \\
&= -\sin(\theta_5 + \theta_1)
\end{aligned}$$

$$\Rightarrow \theta_5 + \theta_1 = \text{atan2}(-X_{6Y}, -X_{6X})$$

$$\text{for } \phi = \frac{\pi}{2}, \theta_5 = \text{atan2}(-X_{6Y}, -X_{6X}) - \theta_1 \quad (\text{A.18})$$

$$\text{for } \phi = -\frac{\pi}{2}, \theta_5 = \theta_1 - \text{atan2}(X_{6Y}, X_{6X}) \quad (\text{A.19})$$

It now becomes useful to look at the radial (R) and vertical (Z) positions of point C, but this time with respect to point A. The radial position $[C_R]_A$ holds the same value of C_R as was found in Eqs. A.14 or A.15. The value of $[C_Z]_A$ can be found using Eq. A.20 where the $(D_Z - b_1)$ term accounts for point A being the joint offset b_1 higher than the origin.

$$[C_Z]_A = (D_Z - b_1) - b_5 \sin \phi \quad (\text{A.20})$$

By observation of Figure A.1, the radial and vertical positions of point $[C]_A$ can also be given by Eqs. A.21 and A.22.

$$[C_R]_A = a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3) \quad (\text{A.21})$$

$$[C_Z]_A = a_2 \sin \theta_2 + a_3 \sin(\theta_2 + \theta_3) \quad (\text{A.22})$$

Squaring both sides of the above equations and adding them together gives Eq. A.23 as follows.

$$\begin{aligned}
[C_R]_A^2 &= a_2^2 \cos^2 \theta_2 + 2a_2a_3 \cos \theta_2 \cos(\theta_2 + \theta_3) + a_3^2 \cos^2(\theta_2 + \theta_3) \\
[C_Z]_A^2 &= a_2^2 \sin^2 \theta_2 + 2a_2a_3 \sin \theta_2 \sin(\theta_2 + \theta_3) + a_3^2 \sin^2(\theta_2 + \theta_3) \\
[C_R]_A^2 + [C_Z]_A^2 &= a_2^2 + a_3^2 + 2a_2a_3(\cos \theta_2 \cos(\theta_2 + \theta_3) + \sin \theta_2 \sin(\theta_2 + \theta_3)) \\
&= a_2^2 + a_3^2 + 2a_2a_3 \cos(\theta_2 - (\theta_2 + \theta_3)) \\
&= a_2^2 + a_3^2 + 2a_2a_3 \cos \theta_3
\end{aligned} \quad (\text{A.23})$$

Equating this to the values for C_R and C_Z found above, solving for $\cos \theta_3$ and noting that $\sin \theta_3 = \pm\sqrt{1 - \cos^2 \theta_3}$, two solutions for θ_3 can be found from Eq. A.24. The algorithm used with this IKP solution will need to determine which (if any) of the values of θ_3 are within the range of motion of the joint.

$$\theta_3 = \text{atan2}(\sin \theta_3, \cos \theta_3) \quad (\text{A.24})$$

Expanding Eqs. A.21 and A.22 and introducing known values $k_1 = a_2 + a_3 \cos \theta_3$ and $k_2 = a_3 \sin \theta_3$ gives the Eqs. A.25 and A.26.

$$\begin{aligned} [C_R]_A &= a_2 \cos \theta_2 + a_3 \cos \theta_2 \cos \theta_3 - a_3 \sin \theta_2 \sin \theta_3 \\ &= \cos \theta_2 (a_2 + a_3 \cos \theta_3) - \sin \theta_2 (a_3 \sin \theta_3) \\ &= k_1 \cos \theta_2 - k_2 \sin \theta_2 \end{aligned} \quad (\text{A.25})$$

$$\begin{aligned} [C_Z]_A &= a_2 \sin \theta_2 + a_3 \sin \theta_2 \cos \theta_3 + a_3 \cos \theta_2 \sin \theta_3 \\ &= \sin \theta_2 (a_2 + a_3 \cos \theta_3) + \cos \theta_2 (a_3 \sin \theta_3) \\ &= k_1 \sin \theta_2 + k_2 \cos \theta_2 \end{aligned} \quad (\text{A.26})$$

Solving the above equations for $\sin \theta_2$ and $\cos \theta_2$, as per below, leads to a unique solution for θ_2 based on the solution for θ_3 that was determined to be within the range of motion of that joint.

$$\sin \theta_2 = \frac{k_1 [C_Z]_A - k_2 [C_R]_A}{k_1^2 + k_2^2} \quad (\text{A.27})$$

$$\cos \theta_2 = \frac{k_1 [C_R]_A + k_2 [C_Z]_A}{k_1^2 + k_2^2} \quad (\text{A.28})$$

$$\theta_2 = \text{atan2}(\sin \theta_2, \cos \theta_2) \quad (\text{A.29})$$

Finally, θ_4 can be determined by rearranging Eq. A.5 to give the following.

$$\theta_4 = \phi - \theta_2 - \theta_3 + 90^\circ \quad (\text{A.30})$$

B Maple Equations

Vectors between DH-axes origins (inches):

$$a_1 := \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix} :$$

$$a_2 := \begin{bmatrix} 10 \cdot \cos(\theta_2(t)) \\ 10 \cdot \sin(\theta_2(t)) \\ 0 \end{bmatrix} :$$

$$a_3 := \begin{bmatrix} 10 \cdot \cos(\theta_3(t)) \\ 10 \cdot \sin(\theta_3(t)) \\ 0 \end{bmatrix} :$$

$$a_4 := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} :$$

$$a_5 := \begin{bmatrix} -2.31 \cdot \cos(\theta_5(t)) \\ -2.31 \cdot \sin(\theta_5(t)) \\ 4.552 \end{bmatrix} :$$

Rotation matrices between DH-axes reference frames:

$$Q_1 := \begin{bmatrix} \cos(\theta_1(t)) & 0 & \sin(\theta_1(t)) \\ \sin(\theta_1(t)) & 0 & -\cos(\theta_1(t)) \\ 0 & 1 & 0 \end{bmatrix} :$$

$$Q_2 := \begin{bmatrix} \cos(\theta_2(t)) & -\sin(\theta_2(t)) & 0 \\ \sin(\theta_2(t)) & \cos(\theta_2(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} :$$

$$Q_3 := \begin{bmatrix} \cos(\theta_3(t)) & -\sin(\theta_3(t)) & 0 \\ \sin(\theta_3(t)) & \cos(\theta_3(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} :$$

$$Q_4 := \begin{bmatrix} \cos(\theta_4(t)) & 0 & \sin(\theta_4(t)) \\ \sin(\theta_4(t)) & 0 & -\cos(\theta_4(t)) \\ 0 & 1 & 0 \end{bmatrix} :$$

$$Q_5 := \begin{bmatrix} \cos(\theta_5(t)) & -\sin(\theta_5(t)) & 0 \\ \sin(\theta_5(t)) & \cos(\theta_5(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} :$$

Vectors from link attached DH-axis reference frame to link center of mass (CofM(i) from Ref-frm(i+1)) (inches):

$$s_1 := \begin{bmatrix} -1.5064 \\ -1.2078 \\ 6.8729e-3 \end{bmatrix} :$$

$$s_2 := \begin{bmatrix} -5.0626 \\ -1.5702e-5 \\ 5.9109007e-3 \end{bmatrix} :$$

$$s_3 := \begin{bmatrix} -6.4000 \\ -6.6428e-2 \\ -8.4311e-3 \end{bmatrix} :$$

$$s_4 := \begin{bmatrix} 6.5737e-6 \\ -3.6723e-2 \\ -3.8014e-2 \end{bmatrix} :$$

$$s_5 := \begin{bmatrix} 2.446 \\ -1.367e-5 \\ -0.562 \end{bmatrix} :$$

Joint axis vector for all joints (all revolute joints rotate about Z-axis):

$$e := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} :$$

Masses of each link of Catalyst-5 Robot (lbm):

$$m_1 := 2.7149655e1 :$$

$$m_2 := 4.114222 :$$

$$m_3 := 4.1563835 :$$

$$m_4 := 5.9499662e-1 :$$

$$m_5 := 2.474 :$$

Inertia Tensor (lbm*in^2) at center of mass of i-th link with respect to body attached DH-axis reference frame (i+1):

$$\begin{aligned}
In_1 &:= \begin{bmatrix} 3.3830e2 & -1.9286e1 & -2.6606e1 \\ -1.9286e1 & 3.1031e2 & 1.1051 \\ -2.6606e1 & 1.1051 & 2.03481e2 \end{bmatrix} : \\
In_2 &:= \begin{bmatrix} 1.2195e1 & -9.1890e-2 & 6.6511e-2 \\ -9.1890e-2 & 5.8207e1 & -2.3140e-2 \\ 6.6511e-2 & -2.3140e-2 & 5.4435e1 \end{bmatrix} : \\
In_3 &:= \begin{bmatrix} 7.1704 & -5.9040e-1 & -2.9083e-1 \\ -5.9040e-1 & 6.2564e1 & -7.0187e-3 \\ -2.9083e-1 & -7.0187e-3 & 6.0431e1 \end{bmatrix} : \\
In_4 &:= \begin{bmatrix} 3.0103e-1 & -3.8890e-6 & 0 \\ -3.8890e-6 & 2.5710e-1 & 2.9609e-3 \\ 0 & 2.9609e-3 & 2.9556e-1 \end{bmatrix} : \\
In_5 &:= \begin{bmatrix} 4.6175 & -3.238e-6 & 0.04047 \\ -3.238e-6 & 6.016 & 6.931e-5 \\ 0.04047 & 6.931e-5 & 5.8178 \end{bmatrix} :
\end{aligned}$$

with(LinearAlgebra) :

Build Jacobian matrices that map joint velocities to angular velocities (J_{ω_i}) and joint velocities to center of mass velocities (J_{c_i}) of the i -th link (each component rotated to ref frame $i+1$):

$$j_{\omega 11} := \text{Transpose}(Q_1).e :$$

$$\text{zero} := \langle 0, 0, 0 \rangle :$$

$$J_{\omega 1} := \langle j_{\omega 11} | \text{zero} | \text{zero} | \text{zero} | \text{zero} \rangle :$$

$$r_{11} := \text{Transpose}(Q_1).a_1 + s_1 :$$

$$j_{c 11} := \text{CrossProduct}(j_{\omega 11}, r_{11}) :$$

$$J_{c 1} := \langle j_{c 11} | \text{zero} | \text{zero} | \text{zero} | \text{zero} \rangle :$$

$$j_{\omega 21} := \text{Transpose}(Q_1 \cdot Q_2).e :$$

$$j_{\omega 22} := \text{Transpose}(Q_2).e :$$

$$J_{\omega 2} := \langle j_{\omega 21} | j_{\omega 22} | \text{zero} | \text{zero} | \text{zero} \rangle :$$

$$r_{21} := \text{Transpose}(Q_1 \cdot Q_2).a_1 + \text{Transpose}(Q_2).a_2 + s_2 :$$

$$r_{22} := \text{Transpose}(Q_2).a_2 + s_2 :$$

$$j_{c 21} := \text{CrossProduct}(j_{\omega 21}, r_{21}) :$$

$$j_{c 22} := \text{CrossProduct}(j_{\omega 22}, r_{22}) :$$

$$J_{c 2} := \langle j_{c 21} | j_{c 22} | \text{zero} | \text{zero} | \text{zero} \rangle :$$

$$\begin{aligned}
j_{\omega 31} &:= \text{Transpose}(Q_1 \cdot Q_2 \cdot Q_3) \cdot e : \\
j_{\omega 32} &:= \text{Transpose}(Q_2 \cdot Q_3) \cdot e : \\
j_{\omega 33} &:= \text{Transpose}(Q_3) \cdot e : \\
J_{\omega 3} &:= \langle j_{\omega 31} | j_{\omega 32} | j_{\omega 33} | \text{zero} | \text{zero} \rangle :
\end{aligned}$$

$$\begin{aligned}
r_{31} &:= \text{Transpose}(Q_1 \cdot Q_2 \cdot Q_3) \cdot a_1 + \text{Transpose}(Q_2 \cdot Q_3) \cdot a_2 + \text{Transpose}(Q_3) \cdot a_3 + s_3 : \\
r_{32} &:= \text{Transpose}(Q_2 \cdot Q_3) \cdot a_2 + \text{Transpose}(Q_3) \cdot a_3 + s_3 : \\
r_{33} &:= \text{Transpose}(Q_3) \cdot a_3 + s_3 : \\
j_{c31} &:= \text{CrossProduct}(j_{\omega 31}, r_{31}) : \\
j_{c32} &:= \text{CrossProduct}(j_{\omega 32}, r_{32}) : \\
j_{c33} &:= \text{CrossProduct}(j_{\omega 33}, r_{33}) : \\
J_{c3} &:= \langle j_{c31} | j_{c32} | j_{c33} | \text{zero} | \text{zero} \rangle :
\end{aligned}$$

$$\begin{aligned}
j_{\omega 41} &:= \text{Transpose}(Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4) \cdot e : \\
j_{\omega 42} &:= \text{Transpose}(Q_2 \cdot Q_3 \cdot Q_4) \cdot e : \\
j_{\omega 43} &:= \text{Transpose}(Q_3 \cdot Q_4) \cdot e : \\
j_{\omega 44} &:= \text{Transpose}(Q_4) \cdot e : \\
J_{\omega 4} &:= \langle j_{\omega 41} | j_{\omega 42} | j_{\omega 43} | j_{\omega 44} | \text{zero} \rangle :
\end{aligned}$$

$$\begin{aligned}
r_{41} &:= \text{Transpose}(Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4) \cdot a_1 + \text{Transpose}(Q_2 \cdot Q_3 \cdot Q_4) \cdot a_2 + \text{Transpose}(Q_3 \cdot Q_4) \cdot a_3 \\
&\quad + \text{Transpose}(Q_4) \cdot a_4 + s_4 : \\
r_{42} &:= \text{Transpose}(Q_2 \cdot Q_3 \cdot Q_4) \cdot a_2 + \text{Transpose}(Q_3 \cdot Q_4) \cdot a_3 + \text{Transpose}(Q_4) \cdot a_4 + s_4 : \\
r_{43} &:= \text{Transpose}(Q_3 \cdot Q_4) \cdot a_3 + \text{Transpose}(Q_4) \cdot a_4 + s_4 : \\
r_{44} &:= \text{Transpose}(Q_4) \cdot a_4 + s_4 : \\
j_{c41} &:= \text{CrossProduct}(j_{\omega 41}, r_{41}) : \\
j_{c42} &:= \text{CrossProduct}(j_{\omega 42}, r_{42}) : \\
j_{c43} &:= \text{CrossProduct}(j_{\omega 43}, r_{43}) : \\
j_{c44} &:= \text{CrossProduct}(j_{\omega 44}, r_{44}) : \\
J_{c4} &:= \langle j_{c41} | j_{c42} | j_{c43} | j_{c44} | \text{zero} \rangle :
\end{aligned}$$

$$\begin{aligned}
j_{\omega 51} &:= \text{Transpose}(Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4 \cdot Q_5) \cdot e : \\
j_{\omega 52} &:= \text{Transpose}(Q_2 \cdot Q_3 \cdot Q_4 \cdot Q_5) \cdot e : \\
j_{\omega 53} &:= \text{Transpose}(Q_3 \cdot Q_4 \cdot Q_5) \cdot e : \\
j_{\omega 54} &:= \text{Transpose}(Q_4 \cdot Q_5) \cdot e : \\
j_{\omega 55} &:= \text{Transpose}(Q_5) \cdot e : \\
J_{\omega 5} &:= \langle j_{\omega 51} | j_{\omega 52} | j_{\omega 53} | j_{\omega 54} | j_{\omega 55} \rangle :
\end{aligned}$$

$$\begin{aligned}
r_{51} &:= \text{Transpose}(Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4 \cdot Q_5) \cdot a_1 + \text{Transpose}(Q_2 \cdot Q_3 \cdot Q_4 \cdot Q_5) \cdot a_2 + \text{Transpose}(Q_3 \cdot Q_4 \cdot Q_5) \cdot a_3 \\
&\quad + \text{Transpose}(Q_4 \cdot Q_5) \cdot a_4 + \text{Transpose}(Q_5) \cdot a_5 + s_5 :
\end{aligned}$$

$$\begin{aligned}
r_{52} &:= \text{Transpose}(Q_2 \cdot Q_3 \cdot Q_4 \cdot Q_5) \cdot a_2 + \text{Transpose}(Q_3 \cdot Q_4 \cdot Q_5) \cdot a_3 + \text{Transpose}(Q_4 \cdot Q_5) \cdot a_4 \\
&\quad + \text{Transpose}(Q_5) \cdot a_5 + s_5 : \\
r_{53} &:= \text{Transpose}(Q_3 \cdot Q_4 \cdot Q_5) \cdot a_3 + \text{Transpose}(Q_4 \cdot Q_5) \cdot a_4 + \text{Transpose}(Q_5) \cdot a_5 + s_5 : \\
r_{54} &:= \text{Transpose}(Q_4 \cdot Q_5) \cdot a_4 + \text{Transpose}(Q_5) \cdot a_5 + s_5 : \\
r_{55} &:= \text{Transpose}(Q_5) \cdot a_5 + s_5 : \\
j_{c51} &:= \text{CrossProduct}(j_{\omega 51}, r_{51}) : \\
j_{c52} &:= \text{CrossProduct}(j_{\omega 52}, r_{52}) : \\
j_{c53} &:= \text{CrossProduct}(j_{\omega 53}, r_{53}) : \\
j_{c54} &:= \text{CrossProduct}(j_{\omega 54}, r_{54}) : \\
j_{c55} &:= \text{CrossProduct}(j_{\omega 55}, r_{55}) : \\
J_{c5} &:= \langle j_{c51} | j_{c52} | j_{c53} | j_{c54} | j_{c55} \rangle :
\end{aligned}$$

Generalized Inertia Tensor of the Catalyst-5 Robot:

$$\begin{aligned}
M &:= m_1 \cdot \text{Transpose}(J_{c1}) \cdot J_{c1} + \text{Transpose}(J_{\omega 1}) \cdot \text{In}_1 \cdot J_{\omega 1} + m_2 \cdot \text{Transpose}(J_{c2}) \cdot J_{c2} + \text{Transpose}(J_{\omega 2}) \cdot \text{In}_2 \\
&\quad \cdot J_{\omega 2} + m_3 \cdot \text{Transpose}(J_{c3}) \cdot J_{c3} + \text{Transpose}(J_{\omega 3}) \cdot \text{In}_3 \cdot J_{\omega 3} + m_4 \cdot \text{Transpose}(J_{c4}) \cdot J_{c4} \\
&\quad + \text{Transpose}(J_{\omega 4}) \cdot \text{In}_4 \cdot J_{\omega 4} + m_5 \cdot \text{Transpose}(J_{c5}) \cdot J_{c5} + \text{Transpose}(J_{\omega 5}) \cdot \text{In}_5 \cdot J_{\omega 5} :
\end{aligned}$$

Simplifying M and dMdt trigonometrically to reduce the size of the 5x5 matrix equations:

$$\text{simpM} := \text{simplify}(M, \text{trig}) :$$

Removing the time dependant function of theta and swaping theta dot for the omega symbol to be able to convert the Maple generated equations to MATLAB code:

$$NM := \text{subs}(\theta_1(t) = \theta_1, \theta_2(t) = \theta_2, \theta_3(t) = \theta_3, \theta_4(t) = \theta_4, \theta_5(t) = \theta_5, \text{simpM}) :$$

Attempt to take the algebraic inverse of the generalized inertia tensor. Command does not solve:

$$N\text{Minv} := \text{MatrixInverse}(NM) :$$

Warning, computation interrupted

Partial differential of M by each element of theta for use in the Coriolis term of the dynamic equation:

$$\delta M \delta \theta_1 := \text{map}(\text{diff}, NM, \theta_1) :$$

$$\text{simp} \delta M \delta \theta_1 := \text{simplify}(\delta M \delta \theta_1, \text{trig}) :$$

$$\delta M \delta \theta_2 := \text{map}(\text{diff}, NM, \theta_2) :$$

$$\text{simp} \delta M \delta \theta_2 := \text{simplify}(\delta M \delta \theta_2, \text{trig}) :$$

$$\delta M \delta \theta_3 := \text{map}(\text{diff}, NM, \theta_3) :$$

$$\text{simp} \delta M \delta \theta_3 := \text{simplify}(\delta M \delta \theta_3, \text{trig}) :$$

$$\delta M \delta \theta_4 := \text{map}(\text{diff}, NM, \theta_4) :$$

$$\text{simp}\delta M\delta\theta_4 := \text{simplify}(\delta M\delta\theta_4, \text{trig}) :$$

$$\delta M\delta\theta_5 := \text{map}(\text{diff}, NM, \theta_5) :$$

$$\text{simp}\delta M\delta\theta_5 := \text{simplify}(\delta M\delta\theta_5, \text{trig}) :$$

Removing the time dependant function of theta to be able to convert the Maple generated equations to MATLAB code:

$$N\delta M\delta\theta_1 := \text{subs}(\theta_1(t) = \theta_1, \theta_2(t) = \theta_2, \theta_3(t) = \theta_3, \theta_4(t) = \theta_4, \theta_5(t) = \theta_5, \text{simp}\delta M\delta\theta_1) :$$

$$N\delta M\delta\theta_2 := \text{subs}(\theta_1(t) = \theta_1, \theta_2(t) = \theta_2, \theta_3(t) = \theta_3, \theta_4(t) = \theta_4, \theta_5(t) = \theta_5, \text{simp}\delta M\delta\theta_2) :$$

$$N\delta M\delta\theta_3 := \text{subs}(\theta_1(t) = \theta_1, \theta_2(t) = \theta_2, \theta_3(t) = \theta_3, \theta_4(t) = \theta_4, \theta_5(t) = \theta_5, \text{simp}\delta M\delta\theta_3) :$$

$$N\delta M\delta\theta_4 := \text{subs}(\theta_1(t) = \theta_1, \theta_2(t) = \theta_2, \theta_3(t) = \theta_3, \theta_4(t) = \theta_4, \theta_5(t) = \theta_5, \text{simp}\delta M\delta\theta_4) :$$

$$N\delta M\delta\theta_5 := \text{subs}(\theta_1(t) = \theta_1, \theta_2(t) = \theta_2, \theta_3(t) = \theta_3, \theta_4(t) = \theta_4, \theta_5(t) = \theta_5, \text{simp}\delta M\delta\theta_5) :$$

Determine the potential energy component of the dynamic model:

$$g := 32.174 \cdot 12 :$$

$$eg := \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} :$$

$$c_1 := a_1 + Q_1 \cdot s_1 :$$

$$c_2 := a_1 + Q_1 \cdot a_2 + Q_1 \cdot Q_2 \cdot s_2 :$$

$$c_3 := a_1 + Q_1 \cdot a_2 + Q_1 \cdot Q_2 \cdot a_3 + Q_1 \cdot Q_2 \cdot Q_3 \cdot s_3 :$$

$$c_4 := a_1 + Q_1 \cdot a_2 + Q_1 \cdot Q_2 \cdot a_3 + Q_1 \cdot Q_2 \cdot Q_3 \cdot a_4 + Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4 \cdot s_4 :$$

$$c_5 := a_1 + Q_1 \cdot a_2 + Q_1 \cdot Q_2 \cdot a_3 + Q_1 \cdot Q_2 \cdot Q_3 \cdot a_4 + Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4 \cdot a_5 + Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4 \cdot Q_5 \cdot s_5 :$$

$$V := -g \cdot \text{Transpose}(eg) \cdot (m_1 \cdot c_1 + m_2 \cdot c_2 + m_3 \cdot c_3 + m_4 \cdot c_4 + m_5 \cdot c_5) :$$

Removing the time dependant function of theta to be able to convert the Maple generated equations to MATLAB code.

$$NV := \text{subs}(\theta_1(t) = \theta_1, \theta_2(t) = \theta_2, \theta_3(t) = \theta_3, \theta_4(t) = \theta_4, \theta_5(t) = \theta_5, V) :$$

Partial differential of V by each element of theta to get the vector of forces due to gravity for use in the dynamic equation:

$$\delta V\delta\theta_1 := \text{map}(\text{diff}, NV, \theta_1) :$$

$$\delta V\delta\theta_2 := \text{map}(\text{diff}, NV, \theta_2) :$$

$$\delta V\delta\theta_3 := \text{map}(\text{diff}, NV, \theta_3) :$$

$$\delta V\delta\theta_4 := \text{map}(\text{diff}, NV, \theta_4) :$$

$$\delta V\delta\theta_5 := \text{map}(\text{diff}, NV, \theta_5) :$$

$$\text{simp}\delta V\delta\theta_1 := \text{simplify}(\delta V\delta\theta_1, \text{trig}) :$$

$$\text{simp}\delta V\delta\theta_2 := \text{simplify}(\delta V\delta\theta_2, \text{trig}) :$$

$$\text{simp}\delta V\delta\theta_3 := \text{simplify}(\delta V\delta\theta_3, \text{trig}) :$$

$$\text{simp}\delta V\delta\theta_4 := \text{simplify}(\delta V\delta\theta_4, \text{trig}) :$$

$\text{simp}\delta V\delta\theta_5 := \text{simplify}(\delta V\delta\theta_5, \text{trig}) :$

Determining the Jacobian of the Catalyst-5 to be able to convert the end-effector wrench to joint torques:

$E_1 := e :$

$E_2 := Q_1 \cdot e :$

$E_3 := Q_1 \cdot Q_2 \cdot e :$

$E_4 := Q_1 \cdot Q_2 \cdot Q_3 \cdot e :$

$E_5 := Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4 \cdot e :$

$r_1 := a_1 + Q_1 \cdot a_2 + Q_1 \cdot Q_2 \cdot a_3 + Q_1 \cdot Q_2 \cdot Q_3 \cdot a_4 + Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4 \cdot a_5 :$

$r_2 := Q_1 \cdot a_2 + Q_1 \cdot Q_2 \cdot a_3 + Q_1 \cdot Q_2 \cdot Q_3 \cdot a_4 + Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4 \cdot a_5 :$

$r_3 := Q_1 \cdot Q_2 \cdot a_3 + Q_1 \cdot Q_2 \cdot Q_3 \cdot a_4 + Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4 \cdot a_5 :$

$r_4 := Q_1 \cdot Q_2 \cdot Q_3 \cdot a_4 + Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4 \cdot a_5 :$

$r_5 := Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4 \cdot a_5 :$

$J := \langle \langle E_1|E_2|E_3|E_4|E_5 \rangle, \langle \text{CrossProduct}(E_1, r_1)|\text{CrossProduct}(E_2, r_2)|\text{CrossProduct}(E_3, r_3) | \text{CrossProduct}(E_4, r_4)|\text{CrossProduct}(E_5, r_5) \rangle \rangle :$

$Jdot := \text{map}(\text{diff}, J, t) :$

$NJ := \text{subs}(\theta_1(t) = \theta_1, \theta_2(t) = \theta_2, \theta_3(t) = \theta_3, \theta_4(t) = \theta_4, \theta_5(t) = \theta_5, J) :$

$NJdot := \text{subs}\left(\left(\frac{d}{dt} \theta_1(t)\right) = \omega_1, \left(\frac{d}{dt} \theta_2(t)\right) = \omega_2, \left(\frac{d}{dt} \theta_3(t)\right) = \omega_3, \left(\frac{d}{dt} \theta_4(t) = \omega_4\right), \left(\frac{d}{dt} \theta_5(t) = \omega_5\right), Jdot\right) :$

$NNJdot := \text{subs}(\theta_1(t) = \theta_1, \theta_2(t) = \theta_2, \theta_3(t) = \theta_3, \theta_4(t) = \theta_4, \theta_5(t) = \theta_5, NJdot) :$

$NJsimp := \text{simplify}(NNJdot, \text{trig}) :$

$NJtrans := \text{Transpose}(NJsimp) :$

Divide Generalized Inertia Tensor into separate rows to ensure proper conversion to MATLAB code using codegeneration below:

$NM1 := NM[1] :$

$NM2 := NM[2] :$

$NM3 := NM[3] :$

$NM4 := NM[4] :$

$NM5 := NM[5] :$

Code Generation application that creates MATLAB code from a Maple equation. Used for all elements of the dynamic model calculated above:

$\text{with}(\text{CodeGeneration}) :$

$\text{Matlab}(\text{simp}\delta V\delta\theta_5, \text{resultname} = \text{"deltaV5"}) :$

C Manufacturer's Data

Coordinate frames are oriented such that the z-axis is parallel with the axis of revolution and the x-axis is aligned with the length of the link

The following inertias do not include drive components. Drive component inertias are listed at the end of this document as seen from the output

BASE

VOLUME = 9.5893005e+01 INCH³
SURFACE AREA = 9.0794847e+02 INCH²
AVERAGE DENSITY = 9.7968917e-02 POUND / INCH³
MASS = 9.3945338e+00 POUND

CENTER OF GRAVITY with respect to CSBASE coordinate frame:

X Y Z -2.1263862e-01 4.5663634e-03 3.1557132e+00 INCH

INERTIA with respect to CSBASE coordinate frame: (POUND * INCH²)

INERTIA TENSOR:

Ixx	Ixy	Ixz	1.7616906e+02	1.7190013e-01	3.4739095e+00
Iyx	Iyy	Iyz	1.7190013e-01	1.8247117e+02	-1.0766366e-01
Izx	Izy	Izz	3.4739095e+00	-1.0766366e-01	6.2196999e+01

INERTIA at CENTER OF GRAVITY with respect to CSBASE coordinate frame: (POUND * INCH²)

INERTIA TENSOR:

Ixx	Ixy	Ixz	8.2613152e+01	1.6277818e-01	-2.8300718e+00
Iyx	Iyy	Iyz	1.6277818e-01	8.8490688e+01	2.7712820e-02
Izx	Izy	Izz	-2.8300718e+00	2.7712820e-02	6.1772027e+01

PRINCIPAL MOMENTS OF INERTIA: (POUND * INCH²)

I1 I2 I3 6.1394472e+01 8.2986102e+01 8.8495293e+01

ROTATION MATRIX from CSBASE orientation to PRINCIPAL AXES:

0.13222	0.99081	0.02862
-0.00181	-0.02863	0.99959
0.99122	-0.13222	-0.00199

ROTATION ANGLES from CSBASE orientation to PRINCIPAL AXES (degrees):

angles about x y z -90.114 1.640 -82.399

RADII OF GYRATION with respect to PRINCIPAL AXES:

R1 R2 R3 2.5563895e+00 2.9721113e+00 3.0691808e+00 INCH

SHOULDER

VOLUME = 2.0943701e+02 INCH³

SURFACE AREA = 2.5405291e+03 INCH²

AVERAGE DENSITY = 1.2963160e-01 POUND / INCH³

MASS = 2.7149655e+01 POUND

CENTER OF GRAVITY with respect to CS1 coordinate frame:

X Y Z -1.5063765e+00 -6.8729131e-03 8.7921828e+00 INCH

INERTIA with respect to CS1 coordinate frame: (POUND * INCH²)

INERTIA TENSOR:

Ixx	Ixy	Ixz	2.4090491e+03	8.2397430e-01	3.7886551e+02
Iyx	Iyy	Iyz	8.2397430e-01	2.3638238e+03	2.8247046e+01
Izx	Izy	Izz	3.7886551e+02	2.8247046e+01	3.9990782e+02

INERTIA at CENTER OF GRAVITY with respect to CS1 coordinate frame: (POUND * INCH²)

INERTIA TENSOR:

Ixx	Ixy	Ixz	3.1031222e+02	1.1050600e+00	1.9286308e+01
Iyx	Iyy	Iyz	1.1050600e+00	2.0348103e+02	2.6606449e+01
Izx	Izy	Izz	1.9286308e+01	2.6606449e+01	3.3829935e+02

PRINCIPAL MOMENTS OF INERTIA: (POUND * INCH²)

I1 I2 I3 1.9836226e+02 3.0159109e+02 3.5213926e+02

ROTATION MATRIX from CS1 orientation to PRINCIPAL AXES:

0.02301		-0.90877		0.41667
0.98155		0.09966		0.16316
-0.18980		0.40523		0.89430

ROTATION ANGLES from CS1 orientation to PRINCIPAL AXES (degrees):

angles about x y z -10.339 24.624 88.550

RADII OF GYRATION with respect to PRINCIPAL AXES:

R1 R2 R3 2.7030082e+00 3.3329369e+00 3.6014307e+00 INCH

LINK 1

VOLUME = 3.7471313e+01 INCH³

SURFACE AREA = 5.8635237e+02 INCH²

AVERAGE DENSITY = 1.0979658e-01 POUND / INCH³

MASS = 4.1142220e+00 POUND

CENTER OF GRAVITY with respect to CS2 coordinate frame:

X Y Z 4.9374283e+00 -1.5701652e-05 5.9109007e-03 INCH

INERTIA with respect to CS2 coordinate frame: (POUND * INCH²)

INERTIA TENSOR:

Ixx	Ixy	Ixz	1.2195258e+01	-9.1570892e-02	-5.3560776e-02
Iyx	Iyy	Iyz	-9.1570892e-02	1.5850441e+02	-2.3198358e-02
Izx	Izy	Izz	-5.3560776e-02	-2.3198358e-02	1.5473273e+02

INERTIA at CENTER OF GRAVITY with respect to CS2 coordinate frame: (POUND * INCH²)

INERTIA TENSOR:

Ixx	Ixy	Ixz	1.2195114e+01	-9.1889850e-02	6.6511346e-02
Iyx	Iyy	Iyz	-9.1889850e-02	5.8206944e+01	-2.3198740e-02
Izx	Izy	Izz	6.6511346e-02	-2.3198740e-02	5.4435408e+01

PRINCIPAL MOMENTS OF INERTIA: (POUND * INCH²)

I1 I2 I3 1.2194826e+01 5.4435368e+01 5.8207272e+01

ROTATION MATRIX from CS2 orientation to PRINCIPAL AXES:

1.00000		0.00156	0.00201
0.00200		0.00619	-0.99998
-0.00157		0.99998	0.00619

ROTATION ANGLES from CS2 orientation to PRINCIPAL AXES (degrees):

angles about x y z 89.646 0.115 -0.089

RADII OF GYRATION with respect to PRINCIPAL AXES:

R1 R2 R3 1.7216463e+00 3.6374474e+00 3.7613588e+00 INCH

LINK 2

VOLUME = 2.8248549e+01 INCH³

SURFACE AREA = 6.3666666e+02 INCH²

AVERAGE DENSITY = 1.4713618e-01 POUND / INCH³

MASS = 4.1563835e+00 POUND

CENTER OF GRAVITY with respect to CS3 coordinate frame:

X Y Z 3.5999759e+00 -6.6427899e-02 -8.4311457e-03 INCH

INERTIA with respect to CS3 coordinate frame: (POUND * INCH²)

INERTIA TENSOR:

Ixx	Ixy	Ixz	7.1890688e+00	4.0355527e-01	-1.6467575e-01
Iyx	Iyy	Iyz	4.0355527e-01	1.1643038e+02	-9.3465688e-03
Izx	Izy	Izz	-1.6467575e-01	-9.3465688e-03	1.1431500e+02

INERTIA at CENTER OF GRAVITY with respect to CS3 coordinate frame: (POUND * INCH²)

INERTIA TENSOR:

Ixx	Ixy	Ixz	7.1704327e+00	-5.9039743e-01	-2.9082998e-01
Iyx	Iyy	Iyz	-5.9039743e-01	6.2564072e+01	-7.0187310e-03
Izx	Izy	Izz	-2.9082998e-01	-7.0187310e-03	6.0430655e+01

PRINCIPAL MOMENTS OF INERTIA: (POUND * INCH²)

I1 I2 I3 7.1625523e+00 6.0432236e+01 6.2570371e+01

ROTATION MATRIX from CS3 orientation to PRINCIPAL AXES:

0.99993		-0.00548	0.01065
0.01066		0.00177	-0.99994
0.00546		0.99998	0.00183

ROTATION ANGLES from CS3 orientation to PRINCIPAL AXES (degrees):

angles about x y z 89.895 0.610 0.314

RADII OF GYRATION with respect to PRINCIPAL AXES:

R1 R2 R3 1.3127321e+00 3.8130853e+00 3.8799538e+00 INCH

WRIST

VOLUME = 5.1576792e+00 INCH³

SURFACE AREA = 7.6463844e+01 INCH²

AVERAGE DENSITY = 1.1536131e-01 POUND / INCH³

MASS = 5.9499662e-01 POUND

CENTER OF GRAVITY with respect to CS4 coordinate frame:

X Y Z -3.8013574e-02 6.5736601e-06 -3.6722965e-02 INCH

INERTIA with respect to CS4 coordinate frame: (POUND * INCH²)

INERTIA TENSOR:

Ixx	Ixy	Ixz	2.9635896e-01	0.0000000e+00	2.1303130e-03
Iyx	Iyy	Iyz	0.0000000e+00	3.0268721e-01	-3.7452891e-06
Izx	Izy	Izz	2.1303130e-03	-3.7452891e-06	2.5795584e-01

INERTIA at CENTER OF GRAVITY with respect to CS4 coordinate frame: (POUND * INCH²)

INERTIA TENSOR:

Ixx	Ixy	Ixz	2.9555657e-01	0.0000000e+00	2.9609111e-03
Iyx	Iyy	Iyz	0.0000000e+00	3.0102502e-01	-3.8889239e-06
Izx	Izy	Izz	2.9609111e-03	-3.8889239e-06	2.5709605e-01

PRINCIPAL MOMENTS OF INERTIA: (POUND * INCH²)

I1 I2 I3 2.5686944e-01 2.9578318e-01 3.0102502e-01

ROTATION MATRIX from CS4 orientation to PRINCIPAL AXES:

-0.07631	0.99708	0.00024
0.00009	-0.00023	1.00000
0.99708	0.07631	-0.00007

ROTATION ANGLES from CS4 orientation to PRINCIPAL AXES (degrees):

angles about x y z -90.004 0.000 -94.377

RADII OF GYRATION with respect to PRINCIPAL AXES:

R1 R2 R3 6.5705083e-01 7.0506553e-01 7.1128565e-01 INCH

TOOL FLANGE

VOLUME = 1.2562147e+00 INCH³

SURFACE AREA = 2.0552608e+01 INCH²

AVERAGE DENSITY = 3.1624880e-02 POUND / INCH³

MASS = 3.9727640e-02 POUND

CENTER OF GRAVITY with respect to CS5 coordinate frame:

X Y Z -8.5513705e-04 -8.5134552e-04 1.4717557e+00 INCH

INERTIA with respect to CS5 coordinate frame: (POUND * INCH²)

INERTIA TENSOR:

Ixx	Ixy	Ixz	9.7081263e-02	-8.1642389e-06	3.4153307e-05
Iyx	Iyy	Iyz	-8.1642389e-06	9.6982759e-02	3.4001655e-05
Izx	Izy	Izz	3.4153307e-05	3.4001655e-05	6.0945855e-03

INERTIA at CENTER OF GRAVITY with respect to CS5 coordinate frame: (POUND * INCH²)

INERTIA TENSOR:

Ixx	Ixy	Ixz	1.1028587e-02	-8.1353165e-06	-1.5846028e-05
Iyx	Iyy	Iyz	-8.1353165e-06	1.0930083e-02	-1.5775991e-05
Izx	Izy	Izz	-1.5846028e-05	-1.5775991e-05	6.0945277e-03

PRINCIPAL MOMENTS OF INERTIA: (POUND * INCH²)

I1 I2 I3 6.0944251e-03 1.0929475e-02 1.1029297e-02

ROTATION MATRIX from CS5 orientation to PRINCIPAL AXES:

0.00322		0.08125	-0.99669
0.00327		0.99669	0.08126
0.99999		-0.00352	0.00294

ROTATION ANGLES from CS5 orientation to PRINCIPAL AXES (degrees):

angles about x y z -87.927 -85.336 -87.733

RADII OF GYRATION with respect to PRINCIPAL AXES:

R1 R2 R3 3.9166971e-01 5.2450939e-01 5.2689919e-01 INCH

Drive Components (Output Side)

Motor Inertia = 278.1 (POUND * INCH²)

Harmonic Drive Inertia = 57.0 (POUND * INCH²)

Shoulder, Link 1, Link 2: Harmonic Drive and Motor = 335.1 (POUND * INCH²)

Wrist: Gear Train Inertia and Motor = 26.4 (POUND * INCH²)

Tool Flange: Gear Train Inertia and Motor = 6.6 (POUND * INCH²)

D Circle and Spiral Trajectory Generators

The circle trajectory is generated as two semi-circles using a modified Cartesian space trajectory generator as described in Section 4.1.3. Starting with the straight line generator the quintic polynomial that generates the trajectory along the Y-axis is replaced with Simulink blocks generating the Y-components (position and velocity) as function of X using Eqs. D.1 and D.2. The functions are derived from the formula for a circle where (a, b) are the X and Y components of the center of the circle and r is the radius. As Eq. D.2 results in division by 0 when $y = b$, a Simulink switch was added that sets velocity of Y equal to 0 as this only occurs at the beginning and end of each semi-circle trajectory where the velocity in all directions is zero.

$$(x - a)^2 + (y - b)^2 = r^2$$
$$y = \pm\sqrt{r^2 - (x - a)^2} + b \quad (\text{D.1})$$

$$\dot{y} = \frac{\dot{x}(x - a)}{y - b} \quad (\text{D.2})$$

The circle trajectory was generated by moving from the X position closest to the origin in the positive direction and taking the positive position (negative velocity) values of Y, with one SemiCircle Cartesian planner block, then moving in the negative X direction and taking negative position (positive velocity) values of Y with a second planner block. The same planner block is used for each half of the semi-circle as it contains a switch that selects the appropriate equation for y and \dot{y} based on the direction the end-effector is traveling along the x-axis.

The spiral trajectory generator is similar to the circle generator, but uses Eqs. D.3 and D.4 where σ determines the rate that the radius of the spiral changes.

$$y(x) = \pm \sqrt{(r - \sigma)^2 - (x - a)^2} + b \quad (\text{D.3})$$

$$\dot{y} = \frac{\dot{\sigma}(r - \sigma) - (x - a)}{y - b} \quad (\text{D.4})$$

For this generator, σ is defined as a function of X as shown in Eq. D.5. Due to the limited workspace, the radius reduces evenly by 25.4mm[1"] as it moves from the smallest value of X to the largest value of X, taking positive Y values. Moving in the opposite direction an additional 25.4mm[1"] evenly is removed from the radius, with the start and end points modified accordingly and the negative values of Y used.

$$\sigma(x) = 25.4\text{mm} \left(\frac{|x - X_{start}|}{|X_{end} - X_{start}|} \right) \quad (\text{D.5})$$

As Eq. D.4 requires the time derivative of σ , it is found using the following equation.

$$\dot{\sigma} = \frac{\dot{x}(x - X_{start})}{|x - X_{start}| |X_{end} - X_{start}|}$$

E Joint Space Controllers - Gains Tuning

The gains for joint 1 were tuned by having moving the robot from $\theta_1 = -\pi$ to $\theta_1 = \pi$ with the robot arm extended in the Calibration Ready (Calrdy) position as described in [60] and shown in Figure E.1.

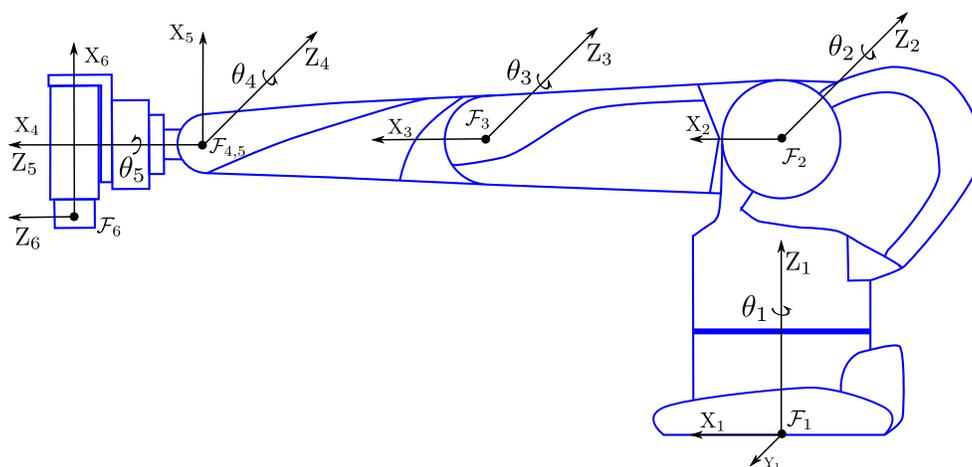


Figure E.1: Calibration Ready (Calrdy) Position

With the initial gains selected for joint 1, as found in Table 4.1, and an initial maximum joint velocity of $125^\circ/s$ the PD independent controller was able to follow the desired trajectory with a maximum error of 0.01083rad, overshoot of 0.002902rad and applied torque of 11.46Nm. As the applied torque was significantly less than the maximum, it was determined that the maximum velocity could be raised in the trajectory generator and still have a minimum error and overshoot. Through a number of simulations, it was decided that the maximum velocity for joint 1 using the PD independent controller would be $160^\circ/s$. Using these same simulations, the proportional

and derivative gains for joint 1 were optimized at 230 and 50 respectively. Using these values, the controller was able to follow the desired trajectory with a maximum error of 0.01242rad, overshoot of 0.00072rad and applied torque of 14.96Nm, as shown in Figure E.2.

Figure E.2 also shows what is a notable deficiency with the PD independent controller, as designed, in that joints 2 through 4 have a notable steady state error and are unable to hold the arm fully extended in the Caldry position. Also of interest is the error that is induced on joint 5 when the movement of joint 1 is initiated. It is suspected that this is due to the center of mass of the end-effector assembly being located away from the joint 5 axis and the large amount of momentum of this assembly causing the joint to move away from 0 degrees. This error will likely not be an issue when the controller is implemented on the actual robot due to joint friction.

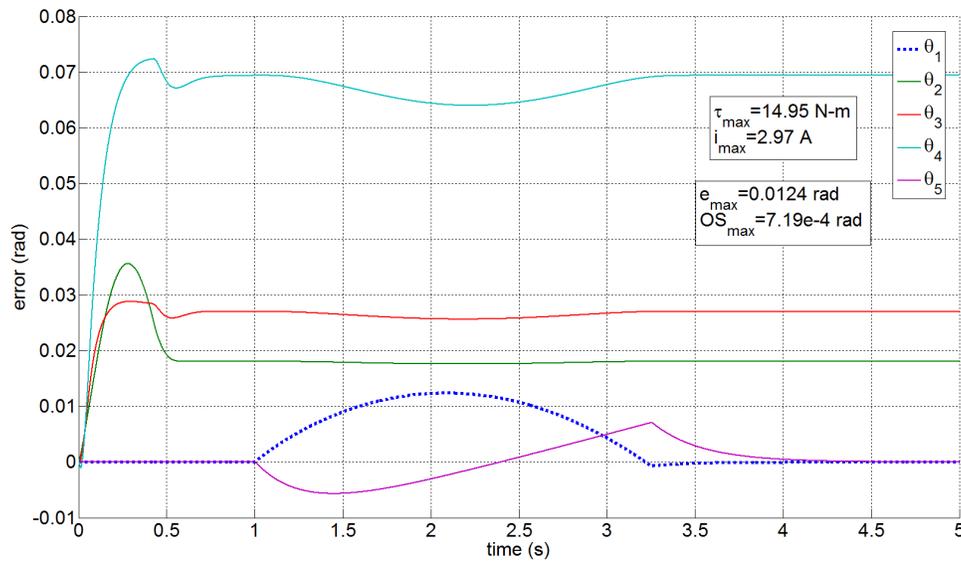


Figure E.2: PD Ind. Control - Joint Errors for Joint 1 Step ($-\pi$ to π)

Using the initial gains selected for joint 1 of $k_p = 400$ and $k_d = 40$, and a maximum joint speed of $160^\circ/s$ the joint space linearized controller was able to follow the desired trajectory with maximum error of 0.1343rad, but with an applied torque of 17.65Nm which shows the joint would be actuated at saturation current. Through a number of simulations, it was decided that the maximum velocity for joint 1 using the PD linearized controller would be $135^\circ/s$ and the gains would be adjusted to $k_p = 1600$ and $k_d = 80$. Using these

values, the controller was able to follow the desired trajectory with a maximum error of 0.01063rad, overshoot of 0rad and applied torque of 16.04Nm, as shown in Figure E.3. As the maximum applied torque is slightly above the 90% maximum identified in Table 4.2, the control currents for this joint will have to be monitored closely when this controller is first used with the robot.

As can be observed in Figure E.3, the linearized controller does not suffer from the same steady state error deficiency for joints 2 through 4 that was observed with the PD independent controller. The notable initial errors for the controller can be attributed to the low pass filter that is used as part of the differentiation of the joint position signal to provide the joint velocities. As these initial errors are due to the controller experiencing an initial step input as it accounts for the sudden mass of the links, increasing the frequency of the low pass filter would result in a reduction of these errors. Too high of a cut-off frequency results in a noisy control signal which is undesirable for the robot motors, so a cut-off frequency of 50Hz was determined experimentally and the initial errors were deemed acceptable.

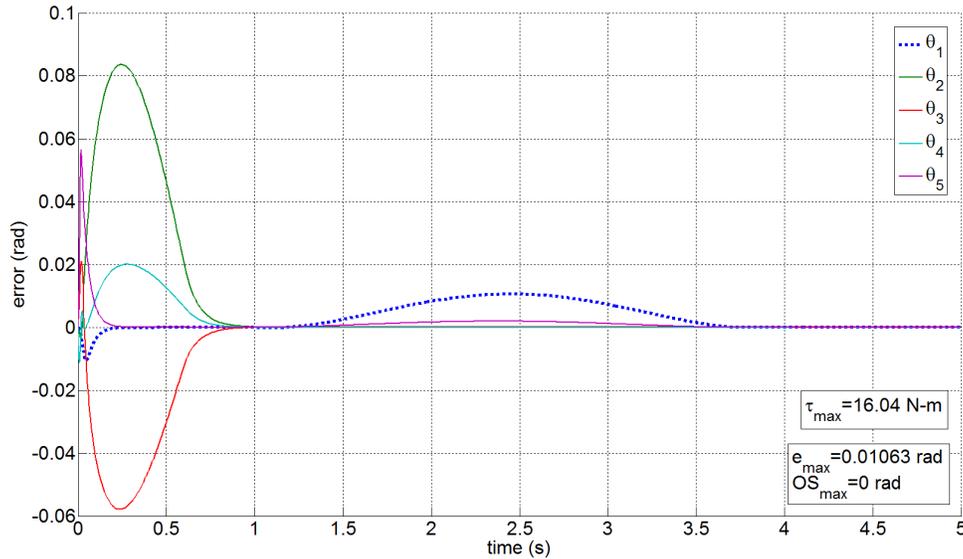


Figure E.3: PD Lin. Control - Joint Errors for Joint 1 Step ($-\pi$ to π)

The gains for joint 2 were tuned by having the robot move from $\theta_2 = 0$ to $\theta_2 = \pi/2$, starting from the Caldry position and moving such that the arm was extended straight up. Using the initial gains for joint 2, as found in Table 4.1, and a maximum speed in the trajectory generator of $90^\circ/s$, the PD

independent controller moved the joint through the desired trajectory with a large overall error ($e_{max} = 1.346rad$) as shown in Figure E.4. The large error is due to the current from the controller reaching saturation ($3.5A$) as soon as the movement is initiated at $t = 1s$ and thus slowing the movement of the joint. Of note, the current output from the PD independent controller to hold joint 2 in the Calrdy position was $3.31A$ which is significantly more than the holding current of $0.85A$ output from the Quanser controller when used with the actual robot. It is suspected that the difference in control currents can be attributed to joint friction which was not formally included as part of the SimMechanics model of the robot used for simulation. The proportional gain and maximum speeds for joint 2 will be left as initially proposed, but may need to be adjusted when the controller is used with the actual robot. The derivative gain was adjusted to 13 to reduce the maximum overshoot for the movement to $1.09e-5rad$.

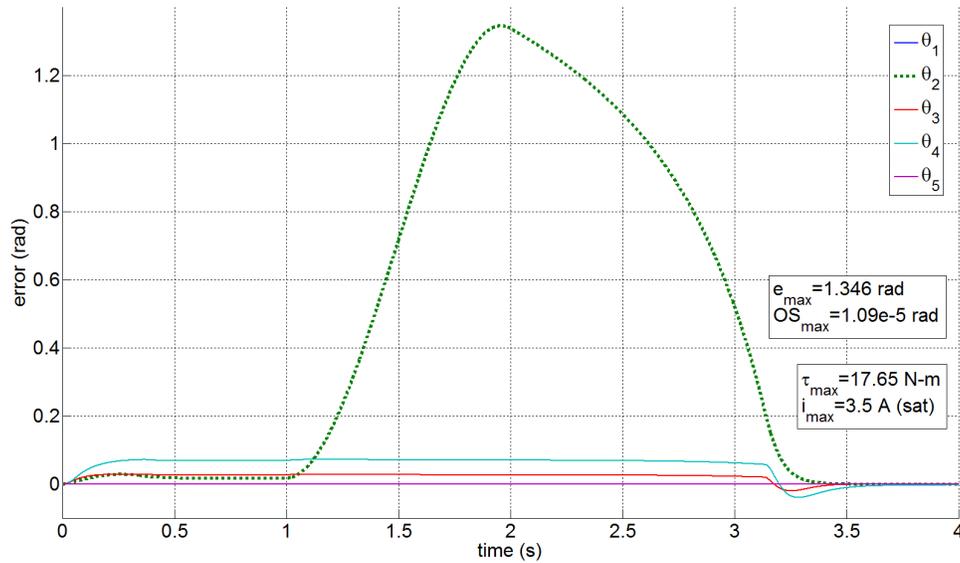


Figure E.4: PD Ind. Control - Joint Errors for Joint 2 Step (0 to $\pi/2$)

The joint space linearized controller, using the initial gains selected for joint 2 of $k_p = 400$ and $k_d = 40$ and a maximum speed in the trajectory generator of $90^\circ/s$, was able to move joint 2 through the prescribed movement with the errors shown in Figure E.5. As with the PD independent controller, the maximum torque ($17.65Nm$) for the joint was reached and maintained for the duration of the movement resulting in a significant error for joint 2

with a maximum error of 1.299rad reached, slightly less than that of the PD independent controller. Where the two controllers greatly differed was in the response of the other joints. The linearized controller caused a movement in joints 3 and 4, attempting to lift their respective links towards the final position, resulting in significant errors for both of these joints. The change in angle of joint 3 effectively reduced the moment of inertia and mass experienced by joint 2 by bringing the center of mass of link 2 and the end effector closer to the Z-axis, thus resulting in the faster movement and slightly smaller error. As described above, the control currents observed when controlling the actual robot are significantly less than the currents (torques) found in simulation, so the gains used with the linearized controller will not be adjusted prior to implementation on the actual robot.

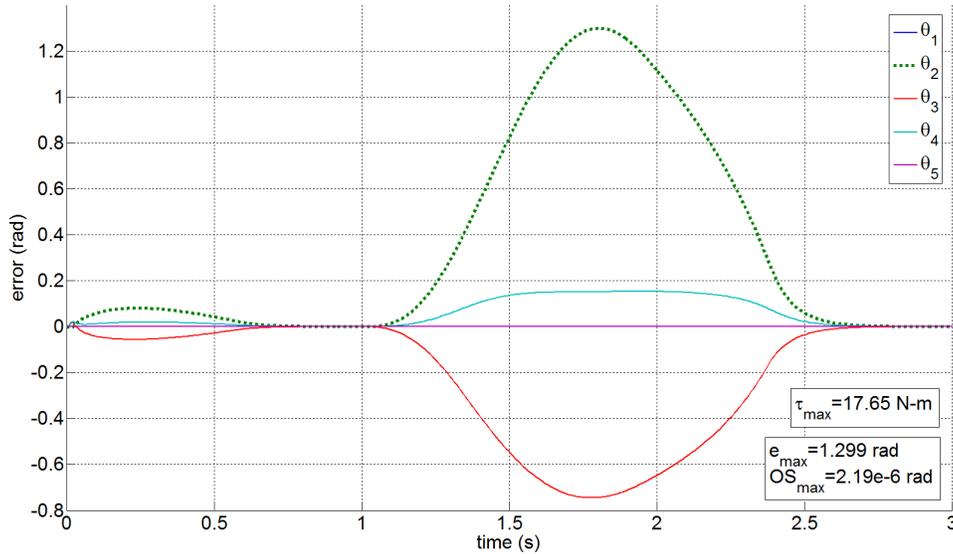


Figure E.5: PD Lin. Control - Joint Errors for Joint 2 Step (0 to $\pi/2$)

The gains for joint 3 were determined by moving the joint from $\theta_3 = -\pi/2$ to $\theta_3 = 0$, starting from the ready position as shown in Figure 3.1 and moving such that the arm was extended straight up. Using the PD independent controller with the initial gains from Table 4.1 and a maximum joint velocity of $90^\circ/s$, the actual joint position followed the desired joint position with a maximum error of $0.04837rad$ and a torque of $11.31Nm$, well below maximum. The maximum joint speed and gains were adjusted through a number of simulations, with the decision that the maximum speed for the PD independent

controller should be $120^\circ/s$ and the proportional and derivative gains 175 and 13 respectively. This combination resulted in the model following the desired trajectory with a maximum error of $0.1447rad$, an overshoot of $1.11e-5rad$ and a torque of $13.32Nm$, with the errors of all joints as shown in Figure E.6. As was observed for the previous 2 joints, the PD independent controller allows a notable steady state error when holding the arm extended in the ready position.

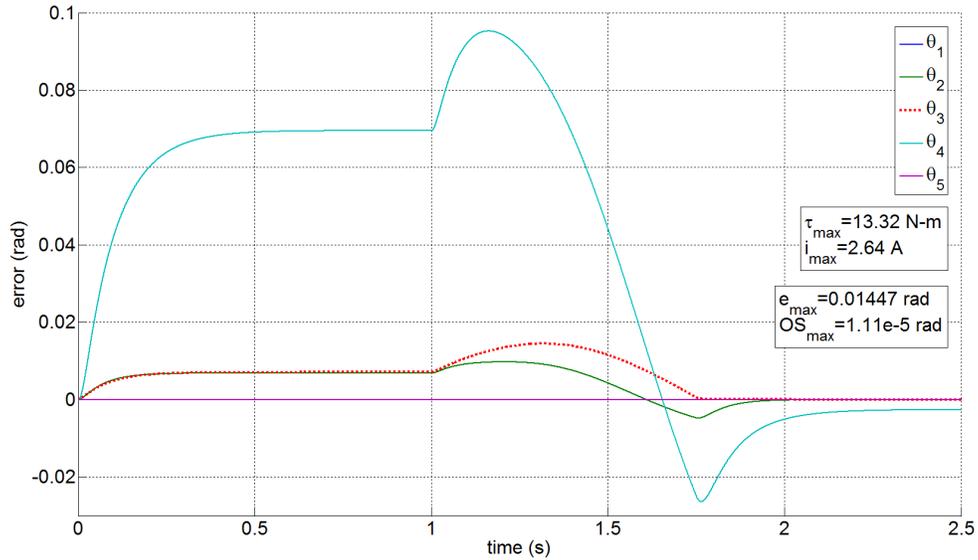


Figure E.6: PD Ind. Control - Joint Errors for Joint 3 Step ($-\pi/2$ to 0)

Using the initial gains of $k_p = 400$ and $k_d = 40$ and a maximum joint speed of $120^\circ/s$, the joint linearized controller was able to follow the desired trajectory for joint 3 with a maximum error of $0.1721rad$ using a actuation torque of $14.41Nm$. To reduce both the error and overshoot, the gains were adjusted through a number of simulations with the final values selected being $k_p = 30000$ and $k_d = 400$. With the same maximum speed, these gains resulted in a maximum error of $0.01196rad$, an overshoot of $1.73e-6rad$ and an actuation torque of $15.37Nm$ as shown in Figure E.7. Similar to what was previously observed for the joint linearized controller, joints 2 and 4 experience a significant error, allowing their respective parts of the serial chain to sag, when joint 3 is actuated. Dependent upon the impact of this error, the gains and maximum speed may need to be further adjusted for joint 3 when this controller is applied to the actual robot.

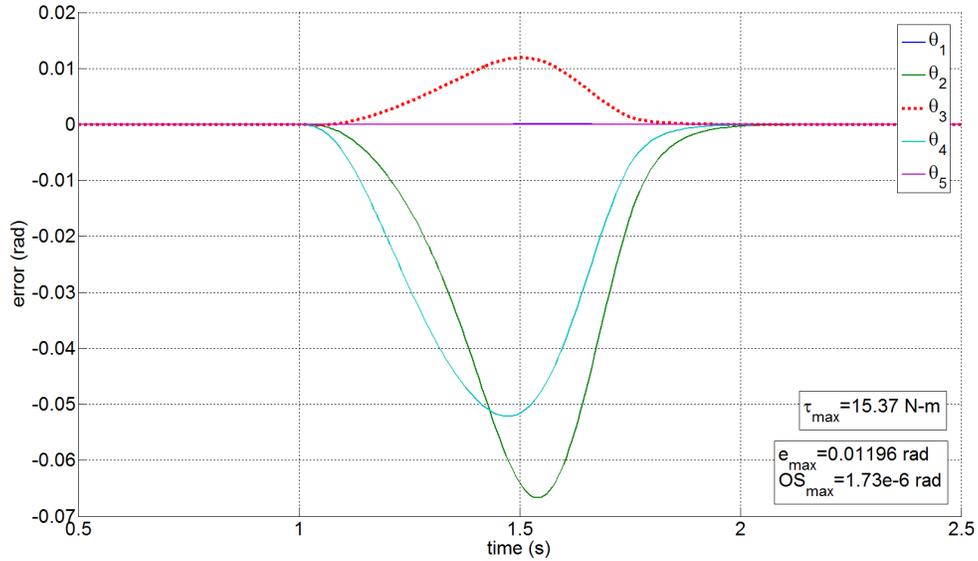


Figure E.7: PD Lin. Control - Joint Errors for Joint 3 Step ($-\pi/2$ to 0)

The gains for joint 4 were tuned by moving the joint from $\theta_4 = 0$ to $\theta_4 = \pi$ with the arm held in the ready position. For the PD independent controller using the initial gains from Table 4.1 and a maximum joint velocity of $225^\circ/s$, the model was unable to follow the desired trajectory as the control current reached saturation ($3.5A$) resulting in a maximum error of $0.6685rad$. Through a number of simulations it was determined that the maximum joint speed using this controller should be $150^\circ/s$, which is notably slower than what the CataLyst-5 is designed for and what has been observed experimentally. The difference in speeds can be attributed to the larger mass and moment of inertia of the end-effector assembly, which includes the F/T sensor and the yaw bracket in addition to the gripper. The gains for joint 4 were selected as $k_p = 40$ and $k_d = 1.8$, resulting in a maximum error of $0.08374rad$, overshoot of $4.71e-4rad$ and actuation torque of $4.64Nm$ as shown in Figure E.8.

Using the initial gains of $k_p = 400$ and $k_d = 40$ and a maximum joint speed of $150^\circ/s$, the joint linearized controller was able to follow the desired trajectory for joint 4 with a maximum error of $0.5271rad$ using a actuation torque of $4.81Nm$ at saturation. To reduce the actuation torque and improve the overall joint response, the maximum joint speed for use with this controller was adjusted to $120^\circ/s$ and the gains tuned to $k_p = 10000$ and $k_d = 200$, resulting in a maximum error of $0.08276rad$, an overshoot of $1.43e-6rad$ and

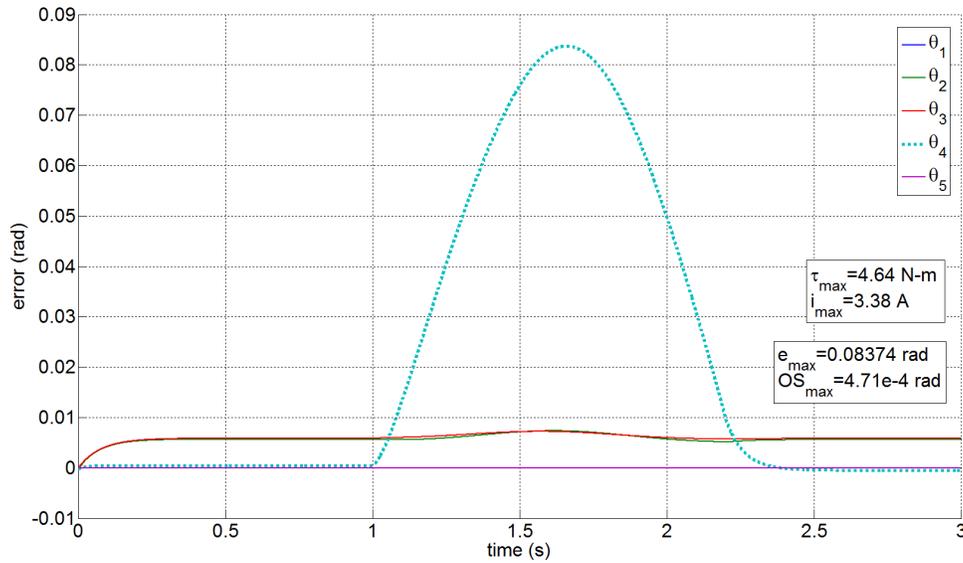


Figure E.8: PD Ind. Control - Joint Errors for Joint 4 Step (0 to π)

an actuation torque of 4.65Nm as shown in Figure E.9. As shown in the figure, the movement of joint 4 using this controller results in a small error for joints 2 and 3 due to the large inertia of the end-effector assembly. As the actuation torque is still greater than the 90% value from Table 4.2 the gains may require further adjustment once the controller is applied to the actual robot.

The gains for joint 5 were tuned by moving the joint through a full rotation from $\theta_5 = -\pi$ to $\theta_5 = \pi$. Using the initial gains from Table 4.1 and a maximum joint velocity of $300^\circ/s$, the PD independent joint controller applied the maximum current (3.5A) for the majority of the movement resulting in an error of $1.935rad$. Through a number of simulations it was determined that the maximum velocity for joint 5 with this controller should be $180^\circ/s$. Setting the gains to $k_p = 40$ and $k_d = 2$ results in a maximum error of $0.07743rad$, overshoot of $1.36e-7rad$ using a maximum applied torque of 2.13Nm as shown in Figure E.10. As was observed for the other joints, the PD independent controller allows for a steady state error for the joints supporting the link and end-effector mass.

Using the initial gains of $k_p = 400$ and $k_d = 40$ and a maximum velocity of $180^\circ/s$ for joint 5, the joint linearized controller applied maximum torque for a portion of the movement resulting in a maximum error of $0.9283rad$. Using a number of simulations it was determined that the maximum speed for joint 5

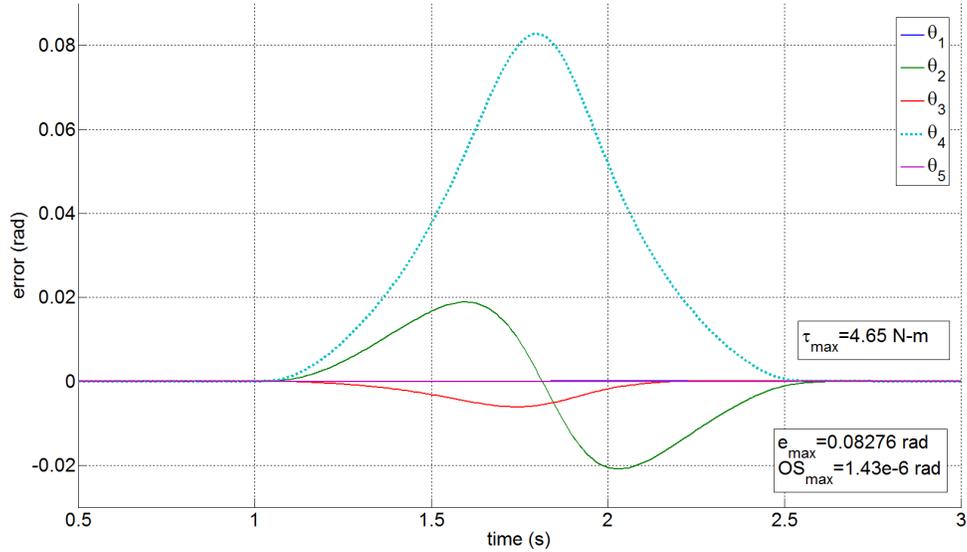


Figure E.9: PD Lin. Control - Joint Errors for Joint 4 Step (0 to π)

using the linearized controller should be $140^\circ/s$ with gains of $k_p = 12000$ and $k_d = 250$. This combination resulted in the controller following the desired trajectory with a maximum error of $0.1007rad$ and overshoot of $1.0e-5rad$ using a maximum applied torque of $2.07Nm$ as shown in Figure E.11. Of note, Figure E.11 shows small errors for joints 1 and 4 during the actuation of joint 5. This again can be attributed to the center of mass of the end-effector assembly being relatively large and located off the joint 5 axis resulting in small disturbances being felt at the other joints.

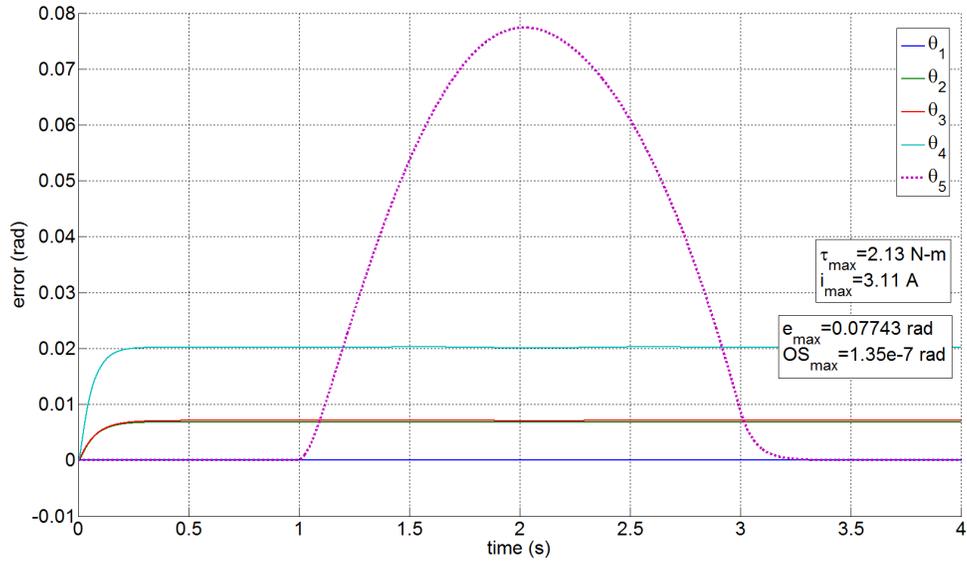


Figure E.10: PD Ind. Control - Joint Errors for Joint 5 Step ($-\pi$ to π)

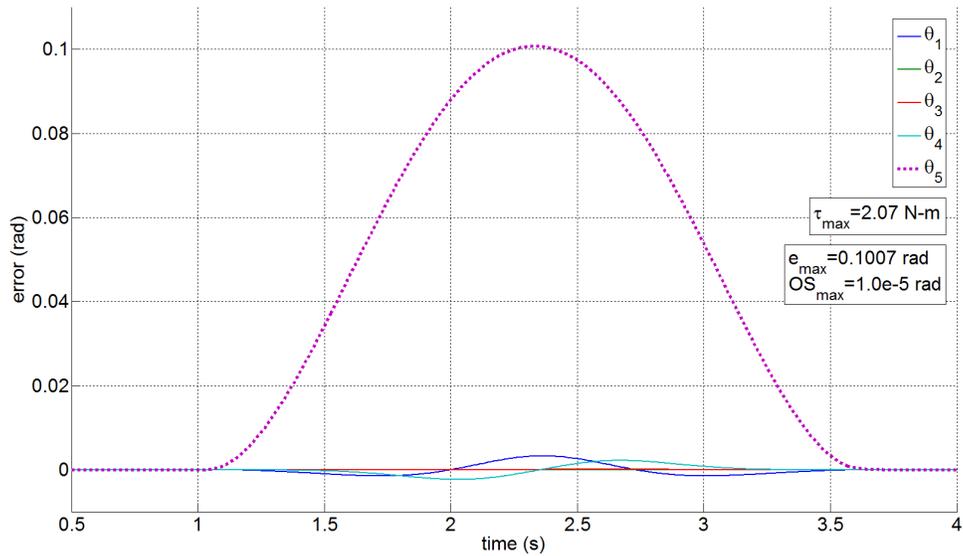


Figure E.11: PD Lin. Control - Joint Errors for Joint 5 Step ($-\pi$ to π)

F Simulation Results - Spiral Trajectory

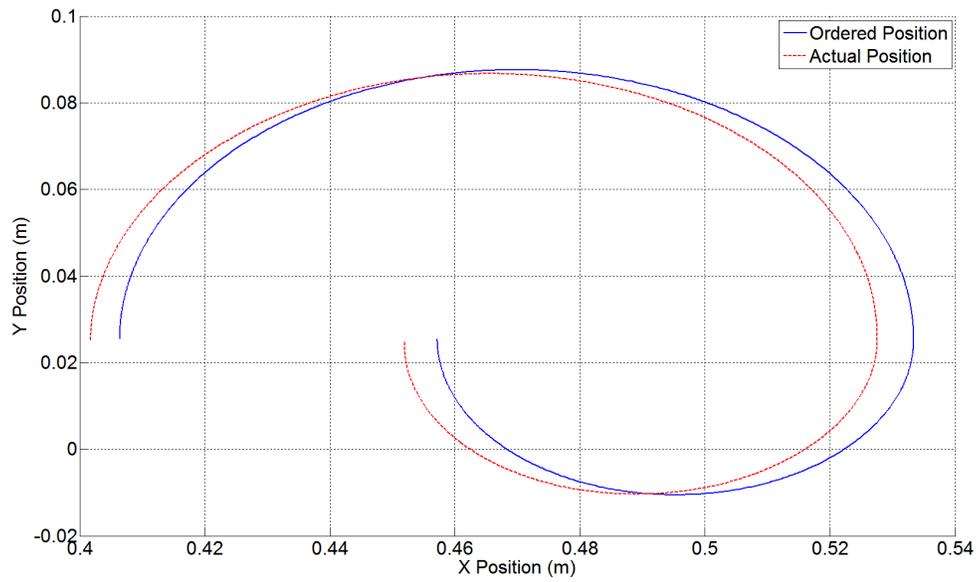


Figure F.1: PD Independent Control - Spiral Trajectory

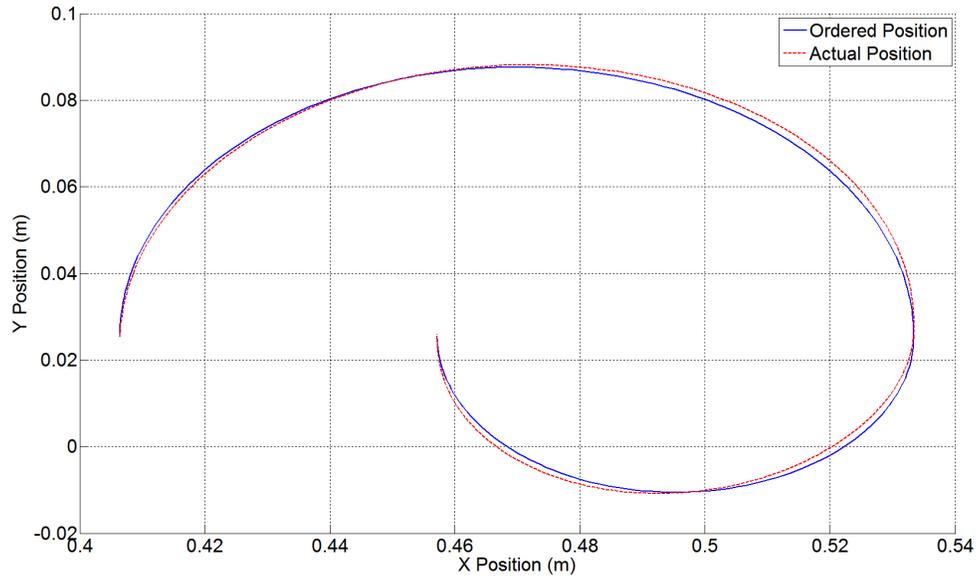


Figure F.2: Joint Space Linearized Control - Spiral Trajectory

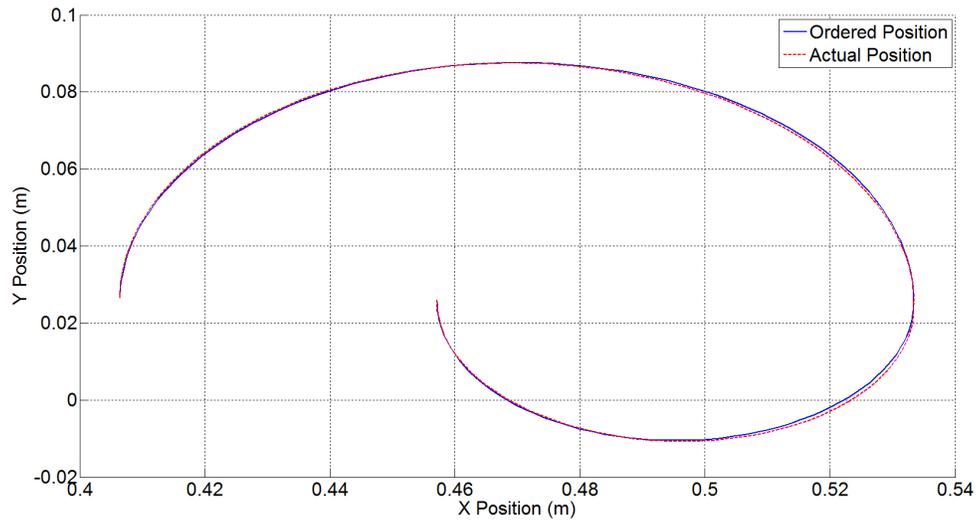


Figure F.3: Cartesian Space Linearized Control - Spiral Trajectory

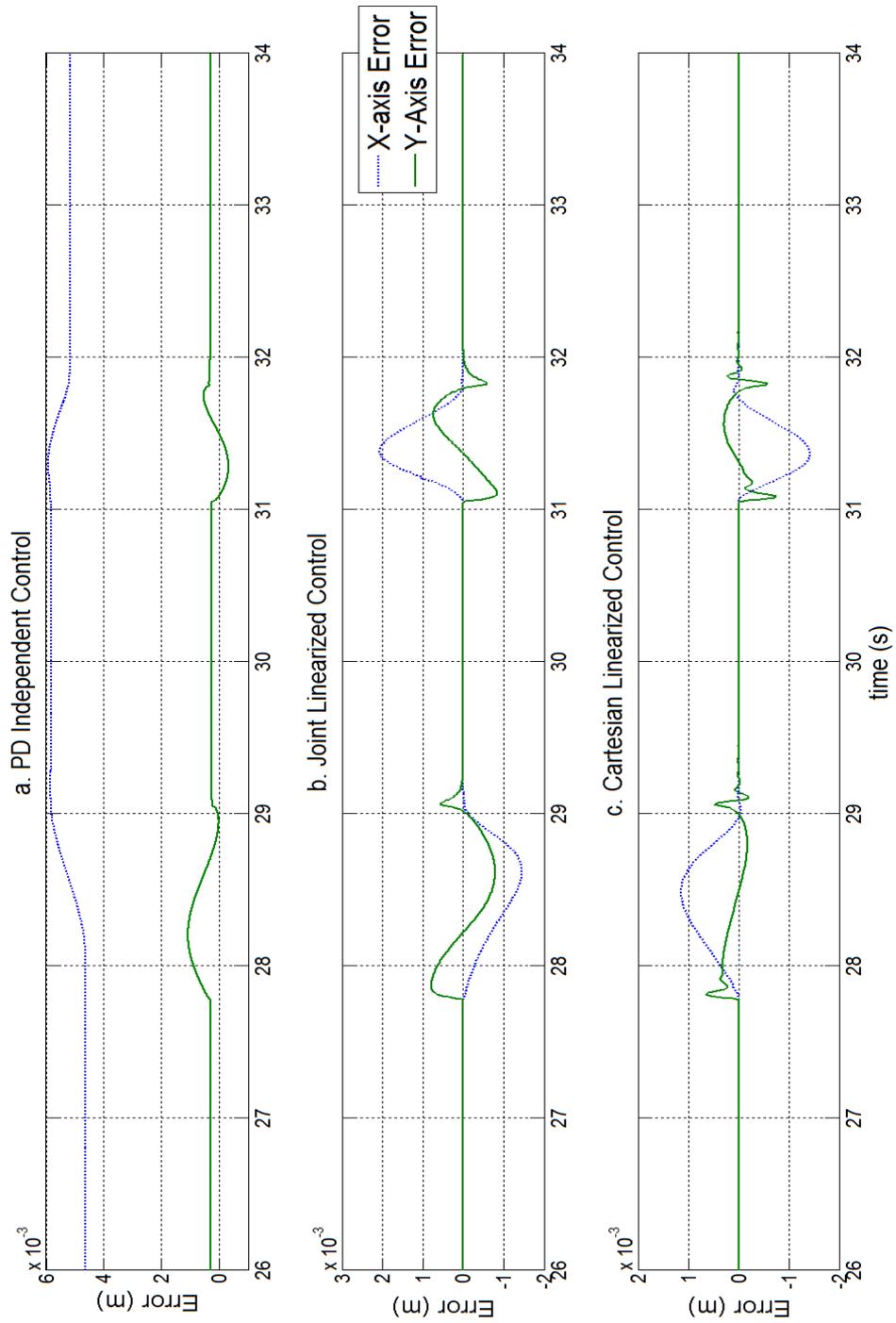


Figure F.4: Trajectory Tracking Errors X & Y Axes - Spiral Trajectory 151

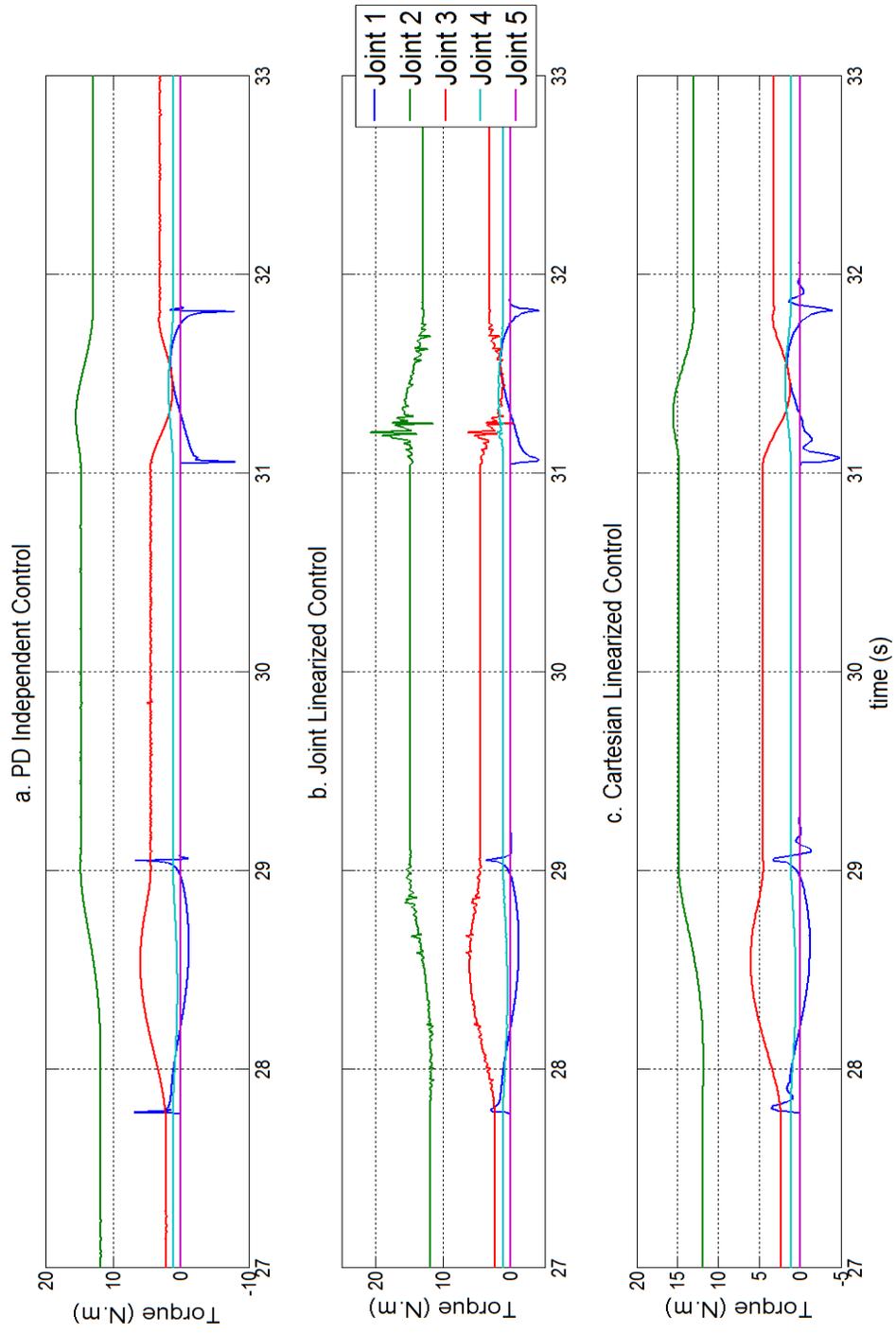


Figure F.5: Control Torques - Spiral Trajectory

G Force/Torque Sensor

The CataLyst-5 at RMC is fitted with an Assurance Technologies Inc. model 15/50 force/torque sensor, with the specifications found in Table G.1. The sensor was reused from previous research conducted at RMCC on a 2-DOF mechanism as described in [66]. The F/T sensor system consists of the transducer itself, the F/T controller and the associated cabling. The transducer is described in [67] as structurally compact, rugged and monolithic and used to measure forces and torques as analog strain gauge displacement data for conversion using the F/T controller. Commonly used as a wrist sensor for robot applications, it is designed with overload pins to protect the transducer from excess loading.

Table G.1: ATI 15/50 Force/Torque Sensor Parameters (adapted from [67])

		F_X/T_X	F_Y/T_Y	F_Z/T_Z
Sensing Ranges	Force (N[lbf])	66.7 [15]	66.7 [15]	66.7 [15]
	Torque (Nm[lbf in])	5.65 [50]	5.65 [50]	5.65 [50]
Resolution	Force (mN[ozf])	55.6 [0.2]	55.6 [0.2]	111.2 [0.4]
	Torque (mNm[ozf in])	2.82 [0.4]	2.82 [0.4]	2.82 [0.4]
Stiffness	Linear (MN/m[lbf/in])	8.76 [50E3]	8.76 [50E3]	17.5 [100E3]
	Torsional (kNm/rad[lbf in/rad])	10.2 [90E3]	10.2 [90E3]	15.8 [140E3]
Overload Protection	Force (N[lbf])	667 [150]	667 [150]	1334 [300]
	Torque (Nm[lbf in])	56.5 [500]	56.5 [500]	56.5 [500]
Analog Pin Number		#5/#2	#4/#1	#3/#0

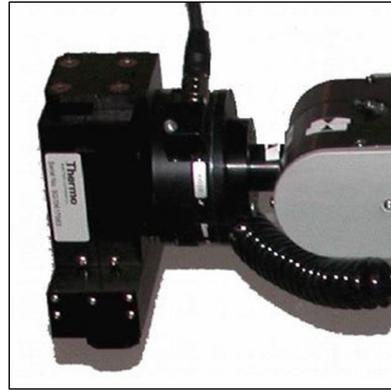
The F/T controller is used to convert the analog strain gauge data received from the transducer into Cartesian force and torque measured values. The rear panel of the controller contains a number of connection ports. The two ports that are used in the RMC installation are the cable connection between the transducer and the controller and an analog output port that sends analog force and torque signals to the modified panel on the back of the

C500D controller to be read via the Q8 card. In the current F/T controller configuration, the analog cable pin configuration is as per Table G.1.

In order to connect the force/torque sensor between the robot arm and the gripper, a yaw bracket had to be designed and manufactured in order to bolt the pieces together. The yaw bracket results in the orientation of the gripper changing from that in Figure G.1a to Figure G.1b.



(a) Normal Configuration



(b) F/T Sensor Configuration

Figure G.1: CataLyst-5 Gripper Configurations

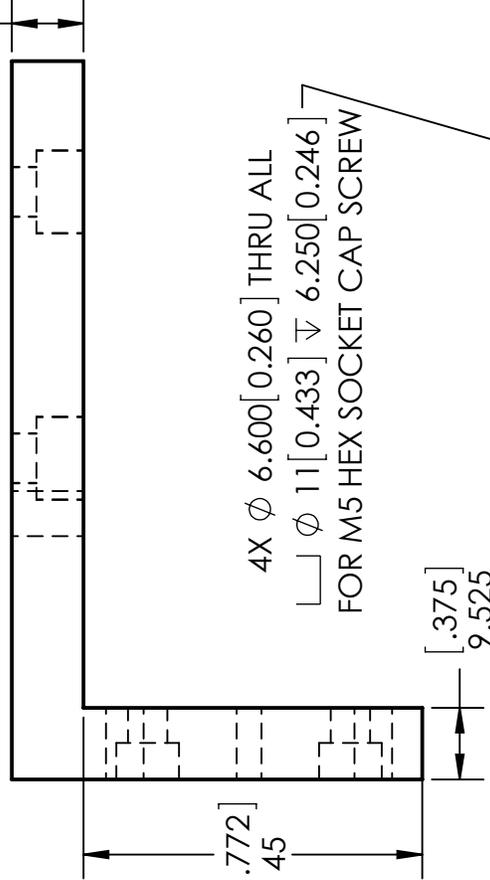
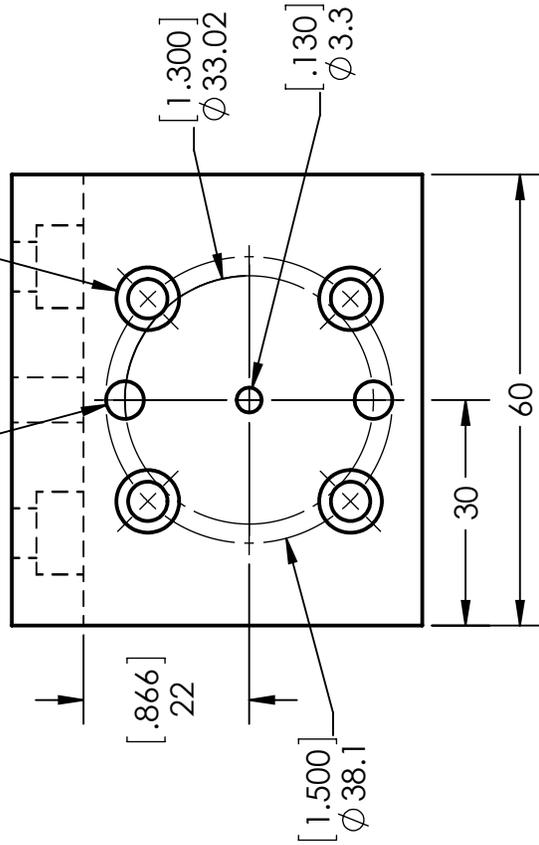
As the F/T sensor was recycled from previous research, a new transducer backplate specific to this application was also manufactured. Dimensioned drawings for both the yaw bracket and the backplate can be found in Appendix H.

H Manufactured Parts

2X $\phi 5$ [.197] THRU
FOR M5 DOWEL PIN
ON $\phi 1.3$ IN HOLE CIRCLE

4X $\phi 5.232$ [0.206] THRU
 $\phi 8.331$ [0.328] $\nabla 4.826$ [0.190]
FOR #10-24 HEX SOCKET CAP SCREW

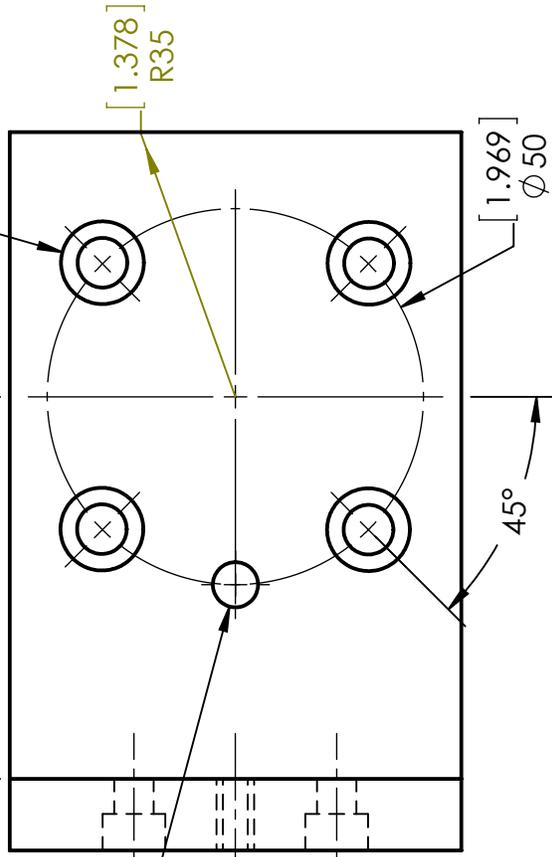
[.375]
9.525



4X $\phi 6.600$ [0.260] THRU ALL
 $\nabla \phi 11$ [0.433] $\nabla 6.250$ [0.246]
FOR M5 HEX SOCKET CAP SCREW

[.375]
9.525

[2.000]
50.8



M6 DOWEL PIN

BRACKET TO BE FABRICATED FROM
3/8 IN ALUMINUM PLATE

DIMENSIONS ARE IN MILLIMETERS [INCHES]

MATERIAL MATIERE	1060 Alloy	WEIGHT MASSE	203.323	9
CREATED DATE DATE DE CREATION	September-24-13 12:16:21			
SAVED SAUVEGARDE	October-10-13 15:08:30			
FOLDER DOSSIER	C:\Users\lmooney\Google Drive\Thesis Information\3_Parts to Manufacture\			
RMCC ENGINEERING FACULTY	CMRC FACULTE D'INGENIERIE	SI	SCALE ECHELLE	1:1
		TOUS LES DIMENSIONS SONT EN MILLIMETRES ALL DIMENSIONS ARE IN MILLIMETERS		
		X.X = ± 0.1 X.XX = ± 0.01		
TITLE TITRE	YAW BRACKET			
NUMBER NUMERO	S22935	FILE NAME NOM FICHIER	Yaw Bracket	
SHEET PAGE	1 OF 1	DE	001	
LT(N) LANCE MOONEY		DWG # / DESSIN #		

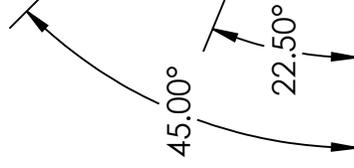
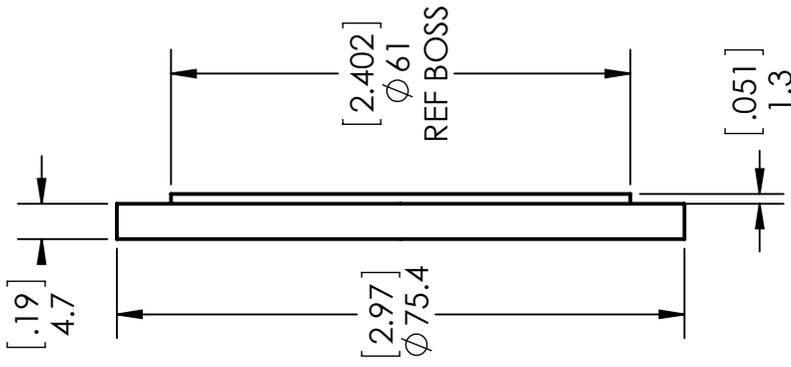
SLIP FIT FOR
M3 DOWEL PIN

[2.68]
Ø 68

2X M5 DOWEL PIN
ON Ø 1.3IN HOLE CIRCLE

8X Ø 3.4 [0.134] THRU ALL
FOR M3 FLAT HEAD SCREW

4X Ø 3.797 [0.150] THRU ALL
FOR #10-24 HEX SOCKET CAP SCREW
ON Ø 1.5IN BOLT CIRCLE

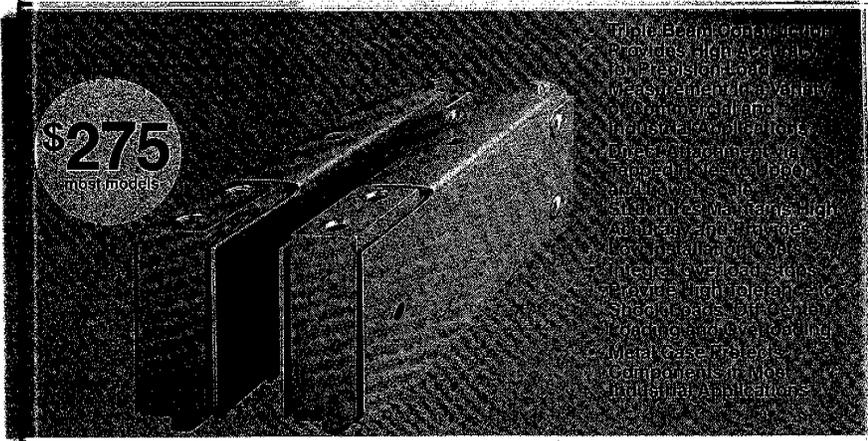


DIMENSIONS ARE IN MILLIMETERS [INCHES]

MATERIAL MATIERE	1060 Alloy	WEIGHT MASSE	64.227	9
CREATED DATE DATE DE CRÉATION	September-24-13 11:06:14			
SAVED SAUVÉ	December-18-14 21:08:13			
FOLDER DOSSIER	C:\Users\mooneya\Google Drive\Thesis Information\3. Parts to Manufacture\			
RMCC ENGINEERING FACULTY	CMRC FACULTÉ D'INGÉNIÉRIE	SCALE ÉCHELLE	1:1	TITLE TITRE
 <small>INTERNATIONAL SYSTEM OF UNITS TOUS LES DIMENSIONS SONT EN MILLIMÈTRES</small>		X.X = ± 0.1 X.XX = ± 0.01	NAME NOM	TRANSUDCER BACKPLATE
			FILE NAME NOM FICHIER	LT(N) LANCE MOONEY
			NUMBER NUMÉRO	s22935
			SHEET PAGE	1 OF 1
			DESIGN#	001

I Load Cell Information

ECONOMICAL, HIGH ACCURACY LCA PLATFORM LOAD CELL FOR LOADS 2 TO 200 LBS.



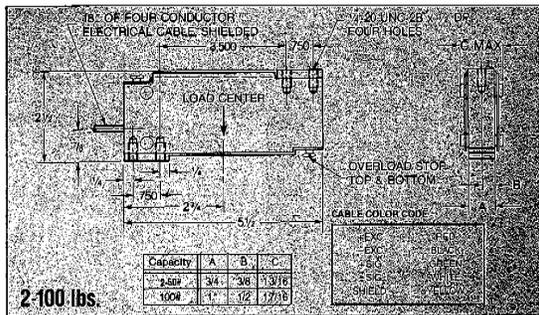
\$275
most models

Triple Beam Cell with which provides high accuracy to precision load measurement. Available in 20 models. Standard models available in 2 lb. to 200 lb. capacity. High accuracy, high resolution, and high overload tolerance. Integral overload stops provide high tolerance to shock loads, off-center loading and repeated overloads. The LCA attaches directly to the upper and lower scale structures, which simplifies assembly, lowers manufacturing cost, and ensures reliable and accurate scale operation.

The LCA triple beam platform load cell is ideal for precision weight and force measurement in commercial, postal, counting and check weighing applications. Unique, integral overload stops provide high tolerance to shock loads, off-center loading and repeated overloads. The LCA attaches directly to the upper and lower scale structures, which simplifies assembly, lowers manufacturing cost, and ensures reliable and accurate scale operation.

SPECIFICATIONS

- Rated Output: 2.2 ± 0.2 mV/V, nominal
- Nonlinearity: 0.02%
- Hysteresis: 0.02%
- Nonrepeatability: 0.01%
- Creep in 20 min.: 0.03%
- Zero Balance: 5.0%
- Compensated Temp. Range: +15 to +115°F (-10 to +45°C)
- Temp. Effect: Output (0.08% of load/100°F)
- Zero Balance (0.15%/100°F)
- Terminal Resistance: Input 400 ohms min., Output 350 ± 8 ohms
- Excitation Voltage: 10 V nominal, 15 V maximum
- Insulation Resistance: 5000 at 50 V dc megohms minimum
- Max. Load: Safe 150%; Ultimate without overload stops engaged 300%



In Order (Specify model number)

Model	Price	Rated Capacity	Deflection at Rated Load	Weight (lbs.)	Meter*
LCA 2	\$295	2 lbs.	.016"	.7	DP 350, DP 2000-S5
LCA 5	275	5 lbs.	.021"	.9	DP 2000-P1, DP 2000-S3
LCA 10	275	10 lbs.	.018"	.9	DP 2000-P2, DP 2000-S4
LCA 20	275	20 lbs.	.012"	.9	DP 2000-P4, DP 2000-S5
LCA 50	275	50 lbs.	.008"	.9	DP 350, DP 2000-S3
LCA 100	275	100 lbs.	.009"	1.2	DP 2000-P2, DP 2000-S4
LCA 200	275	200 lbs.	.005"	1.2	DP 350, DP 2000-S5

LOAD CELLS 105

J RMC CataLyst-5 Open Architecture Operating Procedures

1. Power on the C500D controller using on/off switch, the ROBCOMM computer and the QuaRC (networked) computer.
2. Ensure KVM and serial cable switch are set to position 1 (marked ROBCOMM).
3. Log in to ROBCOMM computer with password *'mechlab123'*.
4. Press the Arm Power button on the front of the C500D and home the robot using the following procedure:
 - a) Start ROBCOMM and open a console window.
 - b) Type *'autohome'* in the console window. (NOTE: if autohome fails move to step e)
 - c) Once homing is complete type *'ready'*.
 - d) Once robot is at the ready position close ROBCOMM.
 - e) If autohome fails, type *'pendant'* and use teach pendant to align all 5 joint markers, then exit from the pendant back to the console.
 - f) Type *'home'* in the console window.
 - g) Once homing is complete type *'ready'*.
 - h) Once robot is at the ready position close ROBCOMM.
5. Switch KVM and serial cable switch to position 2 (marked QuaRC).
6. Log in to the QuaRC computer with RMC network username and password.
7. Open properly configured controller in Matlab/Simulink.
8. Build the real-time code corresponding to the controller model by clicking the Build button or selecting Build from the QuaRC pull down menu.
9. When the build is complete, click the Connect to Target button followed by the Start button. *Be ready to press the E-stop button on the teach pendant when running an unproven controller.*

-
10. If using the Quanser takeover block, switch the mode block from 1 to 0 to enter open architecture control mode.
 11. Perform control sequence on the robot, ensuring it is returned to the ready position on completion.
 12. Switch the mode control block from 0 to 1 exit open architecture control mode.
 13. Stop the controller by pressing the Stop button.
 14. Once experimentation is complete power off the robot using the following steps:
 - a) Log off from QuaRC computer and switch KVM and serial cable switch to position 1.
 - b) Start ROBCOMM and open a console window.
 - c) Type 'ready' in the console window.
 - d) Once the robot is in the ready position type 'shutdown now'.
 - e) Power off the C500D controller using on/off switch when the panel on the front (or the console window) indicates it is safe to do so.