

# Real-time implementation of some Bilateral Teleoperation schemes

BY  
ALAIN RICHARD

UNDER THE SUPERVISION OF DR. ABDELHAMID TAYEBI

A thesis submitted to the faculty of graduate studies  
Lakehead University  
in partial fulfillment of the requirements for the degree of  
Master's of Science in Control Engineering

Electrical Engineering

Lakehead University

August 2010

# Abstract

Today's technology has pushed back many boundaries that we thought were impossible. One of these technologies is bilateral teleoperation. Bilateral teleoperation allows an operator to control a robot, at a distance, over a communication medium and get a feedback of the interaction forces between the robot's end-effector and the remote environment. This technology leads, for instance, to application in telesurgery, hazardous material handling and underwater repairs.

Our main research is to compare different control methods for such a system using a six degree of freedom (DOF) parallel robot for the master device and a 6 DOF serial robot for the slave device. According to the choice of the transmitted variables over the communication medium, we attempt to show the inherent differences between transmitting position/sensor force information, velocity/control force information and, thereafter, wave variables.

The Position Error Based (PEB) controller, the Kinesthetic Force Based (KFB) controller and the transmission of Wave variables are implemented. Experimental results with and without time delays present are carried out and compared. For our implementations a haptic device, based on the twin pantograph architecture, is used as a master manipulator, while the slave manipulator is a 6 DOF serial robot manipulator, the A465 Robot by CRS Robotics.

# Acknowledgements

I am sincerely grateful to my supervisor, Prof. Abdelhamid Tayebi, for his many suggestions and constant support during this research. This thesis would not have been possible without his support and guidance. I highly appreciated the financial support received from Lakehead University and OGSST, thank you.

I also would like to thank Mr. Bruce Misner and Aaron Cole for allowing me to bounce ideas around with them and their constant support.

Finally, I wish to thank and express my appreciation to my colleagues in the Automatic Control Laboratory, Omid Ehtemam-Haghighi and Fereshteh Ghashghaee. I cordially wish them the best in all their future endeavours.

Thunder Bay, Ontario  
August 20, 2010  
Alain Richard

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>ii</b>
<b>Contents</b>	<b>iii</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>ix</b>
<b>1 Background</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Motivation . . . . .	1
1.3 Description . . . . .	2
1.4 History Overview . . . . .	2
1.5 Control schemes . . . . .	3
1.5.1 Supervisory Control . . . . .	4
1.5.2 Position error based (PEB) scheme . . . . .	5
1.5.3 Kinesthetic force-based (KFB) scheme . . . . .	6
1.5.4 4 Channel Architecture . . . . .	7
1.5.5 Wave Variables . . . . .	8
1.5.6 Wave variables in Hamiltonian dynamics . . . . .	11
1.5.7 Wave variables over the internet . . . . .	11
Wave integral transmission . . . . .	12
Communication signal management . . . . .	13
1.6 Outline . . . . .	15

<b>2</b>	<b>Description of the Slave Manipulator</b>	<b>16</b>
2.1	Introduction . . . . .	16
2.2	Forward Kinematics of the CRS robot . . . . .	17
2.3	Jacobian . . . . .	19
2.4	Inverse Kinematics of CRS robot . . . . .	20
2.4.1	Kinematics decoupling . . . . .	20
2.4.2	Anthropomorphic Manipulator (RRR) . . . . .	21
	Facing with Elbow Up . . . . .	23
	Facing with Elbow Down . . . . .	24
	Back with Elbow Up . . . . .	25
	Back with Elbow Down . . . . .	26
2.4.3	Spherical Wrist . . . . .	27
2.4.4	Singularities . . . . .	28
<b>3</b>	<b>Description of the Master Manipulator</b>	<b>30</b>
3.1	Introduction . . . . .	30
3.2	Forward Kinematics of the haptic robot . . . . .	31
3.3	Jacobian . . . . .	39
<b>4</b>	<b>Bilateral Teleoperation</b>	<b>47</b>
4.1	System description . . . . .	47
4.1.1	Multi-rate sampling . . . . .	48
4.2	Kinesthetic force-based scheme . . . . .	49
4.2.1	Safety Considerations . . . . .	50
4.3	Position error-based scheme . . . . .	51
4.3.1	Safety Considerations . . . . .	53
4.4	Wave variables . . . . .	55
<b>5</b>	<b>Experimental Results</b>	<b>57</b>
5.1	Experiment with a compliant environment without communi- cation time delay . . . . .	58
5.1.1	Discussion 1 . . . . .	59
5.2	Experiment with a stiff environment without communication time delay . . . . .	62
5.2.1	Discussion 2 . . . . .	63
5.3	Free motion with communication time delay . . . . .	66
5.3.1	Discussion 3 . . . . .	66

5.4	Experiment with a compliant environment with communication time delay . . . . .	68
5.4.1	Discussion 4 . . . . .	69
5.5	Experiment with a stiff environment with communication time delay . . . . .	72
5.5.1	Discussion 5 . . . . .	73
5.6	Problems faced when running the experiments . . . . .	76
<b>6</b>	<b>Conclusions and future work</b>	<b>78</b>
6.1	Conclusion . . . . .	78
6.2	Status and future work . . . . .	79
	<b>References</b>	<b>79</b>
<b>A</b>	<b>Bilateral Teleoperation using WinCon</b>	<b>84</b>
A.1	Single computer set-up . . . . .	85
A.2	Extending WinCon for Bilateral Teleoperation . . . . .	86
A.2.1	Choice of Transmission Blocks . . . . .	87
<b>B</b>	<b>Hardware modifications made to the system</b>	<b>88</b>
B.1	Modifications to the emergency stop button . . . . .	88
B.2	Modifications to the motor control boards . . . . .	88

# List of Figures

1.1	Block Diagram of Bilateral Teleoperation . . . . .	2
1.2	Block diagram of the Position Error Based controller . . . . .	6
1.3	Block diagram of the Kinesthetic Force Based controller . . . . .	6
1.4	4 channel variable exchange . . . . .	7
1.5	2 port power flow block . . . . .	8
1.6	Block diagram of bilateral teleoperation . . . . .	9
1.7	Wave transformation for PEB . . . . .	10
1.8	One way communication of wave variables contrasted . . . . .	12
	(a) Wave transmission . . . . .	12
	(b) Wave integral transmission . . . . .	12
1.9	CMM diagram for one direction of the data path . . . . .	13
2.1	The CRS manipulator . . . . .	16
2.2	Chosen Co-ordinate Frames . . . . .	17
2.3	Arm and Wrist decoupling . . . . .	20
2.4	Solving for $\theta_1$ . . . . .	21
2.5	Two possible torso configuration . . . . .	22
	(a) One pose . . . . .	22
	(b) Alternate pose . . . . .	22
2.6	Solving with elbow up configuration . . . . .	23
2.7	Solving with elbow down configuration . . . . .	24
2.8	Solving with elbow up configuration . . . . .	25
2.9	Solving with elbow down configuration . . . . .	26
2.10	Two possible wrist configurations . . . . .	27
	(a) One way . . . . .	27
	(b) Alternate way . . . . .	27
2.11	Waist in singularity position . . . . .	28
2.12	Spherical wrist in singularity position . . . . .	29

3.1	The twin pantograph manipulator . . . . .	30
3.2	CAD drawing of the twin pantograph . . . . .	31
3.3	Serial equivalency of the 5-bar linkage, ie. the pantograph . . . . .	32
	(a) 3 DOF Pantograph . . . . .	32
	(b) 3 DOF pseudo-serial . . . . .	32
3.4	Origins on the pantograph . . . . .	33
3.5	Unit vector triangle . . . . .	33
3.6	Visual interpretation of $\gamma_1$ . . . . .	34
	(a) Value calculation for $\gamma_1$ . . . . .	34
	(b) Sign calculation for $\gamma_1$ . . . . .	34
3.7	DH frame of reference for both pantographs . . . . .	35
3.8	Finding the two last axes of rotation . . . . .	37
3.9	Determining $q_5$ . . . . .	38
3.10	Distributing the end-effector velocity . . . . .	40
3.11	Two dimension serial manipulator considered . . . . .	43
4.1	Block diagram of control algorithm . . . . .	48
4.2	Block diagram of the PD controller . . . . .	50
4.3	Block diagram of the PI controller . . . . .	52
4.4	Finding the generalized velocity for the slave manipulator . . . . .	53
4.5	PI controller with safety . . . . .	54
4.6	Wave variable block diagram . . . . .	56
5.1	Experimental setup at the Automatic Control Laboratory . . . . .	57
5.2	Independently calculated position error . . . . .	58
5.3	Position error of the KFB controller without time delays . . . . .	59
5.4	Position error of the PEB controller without time delay, error measured at the controller . . . . .	60
5.5	Position error of the PEB controller without time delay, calculated by forward kinematics . . . . .	60
5.6	Position error using wave variables without time delay, error measured at the controller . . . . .	61
5.7	Position error using wave variables without time delay, calculated by forward kinematics . . . . .	61
5.8	Position error of the KFB controller without time delays . . . . .	62
5.9	Position error of the PEB controller without time delay, error measured at the controller . . . . .	63



5.10	Position error of the PEB controller without time delay, calculated by forward kinematics . . . . .	64
5.11	Position error using wave variables without time delay, error measured at the controller . . . . .	64
5.12	Position error using wave variables without time delay, calculated by forward kinematics . . . . .	65
5.13	Position error of the KFB controller with constant time delay of $50ms$ present . . . . .	66
5.14	Position error of the PEB controller with constant time delay of $50ms$ present . . . . .	67
5.15	Position error using wave variables with constant time delay of $50ms$ present . . . . .	67
5.16	Position error of the KFB controller with constant time delay	68
5.17	Position error of the PEB controller with constant time delay .	69
5.18	Position error of the PEB controller with constant time delay, calculated by forward kinematics . . . . .	70
5.19	Position error using wave variables with constant time delay .	70
5.20	Position error using wave variables with constant time delay, calculated by forward kinematics . . . . .	71
5.21	Position error of the KFB controller with constant time delay	72
5.22	Position error of the PEB controller with constant time delay .	73
5.23	Position error of the PEB controller with constant time delay, calculated by forward kinematics . . . . .	74
5.24	Position error using wave variables with constant time delay .	74
5.25	Position error using wave variables with constant time delay, calculated by forward kinematics . . . . .	75
A.1	Software Interaction of WinCon . . . . .	85
A.2	WinCon used for Bilateral Teleoperation . . . . .	86
B.1	The emergency stop circuit . . . . .	89
B.2	Modified 90 watt motor drive . . . . .	90

# List of Tables

1.1	Technologies used to increase telepresence . . . . .	4
1.2	Comparison between 1-DOF to $n$ -DOF . . . . .	10
1.3	Procedures proposed for $n$ -DOF wave transmission . . . . .	11
2.1	DH parameters for the CRS manipulator . . . . .	18
3.1	DH parameters for both pantographs . . . . .	36
4.1	PD controller gains with joint angle in radians . . . . .	49
4.2	PI controller gains with velocity in centimetres per seconds and radians per seconds . . . . .	52
4.3	Torque calculations algorithm with safety feature . . . . .	54
4.4	Wave transformation variables . . . . .	56
5.1	Haptic device force capabilities . . . . .	77
A.1	Installed software and version numbers . . . . .	84

# Chapter 1

## Background

### 1.1 Introduction

Teleoperation, which directly translates to “*controlling at a distance*,” is a field of research that has already had significant impact in the way we accomplish certain tasks and will continue to find new applications to its technology. It is often desired to have an interaction between an operator and some environment without the two ever being physically in contact. They can be in the same room, by being separated with protective barriers, or they can be on different continents, connected in wired or wire-less fashion. If we desire to add a more realistic experience for the user, force feedback may be introduced to the operator.

### 1.2 Motivation

Teleoperation with force feedback is said to be *Bilateral Teleoperation*. This ability to control at a distance while receiving force feedback was motivated by several applications, described in [1], such as:

**Working in hazardous environments**

Nuclear waste handling, High voltage line repairs, Explosive disarming

**High travel cost of expertise**

Space installation and repairs, Complex telesurgery

**Scaling**

Handling heavy objects, Perform minimally invasive surgery

### 1.3 Description

As depicted in Figure 1.1 we have the local site, the location where the operator is working, with the master manipulator and, possibly, receiving visual/audio feedback. The remote site includes the manipulator, referred to as the slave manipulator, and can include an array of sensors including; encoders, force/torque sensors, camera(s) and microphone. The authors in [2] explain how we can see the system as having the brain removed or distant from the body.

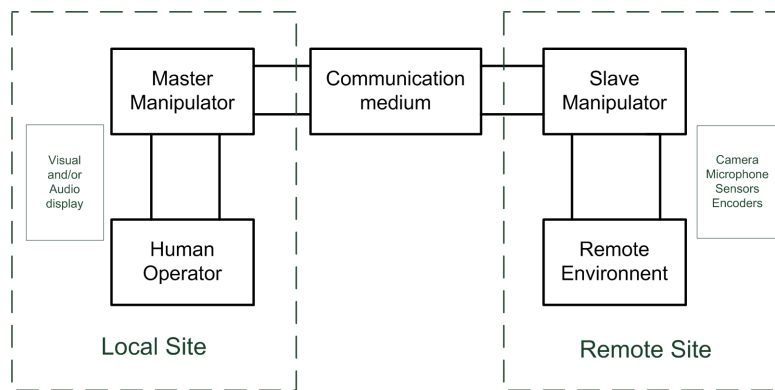


Figure 1.1: Block Diagram of Bilateral Teleoperation

### 1.4 History Overview

The field of teleoperation emerged with the need of new technologies that allowed an operator to safely handle hazardous waste. Throughout the 1940s and 1950s, most teleoperation systems had that goal in mind. Raymond C. Goertz built the first of such systems, where the master and slave were coupled electronically. This set-up had no feedback and the response of the slave device was awkward and slow [3]. He then decided to have two identical robots and link them mechanically. He was able to get a better response but now he was very limited on the distance between the two robots.

In the 1960s, we saw the introduction of force feedback to provide more information to the operator in systems coupled electronically. Force feedback lead to some problems in performance and stability. Also around this time,

engineers started to notice the destabilizing effect of time delays if the system was used in bilateral teleoperation. They also noticed a drop in performance if the system was used without force feedback. To remedy those issues different schemes of supervisory control were developed. Later came a more rigorous analysis using Lyapunov theory in Cartesian space [3]. This allowed robots to be kinematically and dynamically different.

At the end of the '80s Anderson and Spong [4] analyzed the Bilateral Teleoperator as electronic components and compared it with transmission line theory. Transmission line theory was a field with much more understanding and the comparison lead to an encoding of variables, as *waves*, that could show the communication system to be passive, even with time delays present. This was further demonstrated using different analysis in [4, 5, 6, 7].

More recently, the first transatlantic demonstration without force feedback happened in 2001 over a dedicated communication line. Today most of the research is focused on solving problems related to the use of internet as a communication medium. Given the internet's variable time delays and possible packet loss, current methods struggle to perform adequately.

## 1.5 Control schemes

Many different control schemes were developed for bilateral teleoperation. We will try to summarize some of them and point out their advantages and drawbacks. To do so one must be aware that the goal of bilateral teleoperation is that the slave robot follows the movement of the master robot and provides feedback of the interaction forces it feels with the remote environment. The control schemes can be of different forms. They can be of single joint control if both robots are identical or they can seek to control Cartesian variables. Also if the master and slave robots are kinematically different, clatching<sup>1</sup> may be used such that the entire workspace is accessible [2]. All controllers seek to achieve ideal telepresence and do so while guarantying stability of both manipulators.

---

<sup>1</sup>Clutching can be compared to when we lift the mouse in order to replace it on the mat without affecting the position of the cursor on the computer screen. Clutching is often performed with the help of a push button on the end-effector of the master manipulator.

## Telepresence

Telepresence was first introduced in [8], it is a measure of how realistic the experience is for the operator of some system. In our case it is determined by the transparency of the telerobotic system. The ideal system would be fully transparent to the operator and the operator would be fooled to think they are *at* the remote site without realizing everything in between (the master manipulator, the communication medium and the slave manipulator). This can be done by maximizing the use of human senses with visual, audio and force feedback, Figure 1.1.

Telepresence technologies	
Sense	Method
Sight	IP camera(s)
Hearing	microphone and speakers
Touch	haptic devices

Table 1.1: Technologies used to increase telepresence

### 1.5.1 Supervisory Control

The Supervisory Control scheme is one of the first schemes to try to deal with the problems associated with time delays. Instead of transmitting a trajectory for the slave to follow, we transmit information of a much higher level; commands. This requires intelligence on the slave side in order to understand these commands and have the ability to execute them. Supervisory control was introduced by Ferrell and Sheridan [9] and was used until the end of the 1980s. An example of such a command could be to hold the end-effector perpendicular to the ground such that when carrying liquids the slave would not spill regardless of what the master manipulator requested. It is similar to a workplace supervisor-employee relationship where the supervisor would ask the employee to accomplish a certain task without giving them all the steps. The employee knows how to carry on the task and knows when to provide feedback to the supervisor on how things are going. Supervisory control

can include predictive loops, at the master side, to display the predicted trajectory of the slave manipulator before it has happened, thus removing the ambiguity due to the communication delays. This allows the user to know what response to expect from the slave manipulator even before it has started moving. Telepresence in such schemes is poor since the operator is never in full control of the slave manipulator. Since most of the control is done at the slave side, stability is not affected by the time delay; hence, this scheme is usually referenced as stable.

### 1.5.2 Position error based (PEB) scheme

The PEB<sup>1</sup> control scheme uses a proportional-derivative (PD) controller on both the master and the slave manipulator. The PD controller is known to be stable and accurate [11, 12]. This means that the slave follows the movement of the master. Meanwhile the master controller tries to force the master to follow the slave. This can be done in joint space, for identical manipulator, (1.1), or in Cartesian space, for dissimilar manipulators, (1.2).

$$\begin{aligned}\tau_s &= K_p(q_m - q_s) + K_d(\dot{q}_m - \dot{q}_s) \\ \tau_m &= K_p(q_s - q_m) + K_d(\dot{q}_s - \dot{q}_m)\end{aligned}\tag{1.1}$$

$$\begin{aligned}F_c &= K_p(x_m - x_s) + K_d(\dot{x}_m - \dot{x}_s) \\ F_m &= K_p(x_s - x_m) + K_d(\dot{x}_s - \dot{x}_m)\end{aligned}\tag{1.2}$$

If the gains are set equal we can find that the force applied to both manipulators will be the same but in opposite direction, (1.3), demonstrated on Figure 1.2 where  $F_h$  and  $F_{env}$  represent the human and environment force respectively. Although this method gives a good idea of the forces required to move the slave manipulator, it also feeds back to the operator the slave's inertia forces and the forces from the spring-damper representation of the transmission medium. Hence, this system does not provide accurate feedback of the environment forces. Without time delay, this method is very stable but fails when time delays are introduced in the communication medium.

$$\begin{aligned}F_c &= K_p(x_m - x_s) + K_d(\dot{x}_m - \dot{x}_s) \\ F_m &= -F_c\end{aligned}\tag{1.3}$$

---

<sup>1</sup>This nomenclature follows from [10]

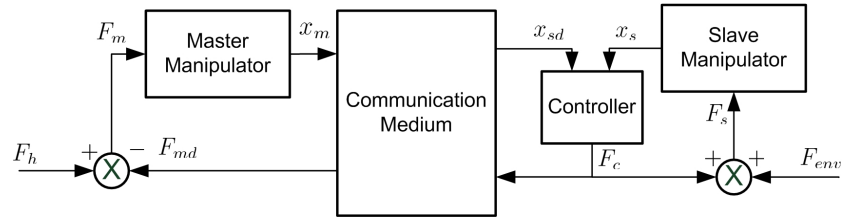


Figure 1.2: Block diagram of the Position Error Based controller

### 1.5.3 Kinesthetic force-based (KFB) scheme

To overcome the problem of inaccurate force feedback from the PEB scheme the KFB scheme includes a force and torque acquisition unit placed on the end-effector of the slave manipulator, Figure 1.3, [10]. This implies that interaction forces are directly sensed and fed back to the master manipulator. Force sensor naturally includes many high-frequency terms in the signal while motors usually have slow time-constants. This difference will lead to instability if the end-effector finds itself in contact with a stiff environment. Roughly speaking, right before the contact happens, a very small motion will create a large interaction force which leads to instability, [2, 7]. This non-passive response in turns disallows stability. If the KFB scheme is used without stiff contact then the user will have a great sense of telepresence while maintaining stability.

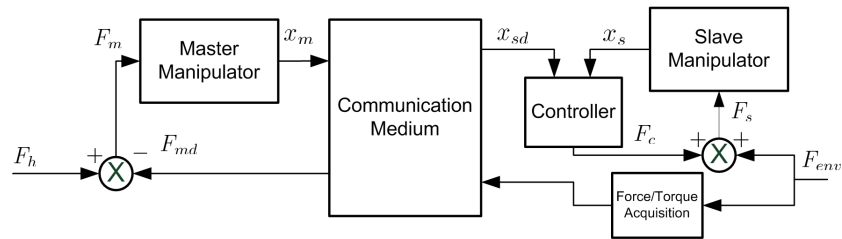


Figure 1.3: Block diagram of the Kinesthetic Force Based controller



### 1.5.4 4 Channel Architecture

The scheme that provides the most transparency to the user is the 4-channel architecture. Instead of transmitting only one variable between the master and the slave, we transmit 2 variables. A force and torque acquisition unit is added to the master end-effector as compared with the KFB controller. We transmit both the velocity and the force sensed in each direction, Figure 1.4. With the extra force sensor on the master manipulator, this allows for a faster response of the slave robot. When the operator starts applying pressure to the master robot the sensor senses it and commands the slave to start moving before the master manipulator has even had time to start moving itself, yielding unrivalled performance, [13]. Although ideal transparency can be obtained, this architecture is not stable in the presence of time delays [14]. It was then concluded that “passivity and transparency are conflicting objectives in teleoperator system design” [14].

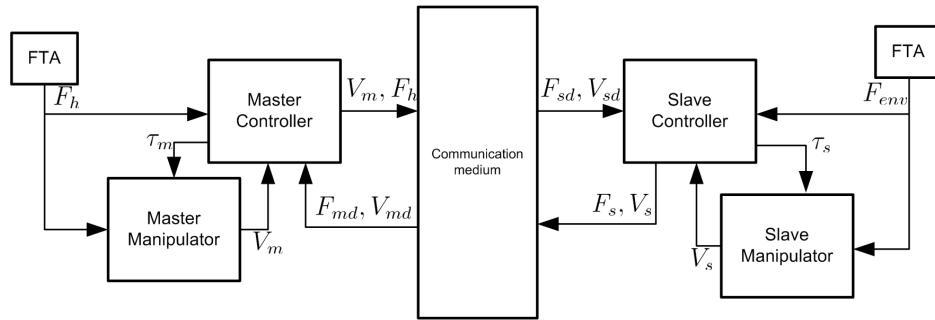


Figure 1.4: 4 channel variable exchange

In Figure 1.4, the FTA block represents a force/torque acquisition unit placed on the end-effector of the corresponding manipulator. Starting from the master manipulator, on the left side of Figure 1.4, the FTA unit detects the force applied by the user,  $F_h$ . This force is applied to the master manipulator and makes the manipulator move at a velocity of  $V_m$ . Both  $V_m$  and  $F_h$  are combined by the controller and transmitted to the slave controller. The slave controller receives these values and they become targets for the slave manipulator,  $V_{sd}$  and  $F_{sd}$ . Simultaneously a FTA unit on the slave manipulator detects the interaction forces between the remote environment and the slave manipulator’s end-effector. Having the desired force and velocity, the slave manipulator determines what torque to apply to the slave,  $\tau_s$ . This

process is also mirrored from slave to master in order to determine the torque to apply to the master manipulator,  $\tau_m$ .

### 1.5.5 Wave Variables

Wave variables were introduced as a method to address the stability issues in the presence of communication time delays. If we use the PEB controller and look at the flow exchange (voltages, forces) and efforts (currents, velocities) we can alter (1.2) to (1.4).

$$F_c(t) = K_i \int_0^t (\dot{x}_m(s) - \dot{x}_s(s)) ds + K_{pi} (\dot{x}_m(t) - \dot{x}_s(t)) \quad (1.4)$$

$$F_m(t) = -F_s(t)$$

This allows us to define a power flow function, (1.5), for the simple case of a 2 port block shown in Figure 1.5. This block is said to be passive if the energy exiting the block is less than or equal to the energy that enters it. It is of great interest here to note that two passive systems connected in cascade will result in a passive system.

$$P(t) = \dot{x}_l^T(t) F_l(t) - \dot{x}_r^T(t) F_r(t) \quad (1.5)$$

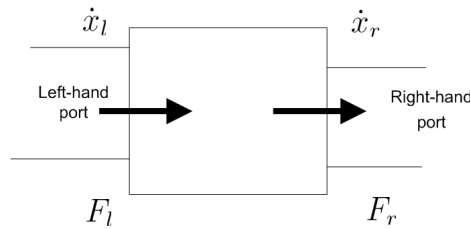


Figure 1.5: 2 port power flow block

Breaking down the bilateral operation structure into blocks that we will thereafter attempt to show passivity resembles Figure 1.6. Both manipulators are clearly passive. The operator and remote environment can be assumed passive without loss of generality. For example, the case where the slave



Figure 1.6: Block diagram of bilateral teleoperation

manipulator is in contact with a wall, the wall will only push back the robot as much as he is being pushed and, therefore, will not generate extra energy.

Only the passivity of the communication medium is left to be dealt with. In [4], the authors demonstrate that the communication medium was not stable under time delays, and that, even under very small delays (order of milliseconds). The power flow through the communication medium takes the form:

$$P = \dot{x}_m^T F_{human} - \dot{x}_s^T F_{environment} \quad (1.6)$$

Using a power flow equation of the form of (1.7) demonstrates that  $u_l/u_r$  adds energy and that  $v_l/v_r$  removes energy.

$$P = \frac{1}{2} u_l^T u_l - \frac{1}{2} v_l^T v_l + \frac{1}{2} u_r^T u_r - \frac{1}{2} v_r^T v_r \quad (1.7)$$

This leads to an input wave,  $u$ , and an output wave  $v$ . Equating the two power flow equations, (1.6) and (1.7), will give us transformation equations of the form (1.8) where we have replaced the general subscripts,  $l$  and  $r$  by  $m$  and  $s$  to represent the master and slave side of the communication medium.

$$\begin{aligned} u_m &= \frac{1}{\sqrt{2b}} (F_m + b\dot{x}_m) & v_m &= \frac{1}{\sqrt{2b}} (F_m - b\dot{x}_m) \\ u_s &= \frac{1}{\sqrt{2b}} (F_s - b\dot{x}_s) & v_s &= \frac{1}{\sqrt{2b}} (F_s + b\dot{x}_s) \end{aligned} \quad (1.8)$$

With these 4 transformation equations we can choose to transmit any combination of signals over the communication medium. Namely, we could put both manipulators in force control, or both manipulators in velocity control or a combination of either force or velocity control. The authors in [15] show that the first transformation block should receive  $\dot{x}_m$  and  $v_l$  and output  $u_l$  and  $F_{md}$ , as illustrated in Figure 1.7. The slave manipulator finds itself in velocity control, tracks the position of the master, and the master is in force control mode, reflects the forces applied by the slave robot. This combination is exactly what we defined as a bilateral teleoperator earlier.

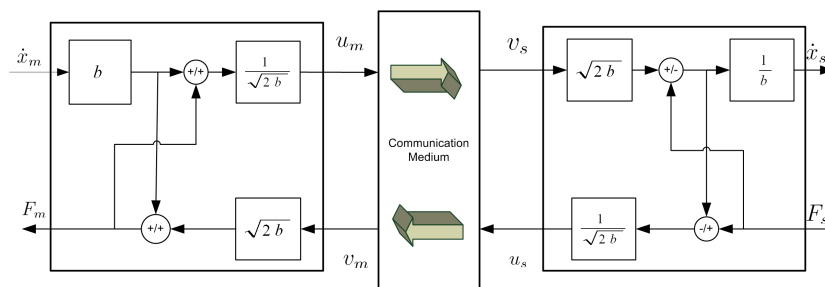


Figure 1.7: Wave transformation for PEB

Transmitting and receiving the wave variables will ensure that no energy is generated within the communication medium, even in the presence of time delays. The length of the time delay will only have an impact on the performance of the teleoperator system. For longer delays, the system is less transparent. In the case of a  $n$ -DOF system the wave equations must be modified such that the impedance matching takes the form of matrices as show in Table 1.2.

1-DOF	$n$ -DOF
$u_m = \frac{1}{\sqrt{2b}}(F_m + b\dot{x}_m)$	$\mathbf{u}_m = A\mathbf{F}_m + B\dot{\mathbf{x}}$
$v_m = \frac{1}{\sqrt{2b}}(F_m - b\dot{x}_m)$	$\mathbf{v}_m = C\mathbf{F}_m - D\dot{\mathbf{x}}$

Table 1.2: Comparison between 1-DOF to  $n$ -DOF

In Table 1.2,  $A, B, C$  and  $D$  are  $n \times n$  matrices. We then need to find the appropriate values that will guarantee stability and maximize the transparency of the communication medium. In [16] the procedure is proposed to obtain the family of all possible matrices in order to guarantee stability. These procedure, shown in Table 1.3, leave the engineer with three control variables,  $A, Q$  and  $S$ .  $A$  represents the damping imposed on the wave,  $Q$  rotates the wave and  $S$  is a trade-off between settling time and steady state error.

Wave variables do present some limitations during implementations, mainly a certain maximum impedance matching,  $b \leq b_{max}$ . Filtering should be done

- 
1.  $A$  is nonsingular
  2.  $S$  is any  $n \times n$  skew-symmetric matrix
  3.  $Q$  is any  $n \times n$  orthogonal matrix
  4.  $B = 0.5(I + S)A^{-T}$
  5.  $C = QA$
  6.  $D = 0.5(I - S)A^{-T}$
- 

Table 1.3: Procedures proposed for n-DOF wave transmission

in the wave space with a cut-off frequency of  $f_{cut}$  [17].

### 1.5.6 Wave variables in Hamiltonian dynamics

The authors of [18, 19, 20, 21] obtained similar wave variables using the equations of motion in Hamiltonian dynamics, (1.9) [22], shown in (1.10) where  $(f, e)$  denote the generalized flow and effort respectively and  $s_+, s_-$  represented the forward moving wave and backward moving wave.

$$\begin{cases} \dot{x} = (J(x) - R(x))\frac{\partial H(x)}{\partial x} + G(x)u \\ y = G^T(x)\frac{\partial H(x)}{\partial x} \end{cases} \quad (1.9)$$

$$\begin{aligned} s_+ &= \frac{N-1}{\sqrt{2}}(e + Zf) \\ s_- &= \frac{N-1}{\sqrt{2}}(e - Zf) \end{aligned} \quad (1.10)$$

The equation of motion shown in (1.9) has  $x$  as state vector,  $H$  as Hamiltonian energy function,  $J(x)$  as the skew-symmetric Poisson tensor,  $R(x)$  as the symmetric positive semi-definite, dissipation tensor,  $u$  as the input and  $y$  as the output, [22].

The wave transformation matrix in (1.10) can be augmented to include impedance matching using an Intrinsically Passive Controller (IPC), [18], and adapt to the scenario where time-varying time delays are present, [21].

### 1.5.7 Wave variables over the internet

The internet is seen as the ultimate frontier as means for bilateral teleoperation communication medium since it is widely available, affordable and does not require extra infrastructure as dedicated communication lines do. On the

other hand, the internet does provide added difficulties to the control task by providing a time-varying time delay that can reach lengths of infinity, such as the case of packet loss.

### Wave integral transmission

Wave transmission uses the generalized velocity and force to compute the waves, although the controller seeks to minimize the position error. A more intuitive method should include the position directly into the wave calculations in order to minimize the error. The authors in [23] proposed an interesting scheme to that end. Instead of transmitting the wave variables, Figure 1.8a, they propose to transmit the wave variable integral and the integral of the squared wave, depicted in Figure 1.8b.

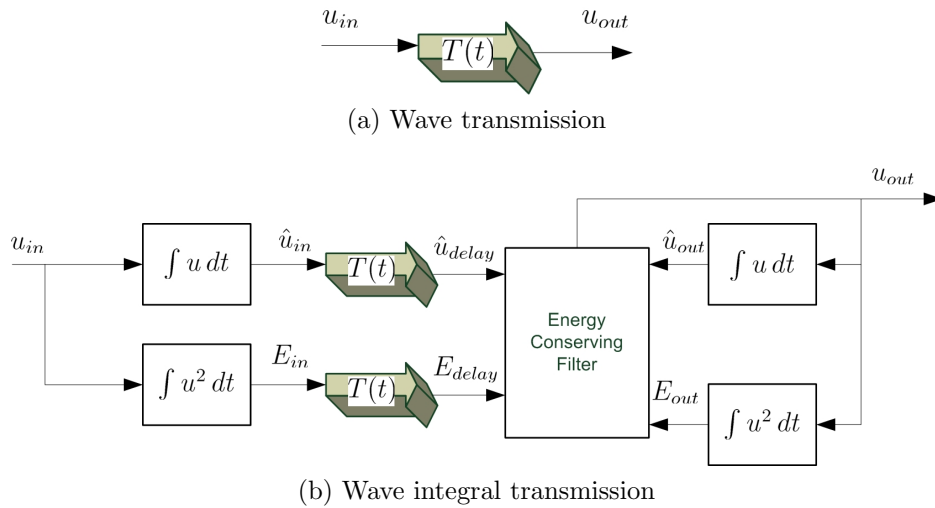


Figure 1.8: One way communication of wave variables contrasted

In Figure 1.8,  $u_{in}$  is the wave variable found through (1.8),  $T(t)$  represents the time-varying time delay and  $E_{in}$  the energy content of the wave. The variable  $\hat{u}_{in}$  now incorporates the position information and not the velocity, (1.11), with  $p$  representing the momentum.

$$\hat{u}_{in}(t) = \int_0^t u_{in}(\tau) d\tau = \int_0^t \frac{b\dot{x}_m(\tau) + F(\tau)}{\sqrt{2b}} d\tau = \frac{bx + p}{\sqrt{2b}} \quad (1.11)$$

The energy conserving filter in Figure 1.8b, seeks  $\hat{u}_{out}(t) \rightarrow \hat{u}_{delay}(t)$  while using only the available energy  $E_{out}(t) \leq E_{delay}(t)$ .

### Communication signal management

A different approach guaranteeing stability in the presence of time varying time delay consists of adding a Communication Signal Management block between the received wave variable,  $u_{in}$ , and the wave variable transformation block, Figure 1.9.

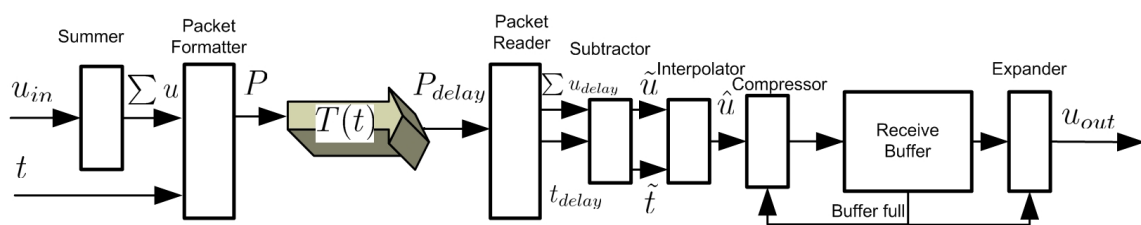


Figure 1.9: CMM diagram for one direction of the data path

In Figure 1.9,  $u_{in}$  is the wave variable found through (1.8). The summer adds all past wave variable into  $\sum u$ . A packet is then formed composed of both the sum of all previous  $u$  and the current time indicator,  $t$ . This packet is sent through the internet using the User Datagram Protocol (UDP) since it provides low time delays but does not provide hand-shaking hence dropped packets will never reach their destination (contrary to TCP/IP that uses hand-shaking to guaranty packet reception at the cost of additional time delays). Once the packet has reached the other side of the communication medium it is then decomposed into its delayed counterparts,  $\sum u_{delay}$  and  $t_{delay}$ . The subtractor finds the difference between the old value and the new value, (1.12) and (1.13), in order to solve for the current wave  ${}_i\tilde{u}$  and time  ${}_i\tilde{t}$ , where  $i$  denotes the discrete time.

$${}_i\tilde{u} = \sum_{i=0}^{\tau} {}_i u_{delay} - \sum_{i=0}^{\tau-1} {}_i u_{delay} \quad (1.12)$$

$${}_i\tilde{t} = {}_i t_{delay} - {}_{i-1} t_{delay} \quad (1.13)$$

Next, the interpolator checks to see if a packet was lost. If the proper order of packets was achieved, then  $\tilde{t} = 1$ , else if the packet is lost  $\tilde{t} > 1$ . The interpolator uses (1.14) to produces  $\tilde{t}$  values of  $\hat{u}$ .

$$\hat{u} = \frac{\tilde{u}}{\tilde{t}} \quad (1.14)$$

As we can see in the ideal case  $\tilde{t} = 1$ , we simply forward the value but if a packet was lost, then we find the average value across that lost packet(s) and produce  $\tilde{t}$  values of  $\hat{u}$ . This method is known to maintain passivity, [24]. The last part of the control scheme consists of maintaining a smooth  $u_{out}$  even if a burst of packets is received or no packets are received for some time. First we have the buffer which stores values of  $u_{out}$  ready to be outputted. They will leave the buffer in a first-in-first-out (FIFO) matter. The buffer keeps track of how full it currently is and informs the compressor and expander. If the buffer is close to being full then the compressor will start compressing the incoming  ${}_{1\dots n}\hat{u}$  signal following (1.15).

$$\bar{u} = sgn\left(\sum_{i=1}^n {}_i\hat{u}\right) \sqrt{\sum_{i=1}^n {}_i\hat{u}^2} \quad (1.15)$$

Dual to the compressor is the expander, which stretches the signal if the buffer is mainly empty. It will expand a sample  $u$  into  $n$  samples  ${}_1\bar{u} \cdots {}_n\bar{u}$ , with each sample defined by (1.16).

$${}_i\hat{u} = sgn(u) \sqrt{\frac{u^2}{n}} \quad (1.16)$$

The compressor and expander may have a discrete ON/OFF or they can use a membership function that would be used to smooth out the amount of compressing and expanding in order to provide the smoothest possible response while maintaining a steady flow of data at  $u_{out}$ .



## 1.6 Outline

Our main research is to compare different control methods for Bilateral Teleoperation. For our implementations a six degree of freedom (DOF) haptic device, based on the twin pantograph architecture, is used as a master manipulator, while the slave manipulator is a 6 DOF serial robot manipulator, the A465 Robot by CRS Robotics.

This thesis is organized as follows. In Chapter 2, the forward kinematics, inverse kinematics and Jacobian calculation for the CRS manipulator are derived while the forward kinematics and Jacobian calculation of the haptic device are shown in Chapter 3. In Chapter 4, the different control schemes examined are expended and experimental results follow in Chapter 5 for the Kinesthetic Force Based controller, Position Error Based controller and wave variables. This is followed by conclusions and proposals for future work in Chapter 6. Appendix A gives some explanation of the software used for implementations and Appendix B describes some hardware modification done by the author.

# Chapter 2

## Description of the Slave Manipulator

### 2.1 Introduction

We seek to give a detailed explanation of the forward and inverse kinematic problems for our serial manipulator, the CRS A645, shown in Figure 2.1. We use the Denavit-Hartenberg (DH) method and kinematic decoupling [12]. First, the DH parameters are found and the forward kinematics problem is solved. Second, using the forward kinematics results, we obtain the manipulator Jacobian. Thereafter, we explain how kinematic decoupling helps to solve the inverse kinematics for the 8 possible solutions. The 8 solutions are examined in details and visually represented. Singularity configurations are also discussed.



Figure 2.1: The CRS manipulator

## 2.2 Forward Kinematics of the CRS robot

The forward kinematics are derived here in order to calculate the end-effector location and orientation given its joint angles. More importantly, we can use this information for the calculation of the Jacobian, which, for bilateral teleoperation, will be needed. This is done through the powerful method known as the DH-method, which is directly applicable for serial robots, such as the CRS robot.

The three orthogonal axes will be represented by ‘ $i$ ’, ‘ $j$ ’ and ‘ $k$ ’ respectively. As described in [12], we first need to assign  $k_n$  axes to correspond with the  $n^{\text{th}}$  revolute joint’s axis of rotation,  $k_0$  through  $k_5$  on Figure 2.2. Following that, we need to establish the  $i_n$  axes,  $n = 0, 1, 2, 3, 4, 5$ , along the common normal between the  $k_{n-1}$  and the  $k_n$  axis going through the origin  $O_n$  or in the direction normal to the  $k_{n-1} - k_n$  plane, if they intersect. Thereafter the ‘ $j$ ’ axes are selected according to the right hand rule. From these axes, illustrated on Figure 2.2, we can get the DH parameters shown in Table 2.1, with units in centimetres.

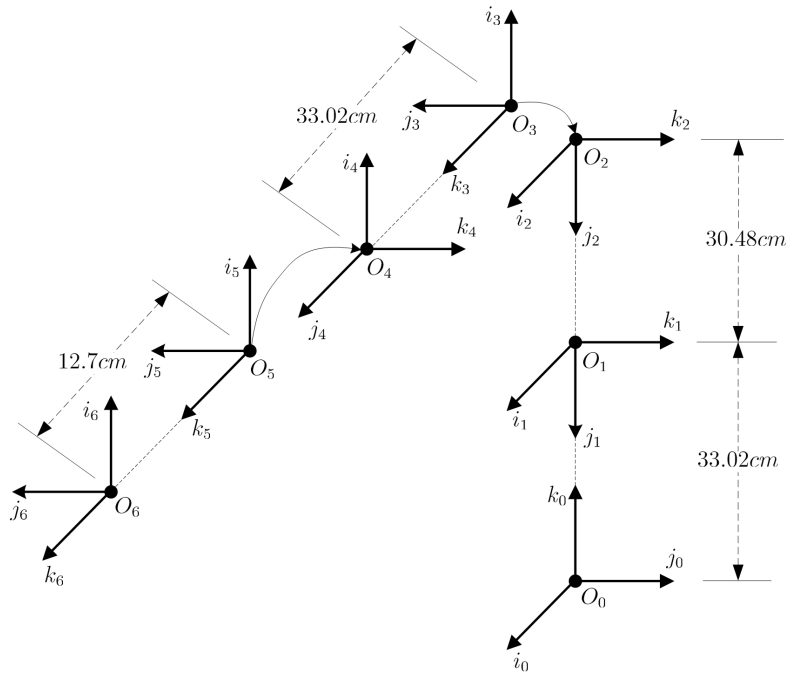


Figure 2.2: Chosen Co-ordinate Frames

link <sub><i>i</i></sub>	DH parameters			
	<i>d<sub>i</sub></i>	<i>a<sub>i</sub></i>	<i>α<sub>i</sub></i>	<i>θ<sub>i</sub></i>
1	33.02	0	$-\pi/2$	$\theta_1$
2	0	30.48	0	$\theta_2 - \frac{\pi}{2}$
3	0	0	$-\pi/2$	$\theta_3$
4	33.02	0	$\pi/2$	$\theta_4$
5	0	0	$-\pi/2$	$\theta_5$
6	12.7	0	0	$\theta_6$

Table 2.1: DH parameters for the CRS manipulator

To find the position and the orientation of the end-effector with respect to our defined origin we find  ${}^0_6T$  as shown in equation (2.1) where we have followed [12] and described  $A_i$  as:

$$A_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \cos(\alpha_i) & \sin(\theta_i) \sin(\alpha_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \cos(\alpha_i) & -\cos(\theta_i) \sin(\alpha_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The parameter  $A_i$  represents a rotation around the axis ‘*k*’ by  $\theta$  followed by a translation on ‘*k*’ by  $d$  centimetres followed by a translation on ‘*i*’ by  $a$  centimetres and finishes with a rotation around the ‘*i*’ axis by  $\alpha$ . Therefore  $a$  represents the length,  $d$  is the offset,  $\alpha$  is for the twist and finally  $\theta$  is the angle.

$${}^0_6T = A_1 A_2 A_3 A_4 A_5 A_6 \quad (2.1)$$

A homogenous transformation matrix,  ${}^a_bT$ , describes the transformation from frame  $b$  to frame  $a$ . It can be decomposed as a  $3 \times 3$  rotation matrix ( $R$  in (2.2)) which describes the orientation in  $SO(3)$  and a column vector ( $d$  in (2.2)) that represents the end-effector position in three dimensional space. The rotation matrix can be further decomposed in a per axis rotation, shown in (2.2) as  $r_i, r_j, r_k$ .

$${}^0_6T = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_i & r_j & r_k & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

Now we can use this transformation matrix to find the end-effector location in generalized world coordinates,  $x, y, z$ , given its joint angles by:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = d(q)$$

The orientation of the end-effector is given by the rotation matrix  $R$  from which we can obtain the Euler angles  $\phi, \theta$  and  $\psi$ .

## 2.3 Jacobian

The Jacobian relates the generalized velocities to the joint velocities.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = J(q)\dot{q} \quad \text{where} \quad \begin{aligned} \dot{q} &= [\dot{q}_1 \quad \dot{q}_2 \quad \cdots \quad \dot{q}_n]^T \\ v &= [\dot{x} \quad \dot{y} \quad \dot{z}]^T \\ \omega &= [\omega_x \quad \omega_y \quad \omega_z]^T \end{aligned}$$

$J$  is therefore a  $6 \times n$  matrix, or in our case a  $6 \times 6$  matrix since we have 6 articulated joints. We can separate  $J$  into two parts, first a linear velocity term ( $J_v$ ) and second an angular velocity term ( $J_\omega$ ).

$$J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \quad (2.3)$$

In the case of a revolute joint, [12, 25] demonstrate that these values are found through (2.4) and (2.5), where  $\hat{k}_{i-1}$  and  $O_{i-1}$  represent the first three rows of the third and fourth column of the transformation matrix  ${}^0_i T$  respectively while  $O_n$  represents the first three rows of the last column of the  ${}^0_6 T$  transformation matrix found in (2.1). The Jacobian is then the two matrices combined, (2.3).

$$J_{vi} = \hat{k}_{i-1} \times (O_n - O_{i-1}) \quad (2.4)$$

$$J_{\omega i} = \hat{k}_{i-1} \quad (2.5)$$

## 2.4 Inverse Kinematics of CRS robot

In this section, we would like to solve for the robot's joint angles given a desired position and orientation of the end-effector. To this end, we use the kinematics decoupling approach, [12]. The problem of inverse kinematics can be tricky for serial robots since there may be more than one joint combination that leads to the desired end-effector location and orientation. As we will see in the case of the CRS robot there is a total of 8 possible scenarios since 3 joints have 2 possible solutions hence  $2^3$  gives 8 solutions.

### 2.4.1 Kinematics decoupling

Solving the inverse kinematics can be quite difficult for manipulators with six joints. If a manipulator has the last three joints intersecting at a point, which is the case for the A645, then the problem can be divided into an inverse position kinematics problem and an inverse orientation kinematics problem, [12]. The intersection of the last three joints is defined as wrist center,  $P_c$ , on Figure 2.3. A desired wrist center can be calculated given a desired position and orientation. Once the desired wrist center is known, the inverse position kinematics solves for the first three joints such that the desired wrist center is obtained, solving the anthropomorphic manipulator for the A645. Thereafter inverse orientation kinematics solves the last three joints in order to achieve the desired position and orientation of the end-effector, solving the spherical wrist for the A645.

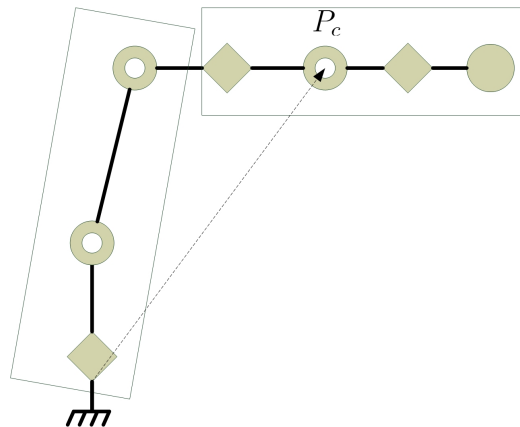


Figure 2.3: Arm and Wrist decoupling

### 2.4.2 Anthropomorphic Manipulator (RRR)

The anthropomorphic manipulator consists of the first three joints of the A645 robot. In order to solve for the wrist center,  $P_c$ , we further decompose the problem in two parts: a waist and a planar manipulator. First the first joint angle,  $\theta_1$ , is determined with the triangle shown in Figure 2.4 through the following equation.

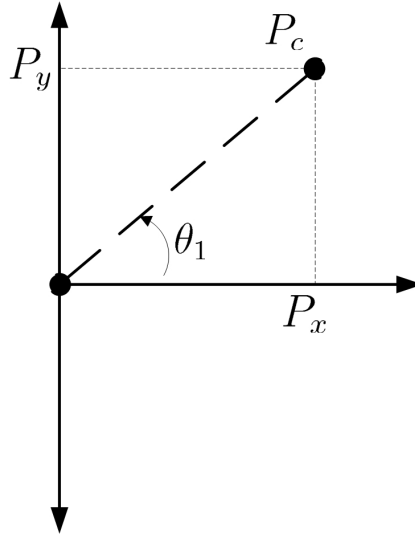


Figure 2.4: Solving for  $\theta_1$

$$\theta_1 = \text{atan2}(P_y, P_x)$$

Where  $\text{atan2}$  is the two argument arctangent. We should note that there may be a singularity if  $P_y = 0$  and  $P_x = 0$ , see the *Singularity* subsection for details. Also, note here that the robot may be facing the work-area or may be backed to it. because of this, we also have a configuration where:

$$\theta_1 = \pi + \text{atan2}(P_y, P_x)$$

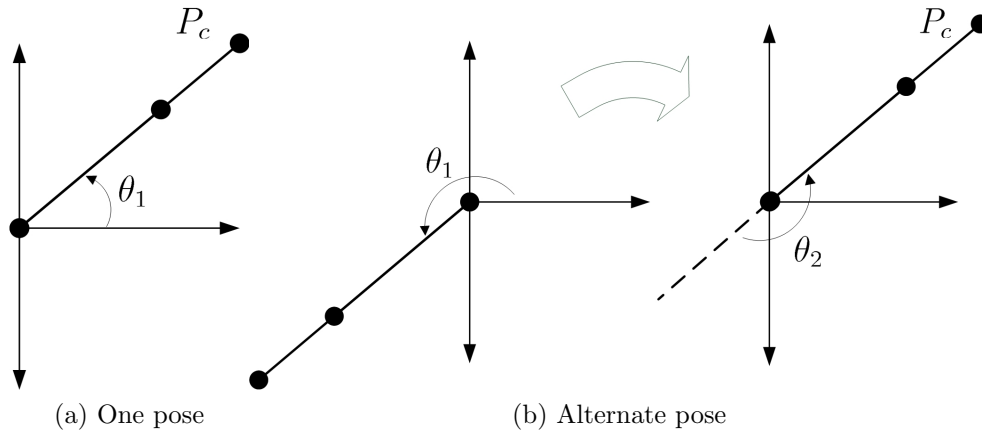


Figure 2.5: Two possible torso configuration

As seen in Figure 2.5b, this alternate possibility will yield different solutions for the following two joints,  $\theta_2$  and  $\theta_3$ . Once  $\theta_1$  is found we can isolate  $\theta_2, \theta_3$  as a planar manipulator and solve for them. We now have 2 possible scenarios that will yield the same end result; elbow up and elbow down.



## Facing with Elbow Up

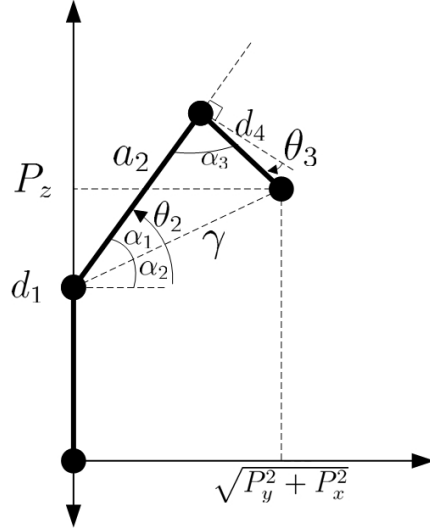


Figure 2.6: Solving with elbow up configuration

Based on Figure 2.6, the following equations, (2.6) through (2.11), are used to find the planar manipulator joint variables  $\theta_2$  and  $\theta_3$  for the case where the waist is facing forward and the elbow pointing upwards. The variables  $d_1, a_2, d_4$  are the DH parameters from Table 2.1 and  $P_y, P_x, P_z$  represent the cartesian components of  $P_c$ .

$$\gamma = \sqrt{P_y^2 + P_x^2 + (P_z - d_1)^2} \quad (2.6)$$

$$\alpha_1 = \text{atan2} \left( (P_z - d_1), \sqrt{P_y^2 + P_x^2} \right) \quad (2.7)$$

$$\alpha_2 = \arccos \left( \frac{\gamma^2 + a_2^2 - d_4^2}{2\gamma a_2} \right) \quad (2.8)$$

$$\alpha_3 = \arccos \left( \frac{a_2^2 + d_4^2 - \gamma^2}{2a_2 d_4} \right) \quad (2.9)$$

$$\theta_2 = \alpha_1 + \alpha_2 \quad (2.10)$$

$$\theta_3 = -\pi/2 + \alpha_3 \quad (2.11)$$

## Facing with Elbow Down

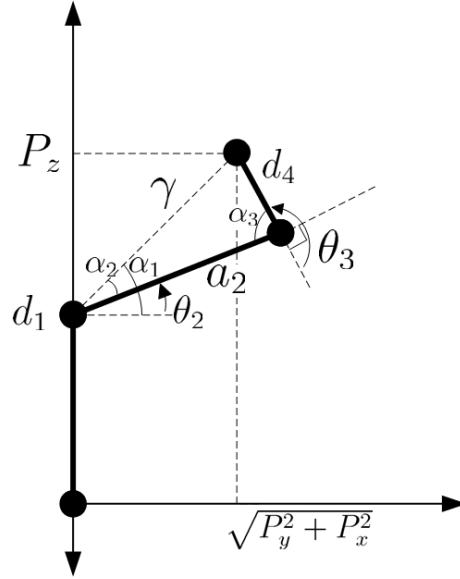


Figure 2.7: Solving with elbow down configuration

Based on Figure 2.7, the following equations, (2.12) through (2.17), are used to find the planar manipulator joint variables  $\theta_2$  and  $\theta_3$  for the case where the waist is facing forward and the elbow pointing downwards.

$$\gamma = \sqrt{P_y^2 + P_x^2 + (P_z - d_1)^2} \quad (2.12)$$

$$\alpha_1 = \text{atan2} \left( \sqrt{P_y^2 + P_x^2}, (P_z - d_1) \right) \quad (2.13)$$

$$\alpha_2 = \arccos \left( \frac{\gamma^2 + a_2^2 - d_4^2}{2\gamma a_2} \right) \quad (2.14)$$

$$\alpha_3 = \arccos \left( \frac{a_2^2 + d_4^2 - \gamma^2}{2a_2 d_4} \right) \quad (2.15)$$

$$\theta_2 = \alpha_1 - \alpha_2 \quad (2.16)$$

$$\theta_3 = 3\pi/2 - \alpha_3 \quad (2.17)$$

## Back with Elbow Up

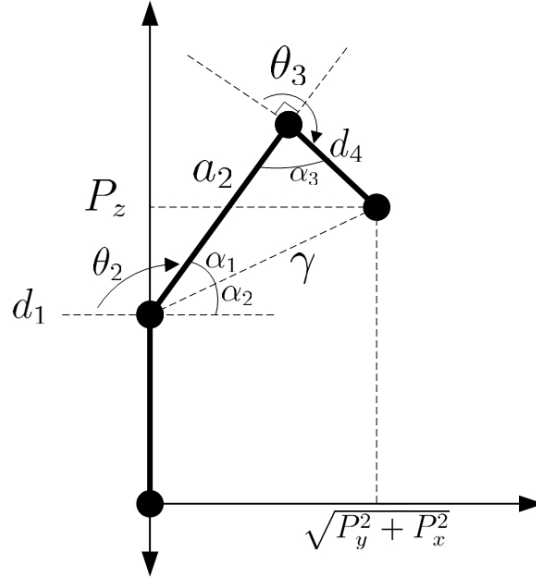


Figure 2.8: Solving with elbow up configuration

Based on Figure 2.8, the following equations, (2.18) through (2.23), are used to find the planar manipulator joint variables  $\theta_2$  and  $\theta_3$  for the case where the waist is facing backwards and the elbow pointing upwards.

$$\gamma = \sqrt{P_y^2 + P_x^2 + (P_z - d_1)^2} \quad (2.18)$$

$$\alpha_1 = \text{atan2} \left( (P_z - d_1), \sqrt{P_y^2 + P_x^2} \right) \quad (2.19)$$

$$\alpha_2 = \arccos \left( \frac{\gamma^2 + a_2^2 - d_4^2}{2\gamma a_2} \right) \quad (2.20)$$

$$\alpha_3 = \arccos \left( \frac{a_2^2 + d_4^2 - \gamma^2}{2a_2 d_4} \right) \quad (2.21)$$

$$\theta_2 = \pi - \alpha_1 - \alpha_2 \quad (2.22)$$

$$\theta_3 = 3\pi/2 - \alpha_3 \quad (2.23)$$

## Back with Elbow Down

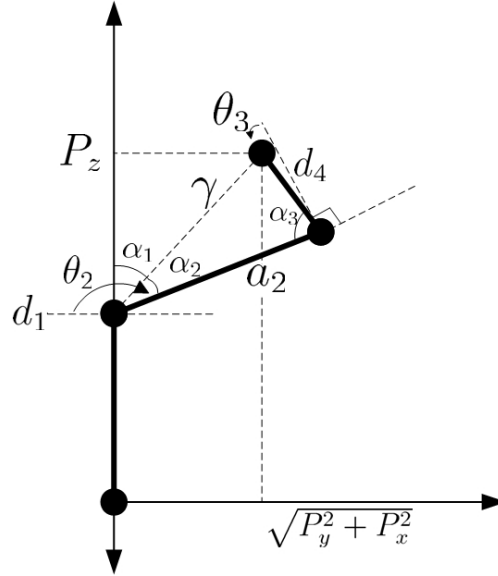


Figure 2.9: Solving with elbow down configuration

Based on Figure 2.9, the following equations, (2.24) through (2.29), are used to find the planar manipulator joint variables  $\theta_2$  and  $\theta_3$  for the case where the waist is facing backward and the elbow pointing downwards.

$$\gamma = \sqrt{P_y^2 + P_x^2 + (P_z - d_1)^2} \quad (2.24)$$

$$\alpha_1 = \text{atan2} \left( \sqrt{P_y^2 + P_x^2}, (P_z - d_1) \right) \quad (2.25)$$

$$\alpha_2 = \arccos \left( \frac{\gamma^2 + a_2^2 - d_4^2}{2\gamma a_2} \right) \quad (2.26)$$

$$\alpha_3 = \arccos \left( \frac{a_2^2 + d_4^2 - \gamma^2}{2a_2 d_4} \right) \quad (2.27)$$

$$\theta_2 = \pi - \alpha_1 - \alpha_2 \quad (2.28)$$

$$\theta_3 = -\pi/2 + \alpha_3 \quad (2.29)$$

### 2.4.3 Spherical Wrist

The solution of the spherical wrist is found by equating 2 matrices, (2.31). The first one  ${}^3_6R$  is the forward kinematics of a spherical wrist, (2.30), and the other,  $U$ , is the multiplication:

$$U = {}^3_0RR_d$$

We can find  ${}^3_0R$  by performing the forward kinematics using  $\theta_1, \theta_2, \theta_3$  and  $R_d$  is provided by the user. The shorthand notations  $c_i$  and  $s_i$  are used to represent  $\cos(\theta_i)$  and  $\sin(\theta_i)$  respectively.

$${}^3_6R = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 \\ -s_5c_6 & s_5s_6 & c_5 \end{bmatrix} \quad (2.30)$$

$${}^3_6R = U = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{bmatrix} \quad (2.31)$$

Assuming we don't have a singularity ( $U_{33} \neq \pm 1$ ) we can see that:

$$c_5 = U_{33} \quad \text{therefore} \quad s_5 = \pm \sqrt{1 - U_{33}^2}$$

and we have two possible configurations:

$$\begin{aligned} \theta_5 &= \text{atan2}\left(+\sqrt{1 - U_{33}^2}, U_{33}\right) \\ \theta_5 &= \text{atan2}\left(-\sqrt{1 - U_{33}^2}, U_{33}\right) \end{aligned} \quad (2.32)$$

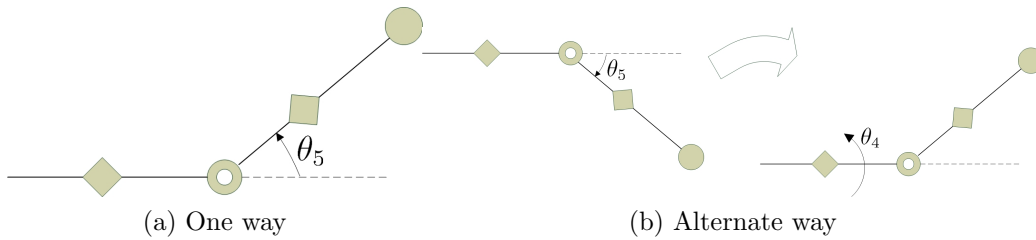


Figure 2.10: Two possible wrist configurations

If we select the case where  $\theta_5$  is positive, Figure 2.10a applies.

$$\frac{U_{23}}{U_{13}} = \frac{s_5s_4}{s_5c_4} = \frac{s_4}{c_4} = \tan(\theta_4)$$

and

$$\frac{U_{32}}{U_{31}} = \frac{s_6}{-c_6} = \tan(\theta_6)$$

will yield

$$\theta_4 = \text{atan2}(U_{23}, U_{13}) \quad (2.33)$$

$$\theta_6 = \text{atan2}(U_{32}, -U_{31}) \quad (2.34)$$

or if we select  $\theta_5 < 0$  we can get the following:

$$\theta_4 = \text{atan2}(-U_{23}, -U_{13}) \quad (2.35)$$

$$\theta_6 = \text{atan2}(-U_{32}, U_{31}) \quad (2.36)$$

We can see in Figure 2.10a the case where  $\theta_5 > 0$  and in Figure 2.10b the case where  $\theta_5 < 0$  with a rotation about  $\theta_4$  by  $180^\circ$  will get the same results, hence why we have 2 possible solutions that will yield the same result.

#### 2.4.4 Singularities

Singularities arise when the algorithm has an infinite amount of possible solutions to choose from in order to achieve the desired position or orientation. For the CRS robot this happens if  $P_y = 0$  and  $P_x = 0$  and if  $\theta_5 = 0$ . When  $P_y = 0$  and  $P_x = 0$  the end-effector find itself directly above its waist axis of rotation, as shown in Figure 2.11. Therefore  $\theta_1 = \theta_{1Old}$  in order to minimize movement and the other joint angles are calculated in the regular manner.

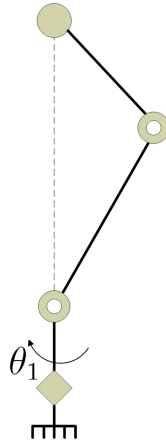


Figure 2.11: Waist in singularity position

The second singularity arises on the spherical wrist. Simply put if  $\theta_5 = 0^\circ$  then we can only solve for the sum of  $\theta_4$  and  $\theta_6$  as demonstrated in Figure 2.12. The actual joint angle that we are seeking for is impossible to know, in the case of Figure 2.12. By convention, the desired angle is split to both joints. Therefore  $\theta_4 = \theta_6$  is half the desired angle. Mathematically this is shown by the fact that;

$$c_5 = U_{33} = 1 \implies U_{13} = U_{23} = U_{31} = U_{32} = 0$$

and we are left with

$$U = \begin{bmatrix} c_{4+6} & -s_{4+6} & 0 \\ s_{4+6} & c_{4+6} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} U_{11} & U_{12} & 0 \\ U_{21} & U_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\theta_4 + \theta_6 = \text{atan2}(U_{21}, U_{11}) \quad (2.37)$$

$$= \text{atan2}(-U_{12}, U_{22}) \quad (2.38)$$

It should be noted that even if mathematically it is possible to have  $U_{33} = -1$ , it is not feasible in implementation since the 2 links at joint 5 would have to be inside or beside one another and that is not the case with the CRS A645 manipulator.

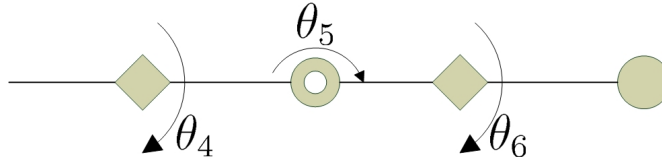


Figure 2.12: Spherical wrist in singularity position

# Chapter 3

## Description of the Master Manipulator

### 3.1 Introduction

The device used for the master manipulator is a twin pantograph manipulator, shown in Figure 3.1 and in Figure 3.2. It was originally designed at the University of British-Columbia [26]. The manipulator used was optimized and built by S. Ahmad at Lakehead University [27]. Given its parallel structure the tools used earlier to determine the forward kinematics or Jacobian cannot be used directly, hence the following description.

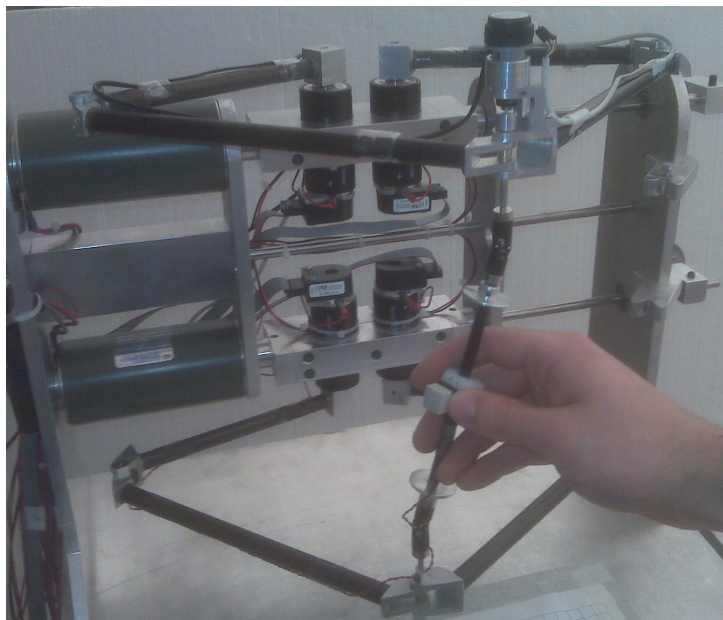


Figure 3.1: The twin pantograph manipulator



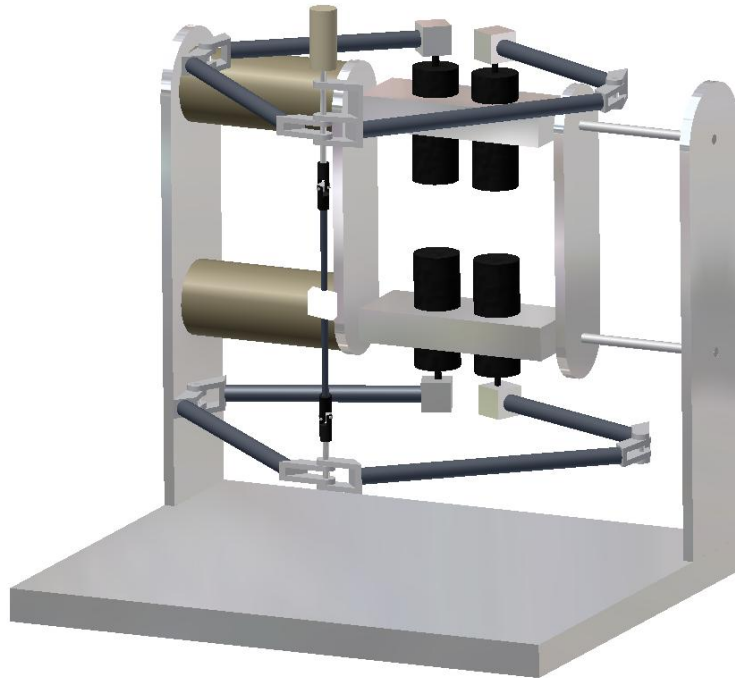


Figure 3.2: CAD drawing of the twin pantograph

## 3.2 Forward Kinematics of the haptic robot

The problem of determining the position and orientation of the end-effector using the angles of the active joint is called the forward kinematics. The twin pantograph manipulator consists of 7 active joints split-up as follows; 3 on the top five-bar linkage, 3 on the bottom five-bar linkage and one for the roll of the end-effector. First we will take each parallel structure, the pantograph, and analyze its pseudo-serial form shown in Figure 3.3.

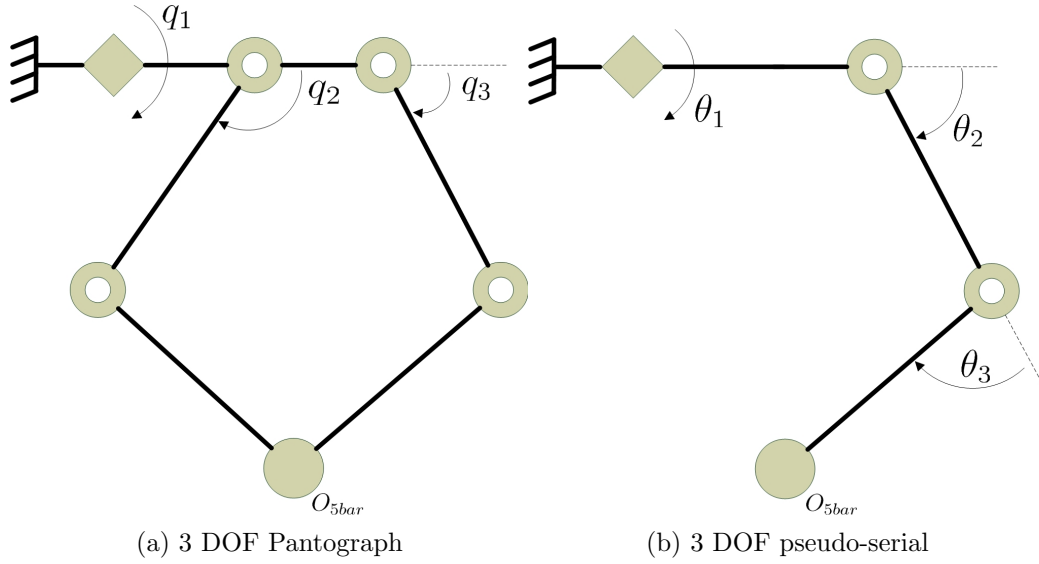


Figure 3.3: Serial equivalency of the 5-bar linkage, ie. the pantograph

Doing so will allow us to use the Denavit-Hartenberg (DH) method to find the transformation matrix up to the ‘tip’ of the 5-bar linkage, that we will define  $O_{5bar}$  and  $O_{5bar,b}$  for the top and bottom pseudo-serial manipulators respectively. From Figure 3.3, it is obvious that:

$$\theta_1 = q_1$$

$$\theta_2 = q_3$$

Hence only  $\theta_3$  is needed to be calculated using the two active joint five-bar linkage shown in Figure 3.4. To do so two vectors are needed;  $\hat{O}_{12}$ , which is the unit vector pointing from  $O_1$  to  $O_2$  and  $\hat{O}_{42}$ , the unit vector pointing from  $O_4$  to  $O_2$ , shown in Figure 3.4. We can find  $O_{12}$  and  $O_{42}$  as follows and thereafter find their corresponding unit vector:

$$O_{12} = \begin{bmatrix} l_2 \cos(q_3) \\ l_2 \sin(q_3) \end{bmatrix}$$

$$O_{42} = O_{12} + \begin{bmatrix} 2l_1 - l_5 \cos(q_2) \\ -l_5 \sin(q_2) \end{bmatrix}$$

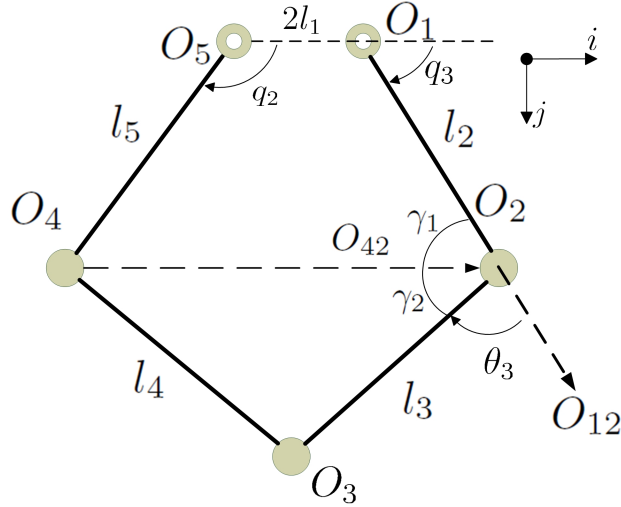


Figure 3.4: Origins on the pantograph

By adding and subtracting these unit vectors it allows us to build a triangle as shown in Figure 3.5. Using the inside angles of the 5-bar linkage ( $\gamma_1, \gamma_2$ ) it will be possible to determine  $\theta_3$ .  $\gamma_1$  is the angle between link 2 and  $O_{42}$ .  $\gamma_2$  is the angle between link 3 and  $O_{42}$ , shown in Figure 3.4.

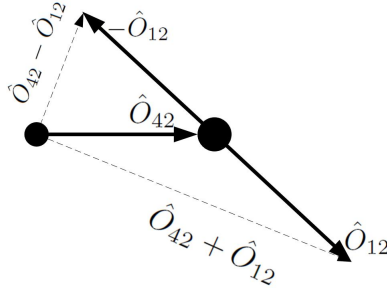
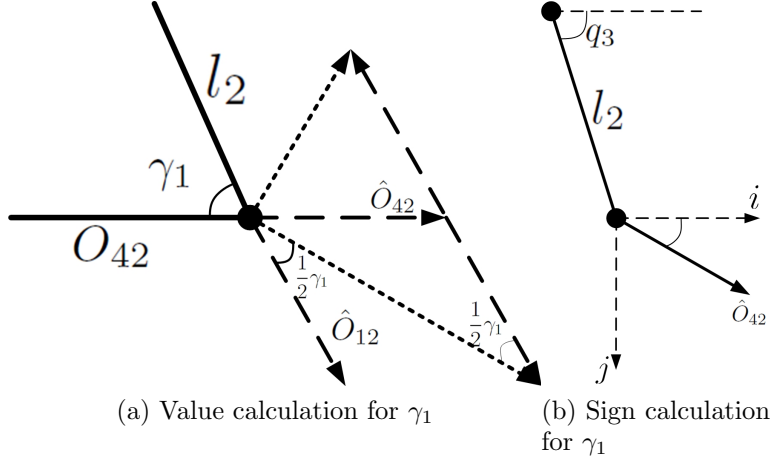


Figure 3.5: Unit vector triangle

We can see through Figure 3.6a that in order to determine  $\gamma_1$  we can double the angle found from the triangle, demonstrated in (3.1). As for the sign of  $\gamma_1$  we need to look at where this joint is, with  $q_3$ , and at what angle is it going through  $O_{42}$ , as seen in Figure 3.6b. This yields a method for finding the sign of  $\gamma_1$ , (3.2) with direction vector ' $i$ ' and ' $j$ ' shown in Figure 3.6b.

Figure 3.6: Visual interpretation of  $\gamma_1$ 

$$|\gamma_1| = 2 \tan^{-1} \frac{\|\hat{O}_{42} - \hat{O}_{12}\|}{\|\hat{O}_{42} + \hat{O}_{12}\|} \quad (3.1)$$

$$\text{sign}(\gamma_1) = q_3 - \tan^{-1} \left( \frac{j^T \hat{O}_{42}}{i^T \hat{O}_{42}} \right) \quad (3.2)$$

The angle  $\gamma_2$  can be found using the cosine rule, as stated by (3.3).

$$\gamma_2 = \cos^{-1} \left( \frac{l_3^2 + \|O_{42}\|^2 - l_4^2}{2l_3 \|O_{42}\|} \right) \quad (3.3)$$

Given that  $\theta_3$  is the angle between the vector representing link 2 going to link 3 it follows that:

$$\theta_3 = \pi - \gamma_1 - \gamma_2$$

Having values for  $\theta_1, \theta_2, \theta_3$  for the pseudo-serial manipulator we can find the transformation matrix from the base to the fifth origin,  ${}^b_5T$ . Please note that the base frame, indicated  $O_b$ , on Figure 3.7, was chosen such that it will match the base frame of the CRS manipulator.

$${}^b_5T = A_1 A_2 A_3 A_4 A_5 \quad (3.4)$$

This choice of origins yields DH-parameters shown in Table 3.1. Using the DH method, we get (3.4). We can now find the position and orientation of

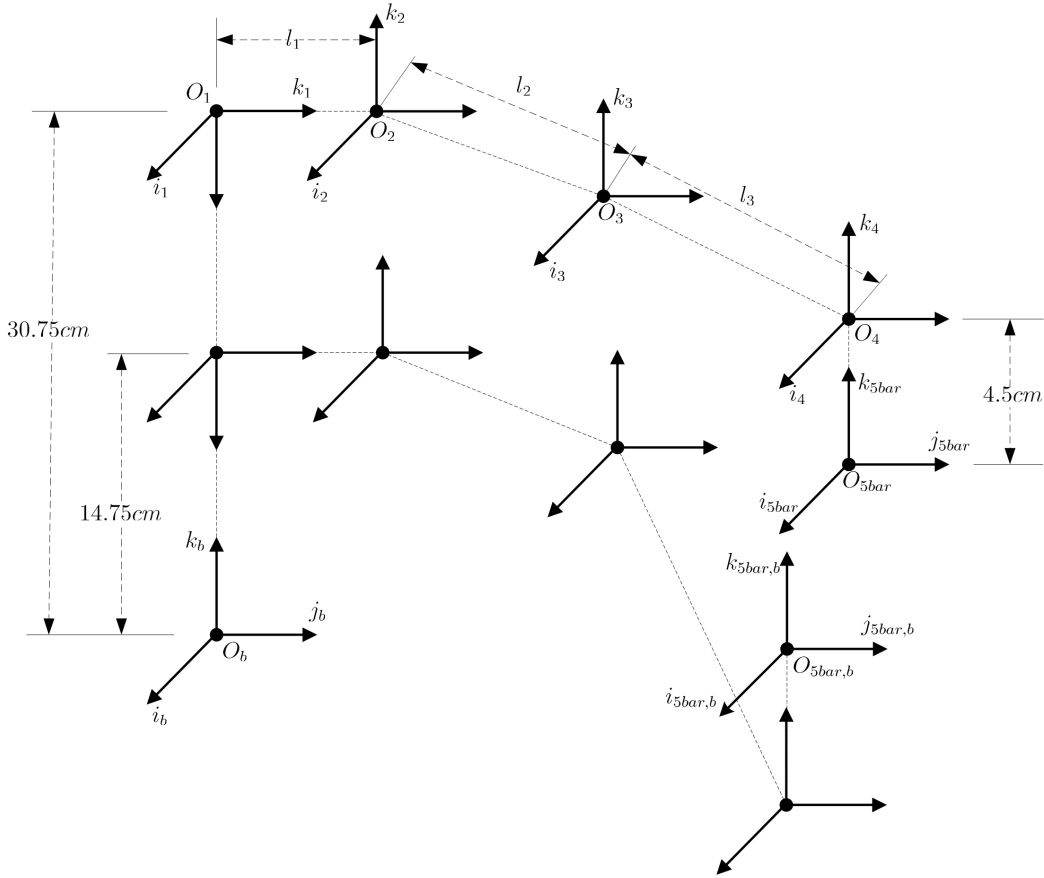


Figure 3.7: DH frame of reference for both pantographs

$O_{5bar}$  with respect to the base frame. Since errors can creep in with encoder errors and limitations of trigonometric functions, we will incorporate a safety feature here. Given how the top and bottom pantographs are physically connected with 16cm in between them we will check to see if the distance between the two calculated points is within 10% error.

$$14.4 < \|O_{5bar} - O_{5bar,b}\| < 17.6$$

In order for us to continue with the forward kinematics we will assume the frame for the end-effector has 'j'-axis pointing down, 'i'-axis stays the same as the base frame and 'k'-axis follows the right hand rule. This 'j'-axis can be found simply by taking the difference of the tips of the pantographs.

$$j_8 = \frac{O_{5bar,b} - O_{5bar}}{\|O_{5bar,b} - O_{5bar}\|}$$

Pantograph	DH parameters			
	$d_i$	$a_i$	$\alpha_i$	$\theta_i$
Bottom	14.75	0	$-\pi/2$	0
	$l_1$	0	$\pi/2$	0
	0	$l_2$	0	$\theta_{1b}$
	-4.5	$l_3$	0	$\theta_{2b}$
	0	0	0	$\theta_{3b}$
	4.5	0	0	0
Top	30.75	0	$-\pi/2$	0
	$l_1$	0	$\pi/2$	0
	0	$l_2$	0	$\theta_1$
	4.5	$l_3$	0	$\theta_2$
	0	0	0	$\theta_3$
	-4.5	0	0	0

Table 3.1: DH parameters for both pantographs

Knowing the position and orientation of  $O_{5bar}$  and ' $j_8$ ' we can find the rotation axes for the 2 DOF passive joint. The first axis, ' $k_6$ ', is parallel to ' $j_5$ '. Hence the second axis, ' $k_7$ ', needs to be perpendicular to both ' $j_8$ ' and ' $k_6$ ', (3.5) and Figure 3.8.

$$k_7 = j_8 \times k_6 \quad (3.5)$$

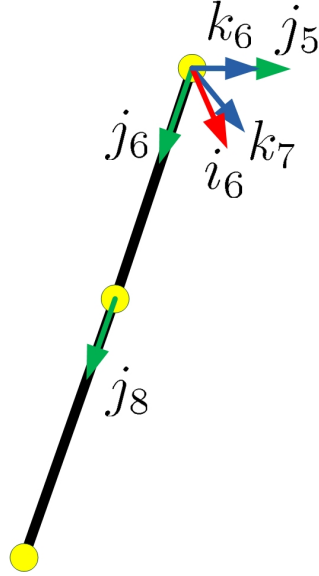


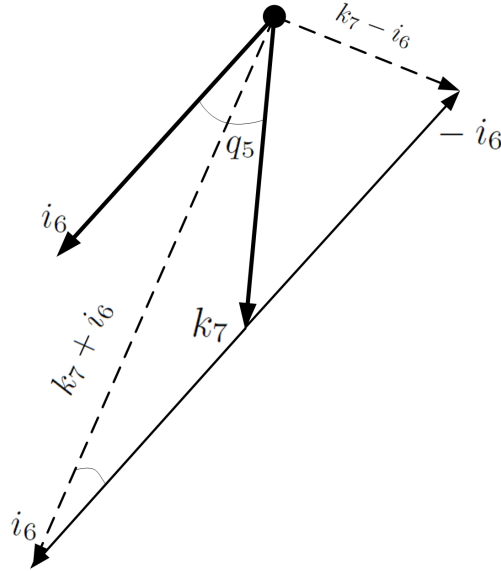
Figure 3.8: Finding the two last axes of rotation

Now that we have our rotation axes,  $k_5, k_6, k_7$ , we need to find the corresponding angles. Information about  $q_4$  is readily available since the corresponding joint is an active joint equipped with an encoder. The joint angles  $q_5$  and  $q_6$  need to be determined mathematically. Joint angle  $q_5$  is found by mapping  $k_7$  and  $i_6$  and determining the angle between the two unit vectors. This is done once again by doubling the angle between the difference and the addition of the two original unit vectors, see Figure 3.9, (3.6) and (3.7).

$$|q_5| = 2 \tan^{-1} \frac{\|k_7 - i_6\|}{\|k_7 + i_6\|} \quad (3.6)$$

$$\text{sign}(q_5) = \frac{k_7^T j_6}{|k_7^T j_6|} = \pm 1 \quad (3.7)$$

At this point, we have yet to find  $q_6$ . Before doing so we need more information, therefore we will continue the forward kinematics in order to know more information up to  $O_7$  using  $A_7, A_8, A_9$ . Thereafter  $q_6$  is found using the same method as  $q_5$  but using the difference between ' $j_7$ ' and ' $j_8$ ', shown in (3.8) and (3.9).

Figure 3.9: Determining  $q_5$ 

$$A_7 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_8 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_9 = \begin{bmatrix} -s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ c_5 & 0 & -s_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$|q_6| = 2 \tan^{-1} \frac{\|j_8 - j_7\|}{\|j_8 + j_7\|} \quad (3.8)$$

$$\text{sign}(q_6) = \frac{j_8^T i_7}{|j_8^T i_7|} = \pm 1 \quad (3.9)$$

Using this angle we can reach the end-effector,  $O_8$ , by using  $A_{10}$ .

$$A_{10} = \begin{bmatrix} c_6 & -s_6 & 0 & -l_8 s_6 \\ s_6 & c_6 & 0 & l_8 c_6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since we are analyzing the twin-pantograph structure as its pseudo-serial equivalent we will extract, (3.10), the spherical wrist angles from the final transformation matrix such that our final structure will essentially be a RRR



manipulator with a spherical wrist.

$$\text{From } {}^b_8T = \begin{bmatrix} {}^b_8R & O_8 \\ 0 & 1 \end{bmatrix} \text{ we can extract } [x \ y \ z \ \phi \ \theta \ \psi]^T \quad (3.10)$$

Where:  $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$  are the generalized coordinates of the end-effector with respect to the base frame

$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix}$  are the yaw, pitch and roll angle respectively that describe the rotation of the end-effector relative to the base frame

### 3.3 Jacobian

The Jacobian allows us to relate joint velocities with real world coordinate velocities. This is usually stated by (3.11). It is the same for parallel manipulators [28]. We can also use the Jacobian to relate applied joint torques and end-effector force/torques, (3.12).

$$\dot{q} = J(q)^{-1} \dot{x} \quad \text{where} \quad \begin{aligned} \dot{q} &= [\dot{q}_1 \ \dot{q}_2 \ \dot{q}_3 \ \dot{q}_{1b} \ \dot{q}_{2b} \ \dot{q}_{3b} \ \dot{q}_4]^T \\ \dot{x} &= [\dot{x} \ \dot{y} \ \dot{z} \ \omega_x \ \omega_y \ \omega_z]^T \end{aligned} \quad (3.11)$$

$$\tau = J(q)^T F \quad \text{where} \quad \begin{aligned} \tau &= [\tau_1 \ \tau_2 \ \tau_3 \ \tau_{1b} \ \tau_{2b} \ \tau_{3b} \ \tau_4]^T \\ F &= [F_x \ F_y \ F_z \ \tau_x \ \tau_y \ \tau_z]^T \end{aligned} \quad (3.12)$$

We will find this Jacobian by first splitting  $\dot{x}$ , the real world velocity of the end-effector, into two parts. Originally this Cartesian world velocity,  $\dot{x}$  will be referred to as  $\dot{O}_8$ . Once split, one part will be for the top pantograph and the other for the bottom pantograph, expressed in (3.13) and shown in Figure 3.10. The 8 denotes the end-effector origin and the 5 denotes the end-effector of the pseudo-serial 5-bar linkage.

$$\begin{pmatrix} \dot{O}_5 \\ \dot{O}_{5b} \\ \omega_8 \end{pmatrix} = T_1 \begin{pmatrix} \dot{O}_8 \\ \omega_8 \end{pmatrix} \quad (3.13)$$

Where  $T_1$  takes the form:

$$T_1 = \begin{bmatrix} I & l_8 S(j_8) \\ I & -l_8 S(j_8) \\ 0 & I \end{bmatrix} \quad (3.14)$$

and  $S(j_8)$  is the angular velocity tensor, which is a second rank skew-symmetric tensor;

$$S(A) = \begin{bmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{bmatrix}, \text{ with } A = \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix}$$

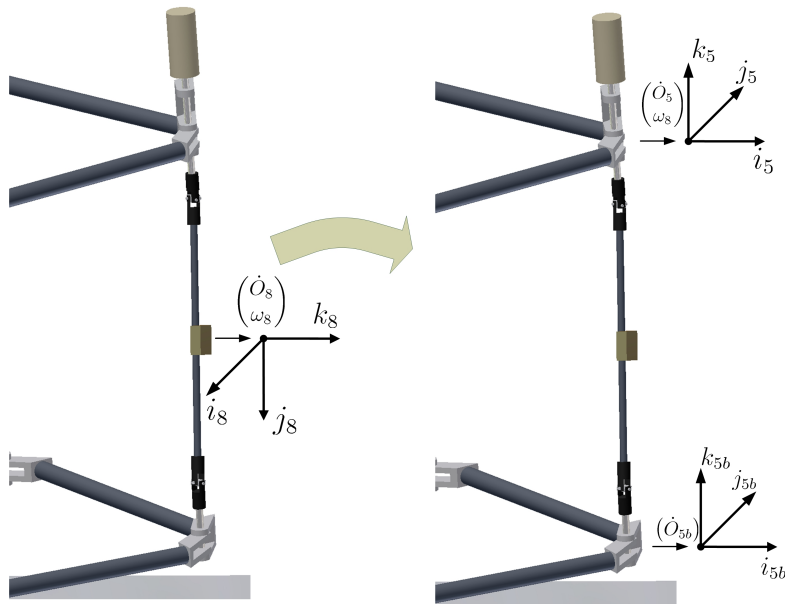


Figure 3.10: Distributing the end-effector velocity

The  $+l_8$  in (3.14) allows us to map the top pantograph, while the  $-l_8$  allows us to map the bottom pantograph. Since we already have made the 5-bar linkage into its pseudo-serial analogy and tools are readily available for Jacobian calculation we can therefore find the torque applied to the pseudo-manipulator, expressed with  $\dot{\theta}$  in (3.15).

$$\begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_{1b} \\ \dot{\theta}_{2b} \\ \dot{\theta}_{3b} \\ \omega_8 \end{pmatrix} = \underbrace{\begin{bmatrix} T_3 & 0 & 0 \\ 0 & T_{3b} & 0 \\ 0 & 0 & I \end{bmatrix}}_{T_2} \begin{pmatrix} \dot{O}_5 \\ \dot{O}_{5b} \\ \omega_8 \end{pmatrix} \quad (3.15)$$

With  $T_3$  being the inverse of the linear velocity Jacobian for a serial manipulator;

$$T_3 = [k_1 \times (O_4 - O_1) \quad k_2 \times (O_4 - O_2) \quad k_3 \times (O_4 - O_3)]^{-1}$$

where  $k_i$  represents the axis of rotation for the pseudo-serial joint  $i + 1$ ,  $O_1, O_2, O_3$  represent the origin of those joints. Since  $k_2 = k_3 = k_4$  and  $O_2$  can be equal to  $O_1$ , we have a simplified version:

$$T_3 = [k_1 \times (O_4 - O_1) \quad k_4 \times (O_4 - O_1) \quad k_4 \times (O_4 - O_3)]^{-1} \quad (3.16)$$

We will now look at the angular velocity term  $\omega_8$ . This term is currently defined with respect to the end-effector but we need to map it into the end-effector of the pseudo-serial manipulator of the top pantograph, defined as  ${}^5\omega_8$  shown in (3.17). We can see that in order to determine this value we take the angular velocity from the end-effector with respect to the origin ( ${}^0\omega_8$ ) and subtract from it the angular velocity found for the pseudo-serial manipulator from the origin to its end-effector ( ${}^0\omega_5$ ).

$${}^5\omega_8 = {}^0\omega_8 - {}^0\omega_5 = \omega_8 - T_5 \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix} \quad (3.17)$$

This is simply done by using known serial tools:

$$T_5 = J_\omega = [k_1 \quad k_4 \quad k_4]$$

It yields a trivial transformation of the form:

$$\begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_{1b} \\ \dot{\theta}_{2b} \\ \dot{\theta}_{3b} \\ {}^5\omega_8 \end{pmatrix} = \underbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -T_5 & 0 & I \end{bmatrix}}_{T_4} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_{1b} \\ \dot{\theta}_{2b} \\ \dot{\theta}_{3b} \\ \omega_8 \end{pmatrix}$$

We now need to transform the pseudo-serial joint rates into their equivalent parallel structure. We will deal with a single pantograph in this analysis since it is the same procedure for both. This transformation matrix,  $\delta$ , needs to satisfy (3.18).

$$\begin{aligned}\dot{\theta} &= \delta \dot{q} \\ \dot{x} &= \hat{J} \delta \dot{q}\end{aligned}\tag{3.18}$$

Where  $\hat{J}$  is the Jacobian that we found as  $T_3$  in (3.15). It was already stated that  $\theta_1 = q_1$  and  $\theta_2 = q_3$ , therefore we can assume that  $\delta$  will take the form stated below. The inverse of  $\delta$  takes the form shown in (3.19).

$$\begin{aligned}\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & V_1 & V_2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \\ \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{-V_2}{V_1} & V_1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}\end{aligned}\tag{3.19}$$

In order to solve for  $V_1$  and  $V_2$  in (3.19), we need to analyze the relationship that  $\dot{q}_2$  with  $\dot{q}_3$  have on  $\dot{\theta}_3$ . To do so, we will consider the case where  $\theta_1 = \dot{\theta}_1 = 0$ , in order to find a special case Jacobian  $\tilde{J}$ . We now have the following structure, Figure 3.11, and are able to determine some variables (3.20) through (3.24).

$$\tilde{O}_1 = \tilde{O}_0 + \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix}\tag{3.20}$$

$$\tilde{O}_2 = \tilde{O}_1 + \begin{bmatrix} l_2 \cos(q_3) \\ l_2 \sin(q_3) \\ 0 \end{bmatrix}\tag{3.21}$$

$$\tilde{O}_3 = \begin{bmatrix} {}^i O_{5bar} \\ \|(j j^T + k \hat{k}^T) O_{5bar}\| \\ 0 \end{bmatrix}\tag{3.22}$$

$$\tilde{O}_4 = \tilde{O}_5 + \begin{bmatrix} l_5 \cos(q_2) \\ l_5 \sin(q_2) \\ 0 \end{bmatrix}\tag{3.23}$$

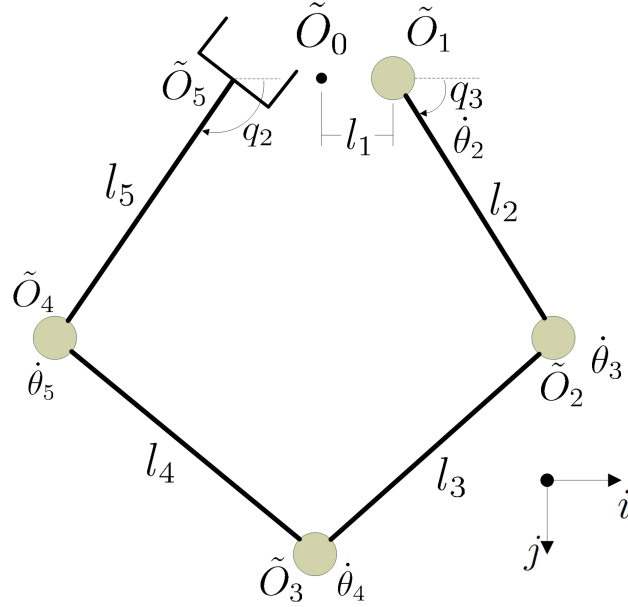


Figure 3.11: Two dimension serial manipulator considered

$$\tilde{O}_5 = \tilde{O}_0 + \begin{bmatrix} -l_1 \\ 0 \\ 0 \end{bmatrix} \quad (3.24)$$

Using serial tools, we can easily find the Jacobian of this structure to be;

$$\tilde{j} = \begin{bmatrix} k \times (\tilde{O}_5 - \tilde{O}_1) & k \times (\tilde{O}_5 - \tilde{O}_2) & k \times (\tilde{O}_5 - \tilde{O}_3) & k \times (\tilde{O}_5 - \tilde{O}_4) \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (3.25)$$

The pantograph is a variant of this architecture and, therefore, the end effector is actually fixed,  $\dot{\tilde{O}}_5 = 0$ , clearly  $\dot{\theta}_2 = \dot{q}_3$  and also  $\omega_5 = \dot{q}_2$ . Expanding and simplifying will yield:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} 0 & -j^T(\tilde{O}_5 - \tilde{O}_2) & -j^T(\tilde{O}_5 - \tilde{O}_3) & -j^T(\tilde{O}_5 - \tilde{O}_4) \\ 2l_1 & i^T(\tilde{O}_5 - \tilde{O}_2) & i^T(\tilde{O}_5 - \tilde{O}_3) & i^T(\tilde{O}_5 - \tilde{O}_4) \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_3 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \end{bmatrix}$$

This yields 3 linearly independent equations with 3 unknowns ( $\dot{\theta}_3, \dot{\theta}_4, \dot{\theta}_5$ ). We will make a change of variable to simplify the following math that will follow from (3.26) through (3.28).

$$\begin{aligned} A &\triangleq (\tilde{O}_5 - \tilde{O}_2) \\ B &\triangleq (\tilde{O}_5 - \tilde{O}_3) \\ C &\triangleq (\tilde{O}_5 - \tilde{O}_4) \end{aligned}$$

$$j^T A \dot{\theta}_3 + j^T B \dot{\theta}_4 + j^T C \dot{\theta}_5 = 0 \quad (3.26)$$

$$i^T A \dot{\theta}_3 + i^T B \dot{\theta}_4 + i^T C \dot{\theta}_5 = 2l_1 \dot{q}_3 \quad (3.27)$$

$$\dot{\theta}_3 + \dot{\theta}_4 + \dot{\theta}_5 = \dot{q}_2 - \dot{q}_3 \quad (3.28)$$

Taking these three equations, our goal becomes solving for  $\dot{\theta}_3$ . First we will multiply (3.26) by  $i^T(B - C)$ , (3.27) by  $j^T(B - C)$  and (3.28) by  $i^T C j^T(B - C) - j^T C i^T(B - C)$ .

$$\begin{aligned} j^T A i^T(B - C) \dot{\theta}_3 + j^T B i^T(B - C) \dot{\theta}_4 + \\ j^T C i^T(B - C) \dot{\theta}_5 = 0 \end{aligned} \quad (3.29)$$

$$i^T A j^T(B - C) \dot{\theta}_3 + i^T B j^T(B - C) \dot{\theta}_4 + i^T C j^T(B - C) \dot{\theta}_5 = 2l_1 j^T(B - C) \dot{q}_3 \quad (3.30)$$

$$\begin{aligned} [i^T C j^T(B - C) - j^T C i^T(B - C)] (\dot{\theta}_3 + \dot{\theta}_4 + \dot{\theta}_5) \\ = (i^T C j^T(B - C) - j^T C i^T(B - C)) (\dot{q}_2 - \dot{q}_3) \end{aligned} \quad (3.31)$$

Now we take equation (3.30) and subtract from it equation (3.29) followed by (3.31).

$$\begin{aligned} i^T (AB^T - AC^T - BA^T + CA^T - C(B - C)^T + (B - C)C^T) j \dot{\theta}_3 \\ = 2l_1 j^T(B - C) \dot{q}_3 - i^T (C(B - C)^T - (B - C)C^T) j (\dot{q}_3 - \dot{q}_2) \end{aligned}$$

Using the fact that  $i^T(AB^T - BA^T)j = k^T(A \times B)$ , one has

$$\begin{aligned} k^T((A \times B) + (C \times A) + ((B - C) \times C)) \dot{\theta}_3 = \\ 2l_1 j^T(B - C) \dot{q}_3 + k^T(C \times (B - C)) (\dot{q}_3 - \dot{q}_2) \end{aligned} \quad (3.32)$$

Moreover, it follows that:

$$\begin{aligned} (A \times B) + (C \times A) + ((B - C) \times C) &= (A \times B) - (A \times C) + ((B - C) \times C) \\ &= A \times (B - C) - (C \times (B - C)) \\ &= (A - C) \times (B - C) \end{aligned}$$

$$\therefore k^T(A - C) \times (B - C) \dot{\theta}_3 = 2l_1 j^T(B - C) \dot{q}_3 + k^T(C \times (B - C)) (\dot{q}_3 - \dot{q}_2)$$

$$\dot{\theta}_3 = \frac{2l_1 j^T (B - C) \dot{q}_3 + k^T (C \times (B - C)) (\dot{q}_3 - \dot{q}_2)}{k^T ((A - C) \times (B - C))} \quad (3.33)$$

Plugging back our original variables into (3.33) yields:

$$\dot{\theta}_3 = \frac{2l_1 j^T (\tilde{O}_4 - \tilde{O}_3) \dot{q}_3 + k^T \left( (\tilde{O}_5 - \tilde{O}_4) \times (\tilde{O}_4 - \tilde{O}_3) \right) (\dot{q}_3 - \dot{q}_2)}{k^T \left( (\tilde{O}_4 - \tilde{O}_2) \times (\tilde{O}_4 - \tilde{O}_3) \right)}$$

Since we are looking for two functions that will satisfy (3.19) we get:

$$V_1(q_2, q_3) = \frac{-k^T \left( (\tilde{O}_5 - \tilde{O}_4) \times (\tilde{O}_4 - \tilde{O}_3) \right)}{k^T \left( (\tilde{O}_4 - \tilde{O}_2) \times (\tilde{O}_4 - \tilde{O}_3) \right)} \quad (3.34)$$

$$V_2(q_2, q_3) = \frac{2l_1 j^T (\tilde{O}_4 - \tilde{O}_3)}{k^T \left( (\tilde{O}_4 - \tilde{O}_2) \times (\tilde{O}_4 - \tilde{O}_3) \right)} - V_1(q_2, q_3) \quad (3.35)$$

We will call the transformation matrix that allows us to go from pseudo-serial joint rates to parallel joint rates  $T_7$  and  $T_{7b}$  for the top and bottom pantograph respectively.

$$T_7 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{-V_2}{V_1} & V_1 \\ 0 & 1 & 0 \end{bmatrix} \quad (3.36)$$

The pseudo-serial joint rate,  $\dot{\theta}_4$ , is the same as the rate of change of the roll motor,  $q_7$ . The angular velocity term  ${}^5\omega_8$  now needs to be expressed with respect to the last three joints of the pseudo serial manipulator through the matrix  $T_8$  which is the inverse Jacobian of a spherical wrist. Using (3.37) and isolating for the joint  $\dot{\theta}_4$  will result in (3.38).

$$T_8 = [k_5 \quad k_6 \quad k_6]^{-1}$$

$$[\dot{\theta}_4 \quad \dot{\theta}_5 \quad \dot{\theta}_6]^T = [k_5 \quad k_6 \quad k_7]^{-1} {}^5\omega_8 \quad (3.37)$$

$$\dot{q}_7 = \dot{\theta}_4 = i^T T_8 {}^5\omega_8 \quad (3.38)$$

Having (3.36), by using (3.34) and (3.35), and (3.38), we are able to express all the pseudo-serial joints in terms of the parallel joints through  $T_6$ .

$$T_6 = \begin{bmatrix} T_7 & 0 & 0 \\ 0 & T_{7b} & 0 \\ 0 & 0 & i^T T_8 \end{bmatrix}$$

Ultimately, this leads to a method for solving the inverse Jacobian.

$$\begin{aligned} \dot{q} &= T_6 T_4 T_2 T_1 \dot{x} \\ \dot{q} &= J^{-1} \dot{x} \end{aligned} \tag{3.39}$$

Our goal of replicating a Cartesian force through motor torques cannot be directly solved through (3.39) since the relationship is (3.40). With (3.39) yielding a  $6 \times 7$  matrix, the left inverse needs to be found using the Moore-Penrose Matrix Inverse method which finds an inverse for non-square matrices, denoted with a  $+$  sign. This pseudo-inverse, which always provides the shortest length least squares solution, may not necessarily give a solution that satisfies all the original equations, [29]. The Moore-Penrose inverse is found as described in (3.41) where  $(H^T H)^{-1}$  is inverted using the Strassen-Newton inverse algorithm described in [30].

$$\tau = J(q)^T F \tag{3.40}$$

$$J^T = (J^{-1})^{+T} \quad \text{where } H^+ = (H^T H)^{-1} H^T \tag{3.41}$$



# Chapter 4

## Bilateral Teleoperation

### 4.1 System description

Bilateral teleoperation was implemented using the master and slave manipulators in the *Automatic Control Laboratory* at Lakehead University. Both manipulators have encoders on all motors and use MultiQ-3 I/O boards to interface with a computer. These boards, built by Quanser Consulting, provide us with 8 digital inputs, 8 digital outputs, 8 analog inputs, 8 analog outputs and 8 quadrature decoder/counters. The boards work in conjunction with many different softwares to implement our controllers in real time. First, a SIMULINK schematic needs to be built which includes acquisition blocks, our control program and output blocks. The control program is written as s-functions in C code. With the help of the Real-Time Workshop toolbox, from Mathworks Inc, we can thereafter compile our code to create a WinCon Client and a WinCon Server. WinCon Server and Client communicate with one another with the use of RTX Runtime from VenturCom Inc. Currently, both the Client and Server are physically located on the same computer but have the capability of being on separate computers allowing for a more flexible implementation [31], see Appendix A for some details. The program itself runs from the Client function but the interaction between the operator and the program is done through the Server function. The code is implemented with a sampling period of  $1ms$ , in order to minimize problems associated with faster sampling, [32], and demands between 40% – 60% of the 666 MHz processor.

The different control techniques will be briefly explained here but for a more thorough introduction to teleoperation prior to wave variable the reader is directed toward [1] and [13] for a more in depth introduction to more advanced control schemes.

### 4.1.1 Multi-rate sampling

The Simulink schematic contains two loops, a fast loop that updates at  $1000Hz$  and a slow loop that updates at  $200Hz$ , shown in Figure 4.1 without the communication medium. The fast loop reads the encoder values, finds the filtered derivatives and, via the Jacobians, calculates the force of the PI controller. This force is converted into joint torques using the Jacobians and is applied to the actuators via the motor controllers. The slow loop recalculates the Jacobians and their transposes using the most recent positions. The clock rate is selected in order to minimize the discretization effect, [32]. Future reference to the Jacobian will omit mentioning the slow loop. Also for simplification, it is assumed that the slave Jacobian is calculated within the safety feature block.

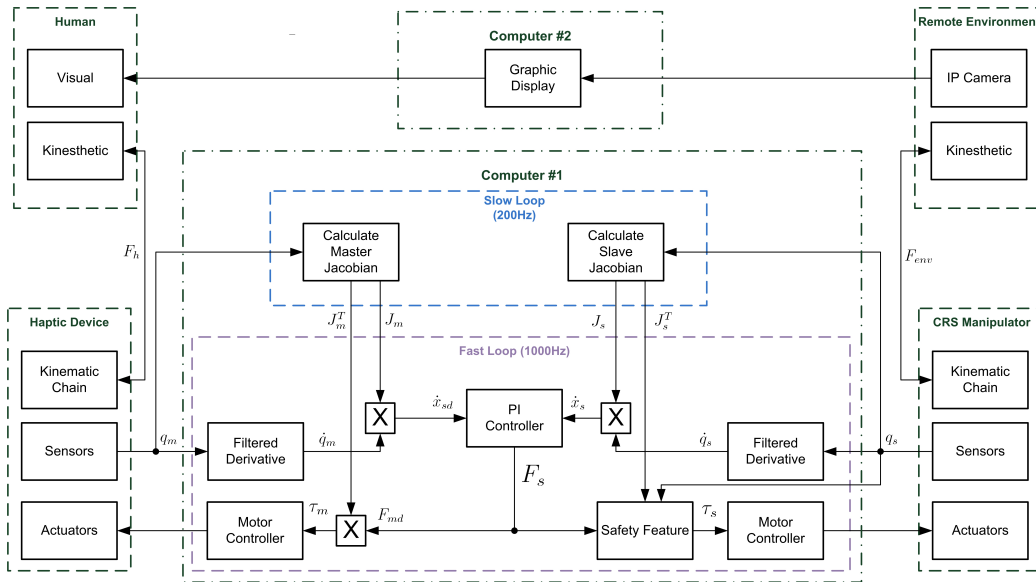


Figure 4.1: Block diagram of control algorithm

## 4.2 Kinesthetic force-based scheme

The KFB control scheme consists of using a proportional-derivative (PD) controller on each individual joint of the slave robot. This allows to move each joint to a desired angle with a controller following (4.1) with controller gains defined in Table 4.1. The corresponding block diagram is illustrated in Figure 4.2.

$$\tau = K_p(q_{sd} - q_s) - K_d \frac{s}{\tau_c s + 1} q_s \quad (4.1)$$

Where:

$\tau$  is the torque to apply to the joint

$q_{sd}$  is the desired joint angle

$q_s$  is the slave joint angle

$K_p$  is the proportional gain

$K_d$  is the derivative gain

$\frac{1}{\tau_c}$  is the cut-off frequency of the low-pass filter

Joint	1	2	3	4	5	6
$K_p$	143.24	143.24	143.24	28.65	28.65	17.19
$K_d$	2.292	2.292	2.292	0.2865	0.2865	0.1146

Table 4.1: PD controller gains with joint angle in radians

This controller scheme, of Figure 4.2, requires a sensor on the end-effector of the slave robot in order to detect the interaction forces between the manipulator and the remote environment. With this sensor, this scheme provides a very true feel to the user but only in the free-motion case or if contact is made with a compliant object. In Figure 4.2 the user moves the master robot and its movement is sensed by the encoders on all active joints and represented as  $q_m$ . Knowing all active joint angles, we then find the generalized coordinate position and orientation of the master manipulator, represented by  $x_m$ , with the use of forward kinematics described earlier. This desired position and orientation is then transmitted through the communication medium and,

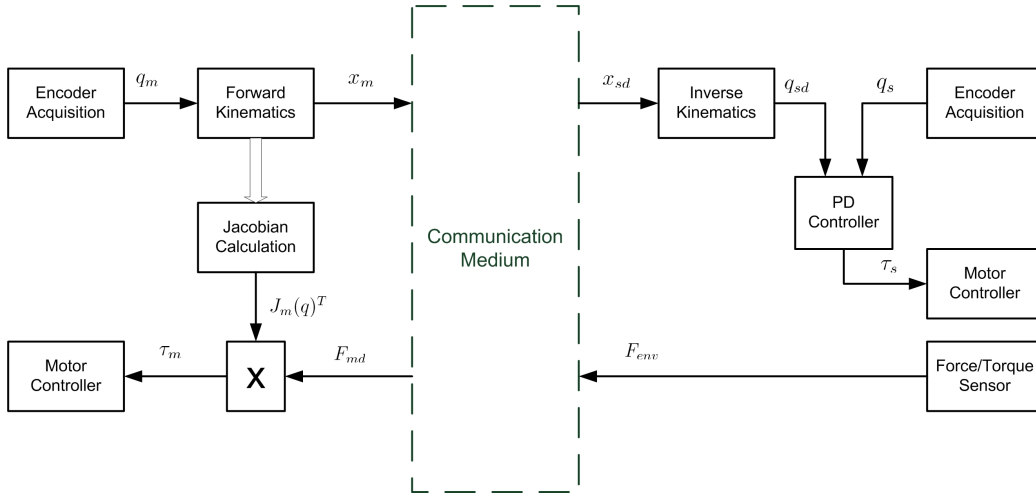


Figure 4.2: Block diagram of the PD controller

with the help of inverse kinematics, we can determine the desired joint angles of the slave manipulator,  $q_{sd}$ . A PD controller seeks to bring the slave joint angles to the desired angles by applying a certain torque,  $\tau_s$ . While the slave manipulator is moving there is a sensor that detects the 3 generalized coordinate forces and the 3 generalized coordinate torques,  $F_{env}$ . These sensed forces are transmitted back to the master manipulator where they are converted to the joint space via the Jacobian,  $J_m(q)^T$ . Finally, these torques are applied to the master manipulator by the motor controller and the operator is able to feel the interaction forces his movements are creating at the remote environment.

### 4.2.1 Safety Considerations

In order to provide some safety during operations we have included a safety feature to our design. The inverse kinematics obtains all 8 solutions and determines which ones are feasible and which one will create the least movement of all joints combined. If no solution is feasible, no motion is generated. Clutching is also implemented in order to maximize the use of the bigger workspace of the slave manipulator.

### 4.3 Position error-based scheme

The PEB control scheme consists of using a proportional-integral (PI) controller<sup>1</sup> on the generalized velocity of the slave and master end-effectors. This determines a generalized force that the robot needs to apply and is then converted to a desired torque for each joint. The controller equation originally take the form of (4.2) which can be simplified to (4.3). The controller gains, found experimentally in this work, are given in Table 4.2 and a block diagram is illustrated in Figure 4.3.

$$\begin{aligned} F_s(t) &= K_i \int_0^t (\dot{x}_m - \dot{x}_s) ds + K_{pi}(\dot{x}_m - \dot{x}_s) \\ F_m(t) &= K_i \int_0^t (\dot{x}_s - \dot{x}_m) ds + K_{pi}(\dot{x}_s - \dot{x}_m) \end{aligned} \quad (4.2)$$

$$\begin{aligned} F_s(t) &= K_i \int_0^t (\dot{x}_{sd} - \dot{x}_s) ds + K_{pi}(\dot{x}_{sd} - \dot{x}_s) \\ F_{md} &= -F_s \end{aligned} \quad (4.3)$$

Where:

$F_s$  is the generalized force to apply by the slave

$F_{md}$  is the desired generalized force to apply by the master

$\dot{x}_{sd}$  is the desired generalized slave velocity

$\dot{x}_s$  is the generalized slave velocity

$K_i$  is the integral gain

$K_{pi}$  is the proportional gain

This type of controller requires no sensor on the end-effector of the slave robot and therefore does not represent an accurate representation of the interaction forces with the remote environment. In Figure 4.3 the user moves the master robot and its movement is sensed by the encoders on all active joints and are represented as  $q_m$ . Having the different joint angles, we are

<sup>1</sup>It is named PI since we are using the integral action, but this controller may be viewed as a PD controller if the integral of velocity is first computed.

Joint	1	2	3	4	5	6
$K_i$	30	30	30	5000	2500	2500
$K_{pi}$	0.3	0.3	0.3	200	100	100

Table 4.2: PI controller gains with velocity in centimetres per seconds and radians per seconds

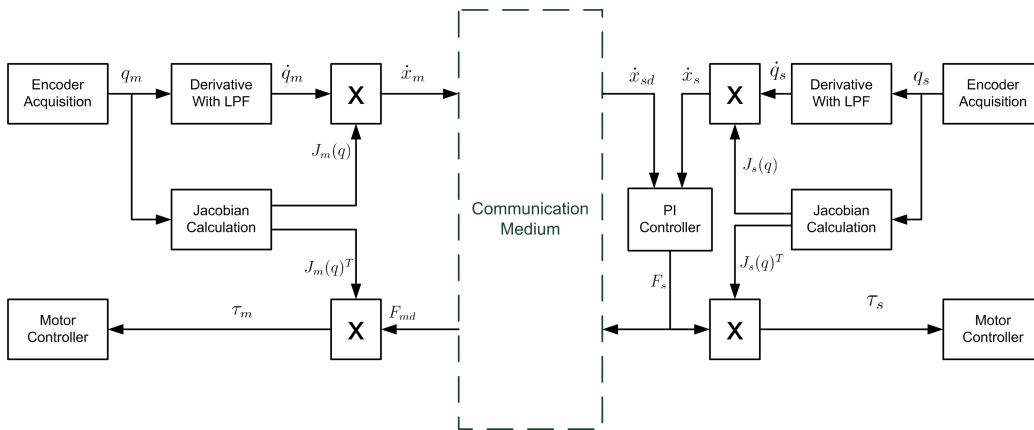


Figure 4.3: Block diagram of the PI controller

able to calculate the Jacobian. The joint velocities are obtained using a filtered derivative, *i.e.*,

$$[\dot{q}_m] = \frac{s}{\tau_c s + 1} [q_m]$$

This filtering creates a problem by introducing time delay to the signals due to the settling time. Given that the CRS manipulator has encoders with 1 000 counts per revolution (CPR) but also has a 1:100 transmission ratio, this means an effective 100 000 CPR. This signal has very little noise and therefore little filtering is done, a time constant of  $2ms$ . The settling time of the filter showed to be very important since we are calculating the Jacobian every five milliseconds and need the signal to be of similar time stamp. Figure 4.4 demonstrates the calculations to determine the generalized velocity for the slave.

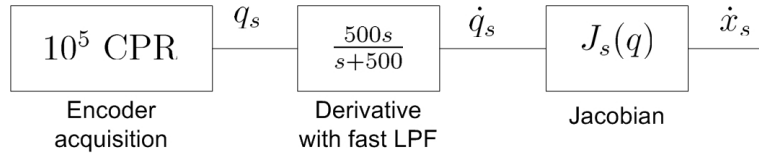


Figure 4.4: Finding the generalized velocity for the slave manipulator

For the haptic device, the signal is not as clean. The encoders also are 1 000 CPR encoders but there is no transmission. We therefore decided to use greater filtering in order to remove more noise and as such are using  $\tau_c = 0.025$  for the filtering of the master's joint velocities.

Having the rate of change of each joints,  $\dot{q}_m$ , and the Jacobian,  $J_m(q)$ , we are able to get the generalized linear velocities and angular velocities of the master manipulator,  $\dot{x}_m$ . The velocity is then transmitted to the slave side via the communication medium and is then referred to as the desired slave velocity,  $\dot{x}_{sd}$ . Meanwhile the slave's general velocities are also obtained in the same fashion. Keeping in mind that if both manipulators are initialized properly, the integral of the velocity will give us a desired position to track. Both  $\dot{x}_s$  and  $\dot{x}_{sd}$  are the inputs to the PI controller which determines the generalized force to apply in order to recreate the movement of the operator's hand. This generalized force,  $F_s$ , is then converted back to the joint space via the Jacobian and finally applied to each joint of the slave robot. The same procedure is applied at the master side such that the forces seek to push back the operator's hand. The feedback to the operator's hand is inaccurate and includes the dynamics of the slave manipulator.

### 4.3.1 Safety Considerations

In order to provide safety during operation, we have included a feature to our design that prevents damage to the CRS robot and to the remote environment. Normally, manipulators are being controlled using inverse kinematics, as with the KFB controller. The inverse kinematics are aware of the maximum joint angle for each joint. When the inverse kinematics are calculated, the program ensures that the desired joint angles is within these limits.

With our PI controller being in Cartesian space, the controller does not have such *a priori*. For that reason we have added such knowledge within the Jacobian calculations, Figure 4.5. The algorithm for the Jacobian calculation block including the safety feature is shown in Table 4.3

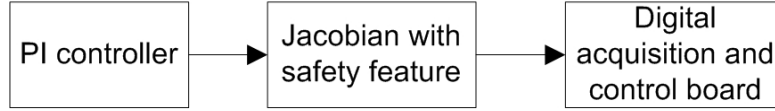


Figure 4.5: PI controller with safety

---

Forward Kinematics
Jacobian Calculation
Control Force
$\tau_{pi} = J^T F_s$
direction = sign( $\tau_{pi}$ )
Safety consideration
if reached $q_{max}$ and direction is 1
use $\tau_{pd} = K_p(q_{max} - q_s) - K_d\dot{q}_s$
else if reached $q_{min}$ and direction is -1
use $\tau_{pd} = K_p(q_{min} - q_s) - K_d\dot{q}_s$
else
use $\tau_{pi}$

---

Table 4.3: Torque calculations algorithm with safety feature

The algorithm has input arguments the calculated generalized force to apply,  $F_s(t) = K_i \int_0^t (\dot{x}_{sd} - \dot{x}_s) ds + K_{pi} (\dot{x}_{sd} - \dot{x}_s)$ , and the joint angle of the CRS manipulator. First, the algorithm calculates the Jacobian as per described in previous section. Via the Jacobian, the algorithm determines what joint torques are needed to produce a force of  $F_s$ , defined as  $\tau_{pi}$ . Next, the algorithm checks in what direction this torque will rotate the joints.

If the joint has reached its maximum, or minimum, value *and* the torque calculated attempts to rotate it past its maximum or minimum value, then the program will not use that torque. Rather it will output the torque required for the joint to stay at its current location. Clutching is also implemented in order to maximize the use of the bigger workspace of the slave manipulator.

Using an algorithm of this form ensures that the safety of the slave manip-



ulator is enforced at the slave side of the communication medium. It would have been simpler to determine if a position was possible or not directly at the master side but this would have rendered future expansion, mainly wave variables, to be more complicated than necessary to implement.

## 4.4 Wave variables

The wave variable controller is an extension of the PEB controller. It is implemented using a single wave for each degree of freedom, a total of 6 independent waves, Figure 4.6. Also depicted on Figure 4.6 is a low-pass filter (LPF) on each out-going waves, as recommended in [17, 33]. In Figure 4.6, the generalized master velocity,  $\dot{x}_m$ , and the computed slave force,  $F_s$ , are calculated in the same manner as with the PEB controller. They are then encoded into waves and simultaneously the received wave is decoded to the desired variable ( $F_{md}$  for the master,  $\dot{x}_{sd}$  for the slave), through (4.4) and (4.5). The wave transformation variables, found experimentally in this work, are given in Table 4.4.

$$F_{md} = b\dot{x}_m + \sqrt{2b}v_m \quad [u_m] = \frac{1}{1 + \tau_m s} \left[ \sqrt{2b}\dot{x}_m + v_m \right] \quad (4.4)$$

$$\dot{x}_{sd} = \sqrt{\frac{2}{b}}v_s - \frac{F_s}{b} \quad [u_s] = \frac{1}{1 + \tau_s s} \left[ \sqrt{\frac{2}{b}}F_s - v_s \right] \quad (4.5)$$

$$\text{With: } v_m(t) = u_s(t - T) \text{ and } v_s(t) = u_m(t - T)$$

Where:

$F_s$  is the generalized force to apply by the slave

$F_{md}$  is the generalized desired master force

$\dot{x}_m$  is the generalized master velocity

$\dot{x}_{sd}$  is the generalized desired slave velocity

$b$  is the wave impedance

$\tau_m$  is the cut-off frequency of the LPF on the master's outgoing wave

$\tau_s$  is the cut-off frequency of the LPF on the slave's outgoing wave

$T$  is the communication time delay

$u_m$  is the outgoing wave from the master

$u_s$  is the outgoing wave from the slave

$v_m$  is the delayed incoming wave to the master

$v_s$  is the delayed incoming wave to the slave

DOF	$x$	$y$	$z$	$\phi$	$\theta$	$\psi$
$b$	1	1	1	2000	2000	2000
$\tau_m$	20	20	20	8	8	8
$\tau_s$	20	20	20	n/a	n/a	n/a

Table 4.4: Wave transformation variables

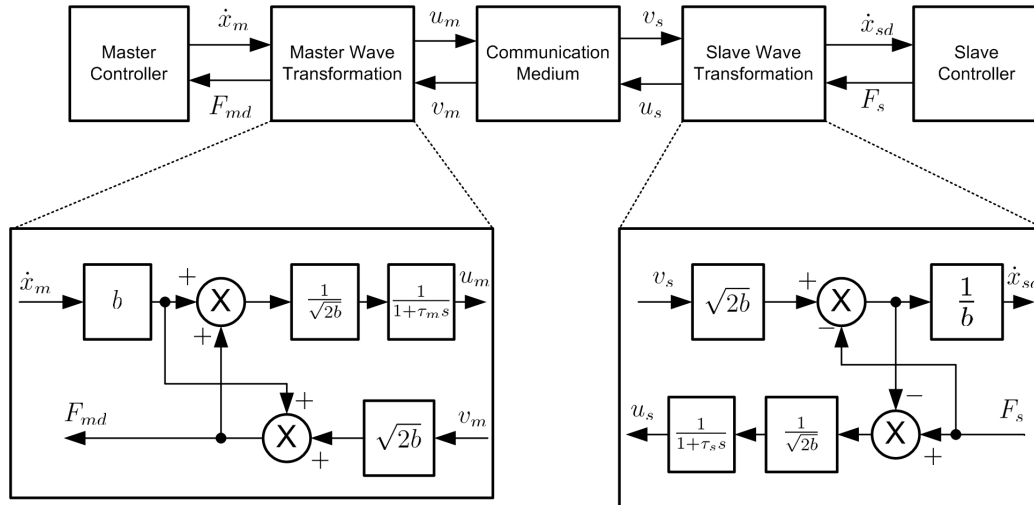


Figure 4.6: Wave variable block diagram

# Chapter 5

## Experimental Results

Different control schemes of bilateral teleoperation have been tested on the experimental setup shown in Figure 5.1. Data has been collected from the real-time experiments and is presented in this section. Five different experiments were conducted with all three control schemes.



Figure 5.1: Experimental setup at the Automatic Control Laboratory

## 5.1 Experiment with a compliant environment without communication time delay

The first experiment consists of applying a force down on a cardboard box's lid. The force at which the box's lid will resist the slave manipulator depends on how close to the edge the force is being applied. Thus by varying locations, the operator should be able to feel different interaction forces. The following graphs show the generalized position error between the master and the slave manipulators and the interaction force measured by the force/torque sensor. The KFB controller was used to obtain the results shown in Figure 5.3. PEB controller was used to obtain the results in Figure 5.4, but the actual error can be seen on Figure 5.5. The error shown in Figure 5.4 is determined by the  $\int_0^t (\dot{x}_{sd} - \dot{x}_s) ds$  term in (4.3). The error in Figure 5.5 shows the error using forward kinematics independently of the PEB controller code, Figure 5.2. The results obtained in Figure 5.6 and Figure 5.7, used wave variables instead of transmitting  $\dot{x}_m$  and  $F_s$ . These two Figures represent the two different errors associated to the PEB controller.

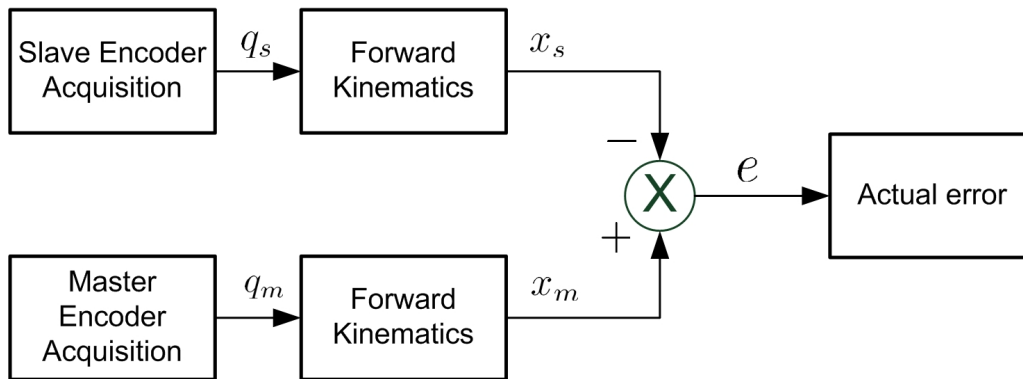


Figure 5.2: Independently calculated position error

.20.0

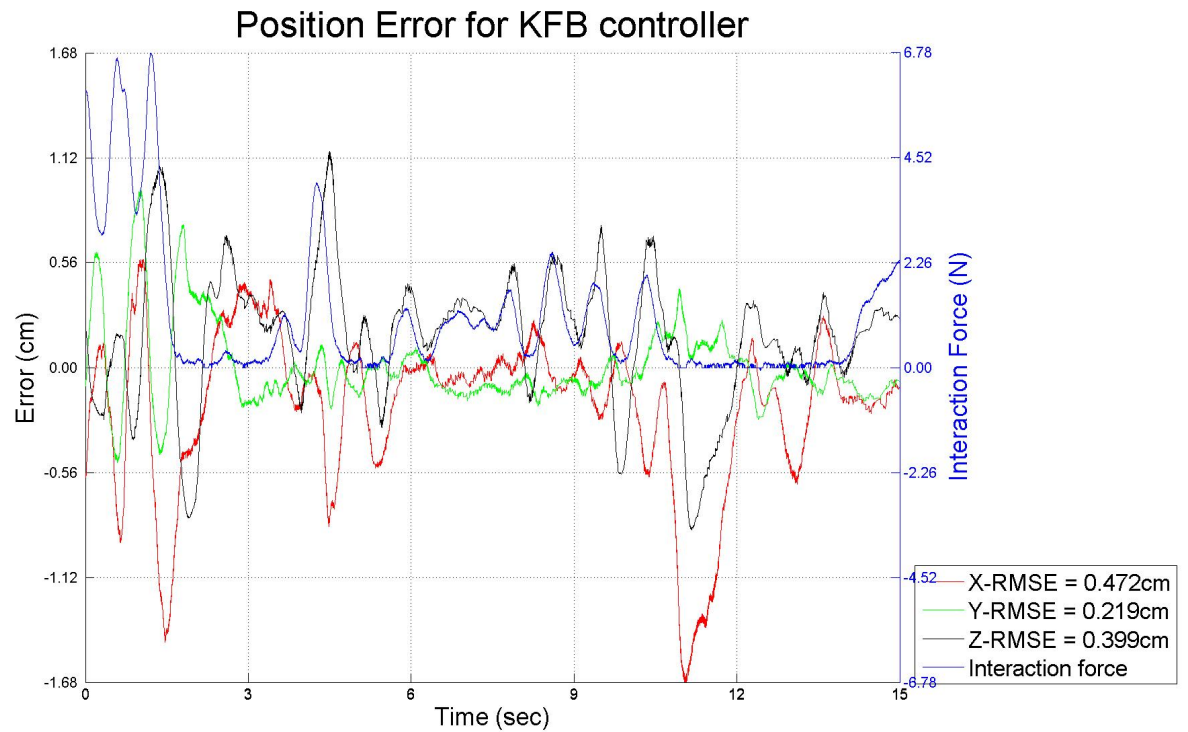


Figure 5.3: Position error of the KFB controller without time delays

### 5.1.1 Discussion 1

The KFB controller allowed for very accurate force feedback and allowed the operator to be aware of the force he was applying to the box, thus limiting the remote interaction to approximately  $7N$ . The PEB controller failed to provide, as for all of the following results, accurate feedback to the operator. Although it seems that the PEB controller outperforms the KFB controller, it is obvious that there is a position drift that happens due to the filtered derivative of the joint angles and integrating the generalized velocity, shown in Figure 5.5 when the position is calculated using forward kinematics. The same result can be said when using wave variables, although wave variables did represent a greater error, Figure 5.7.

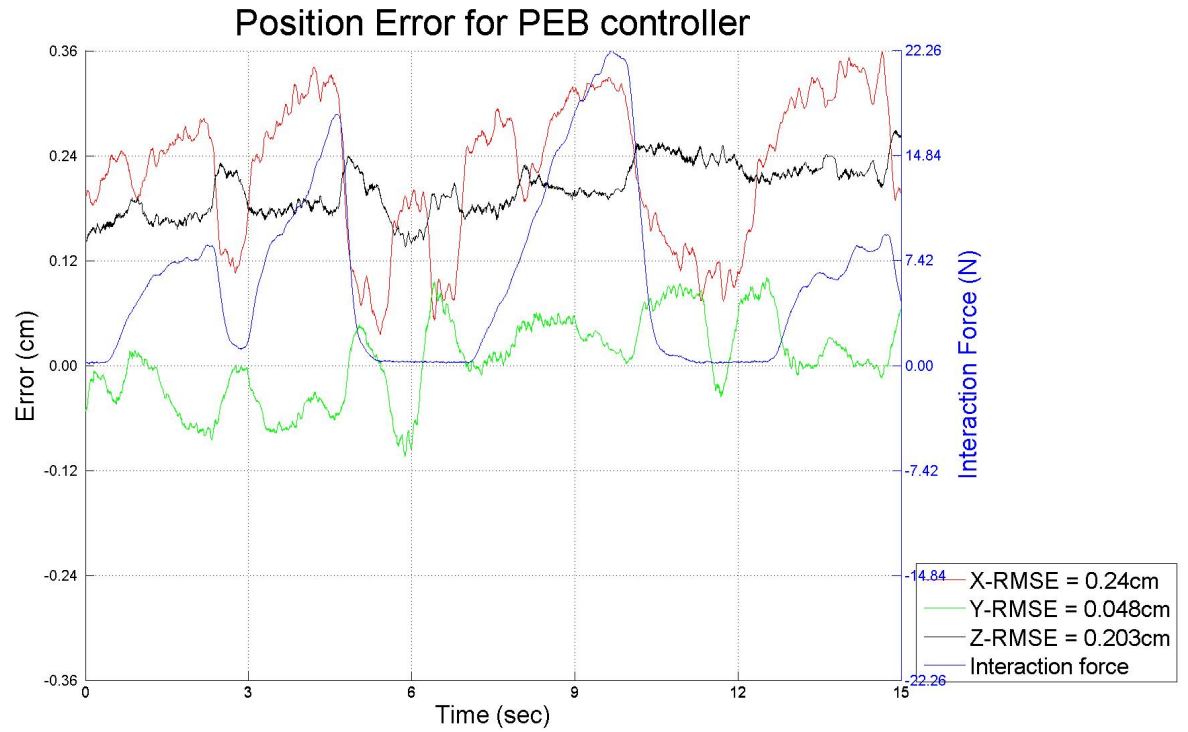


Figure 5.4: Position error of the PEB controller without time delay, error measured at the controller

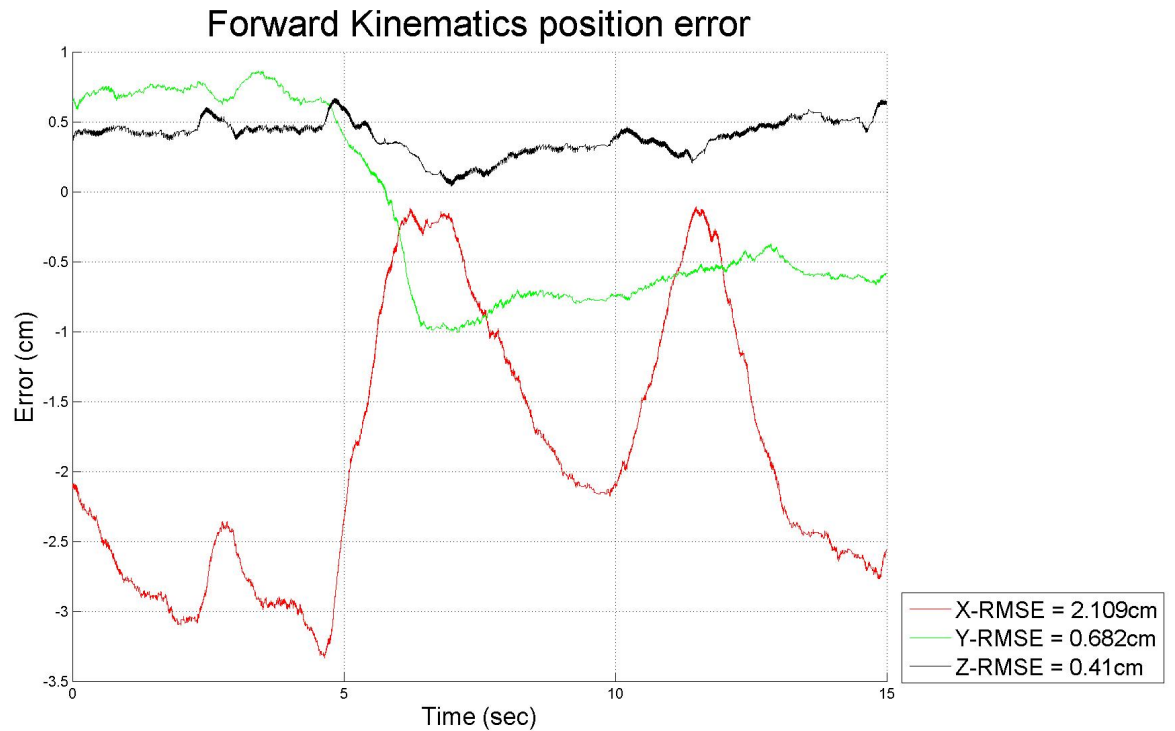


Figure 5.5: Position error of the PEB controller without time delay, calculated by forward kinematics

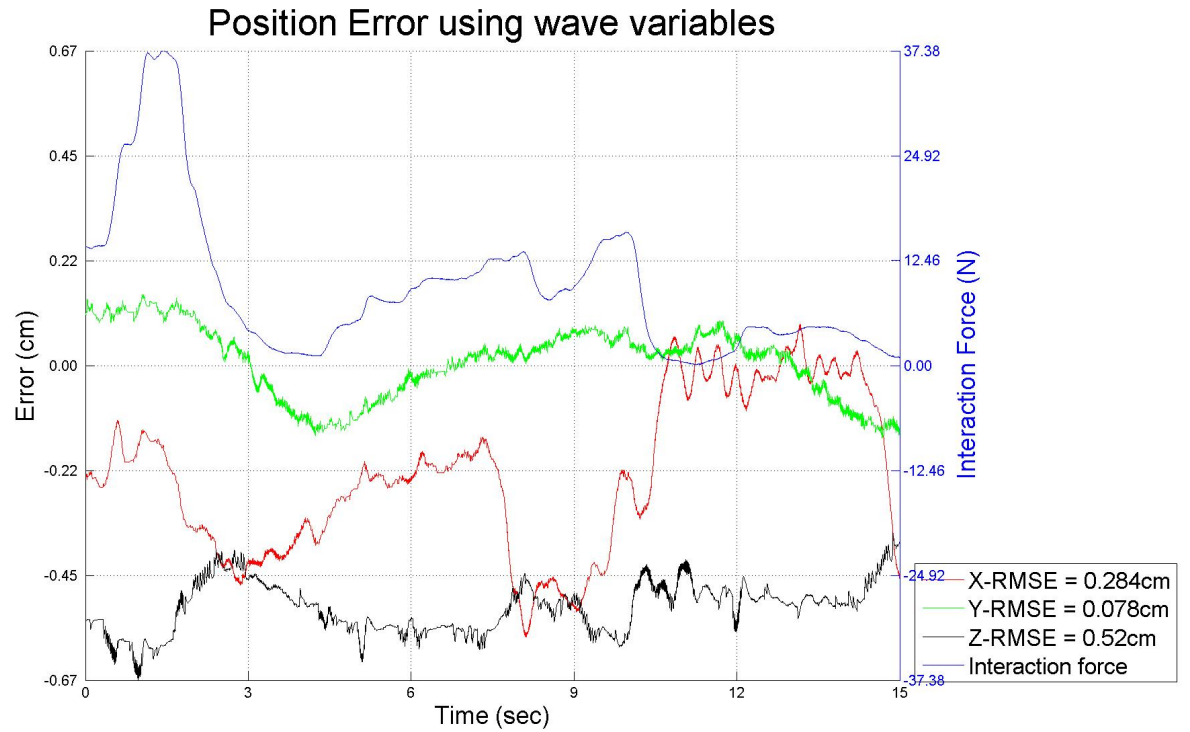


Figure 5.6: Position error using wave variables without time delay, error measured at the controller

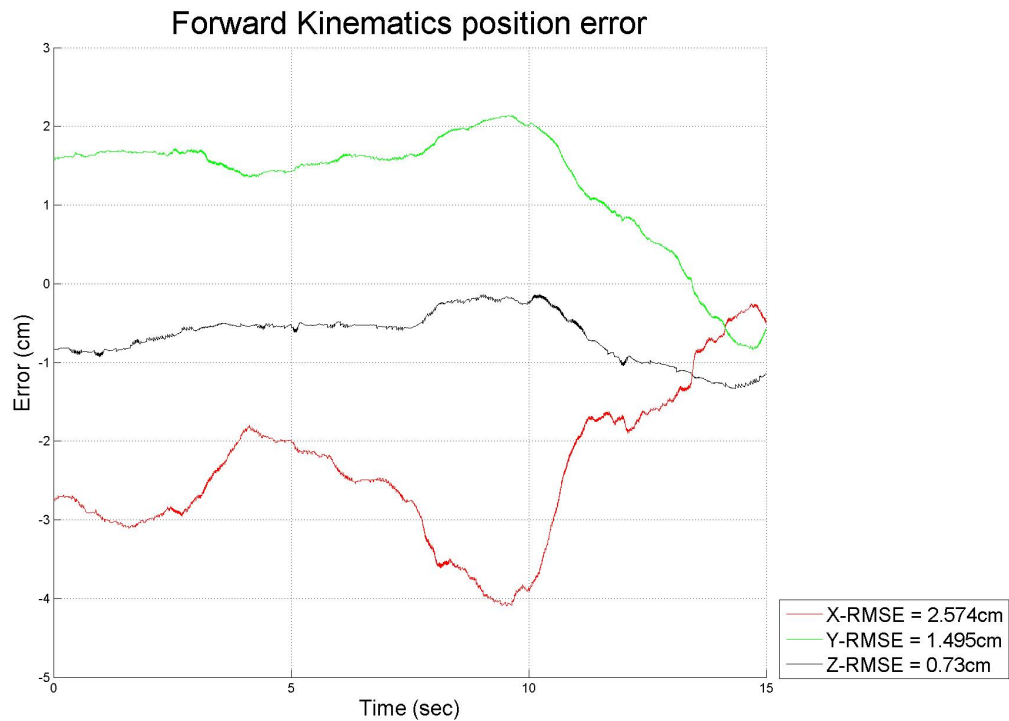


Figure 5.7: Position error using wave variables without time delay, calculated by forward kinematics

## 5.2 Experiment with a stiff environment without communication time delay

This experiment consists of applying a force on a stiff surface, maintain a certain pressure and glide along that surface, similarly to writing. The following graphs show the generalized position error between the master and the slave manipulators and the interaction force measured by the force/torque sensor. The KFB controller was used to obtain the results shown in Figure 5.8. The PEB controller was used to obtain the results in Figure 5.9, and the actual error can be seen in Figure 5.10. Figure 5.10 illustrates the error using forward kinematics independently of the PEB controller code, described earlier. The results, illustrated in Figure 5.11 and Figure 5.12, were obtained by transmitting wave variables.

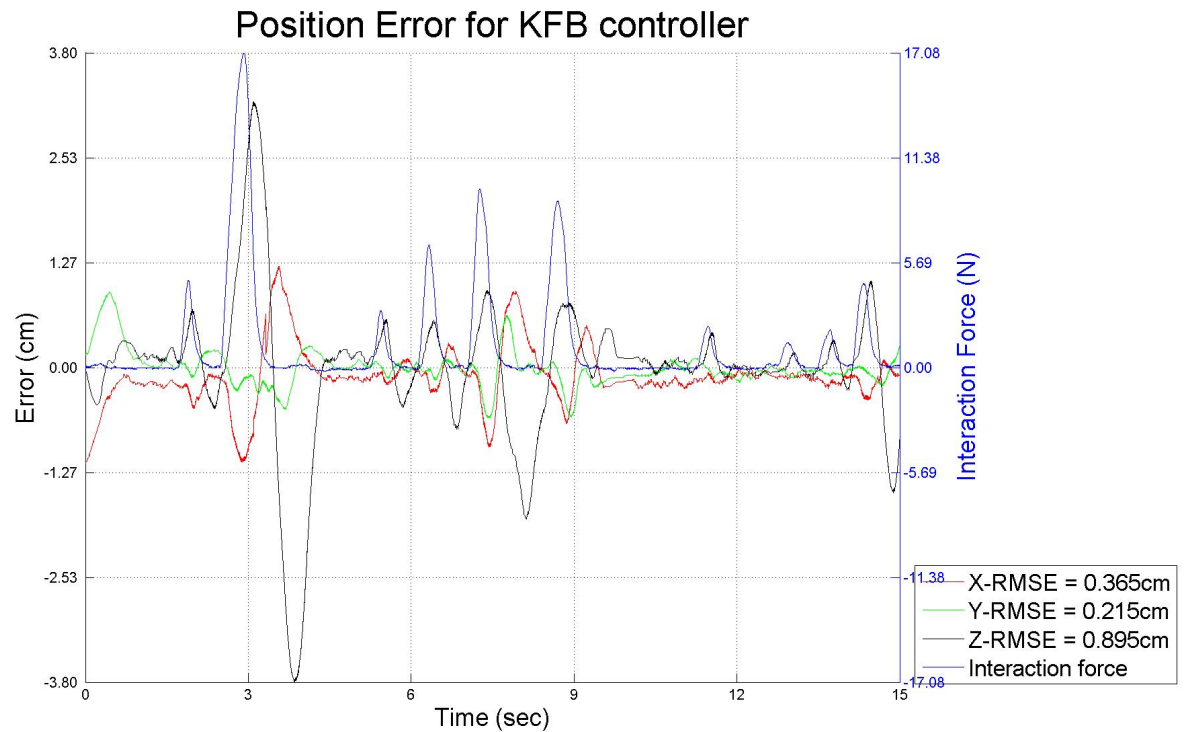


Figure 5.8: Position error of the KFB controller without time delays



### 5.2.1 Discussion 2

The KFB controller demonstrates the non-passive behavior of the KFB scheme. When the slave manipulator made contact with the surface, the force sensed by the sensor pushed back the hand of the operator and oscillation happened. This response was expected and is shown in Figure 5.8. The PEB controller did not have this oscillatory response but rather failed to provide knowledge about how much pressure was actually being exerted on the stiff surface. Although the operator was able to glide along the surface, the surface might have gotten damaged from such strong interaction force, shown in Figure 5.9. A position drift is once again evident from the greater position error shown in Figure 5.10, compared with Figure 5.9. With the transmission of wave variables the response was once again very similar to the PEB controller but demonstrated a greater error.

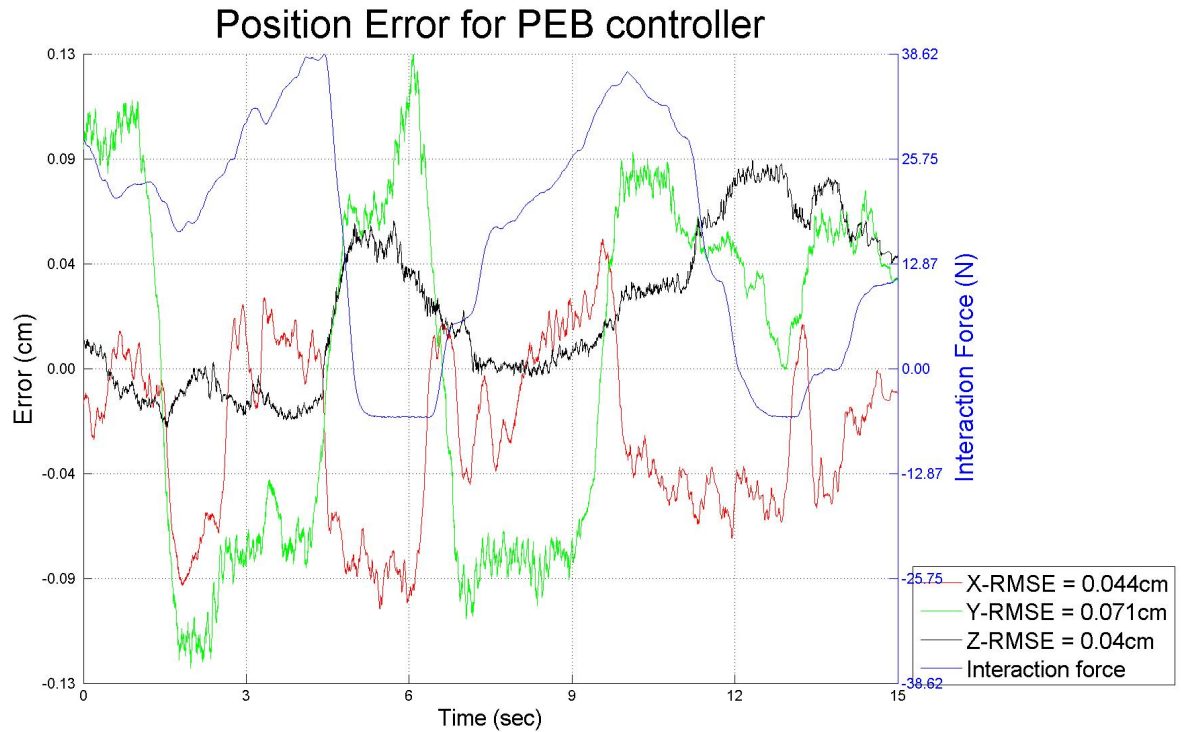


Figure 5.9: Position error of the PEB controller without time delay, error measured at the controller

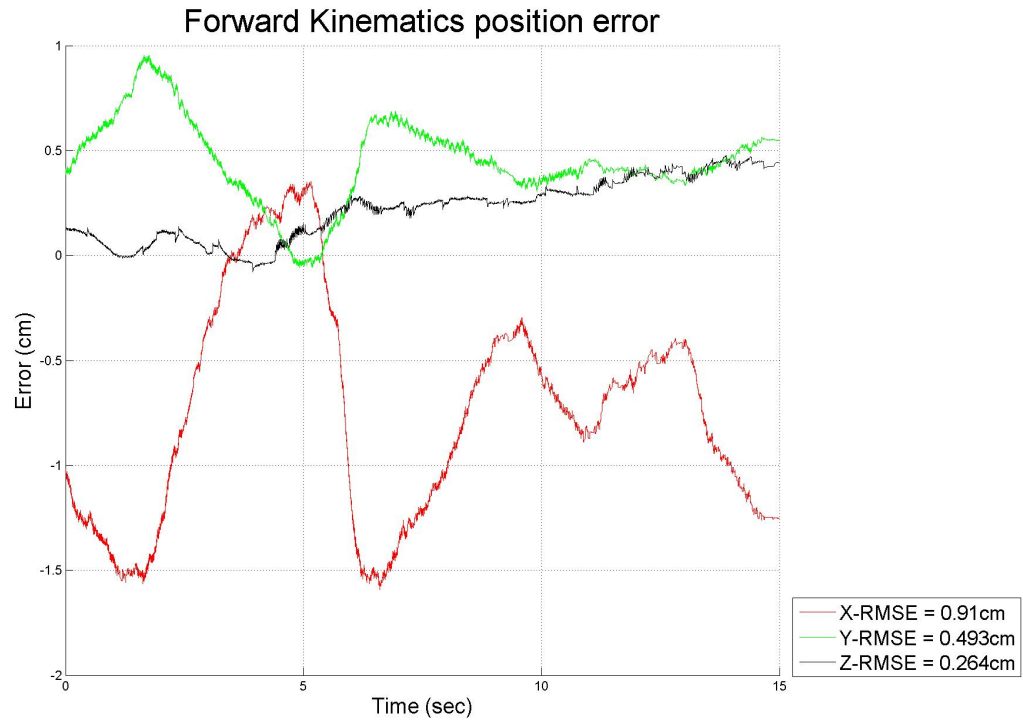


Figure 5.10: Position error of the PEB controller without time delay, calculated by forward kinematics

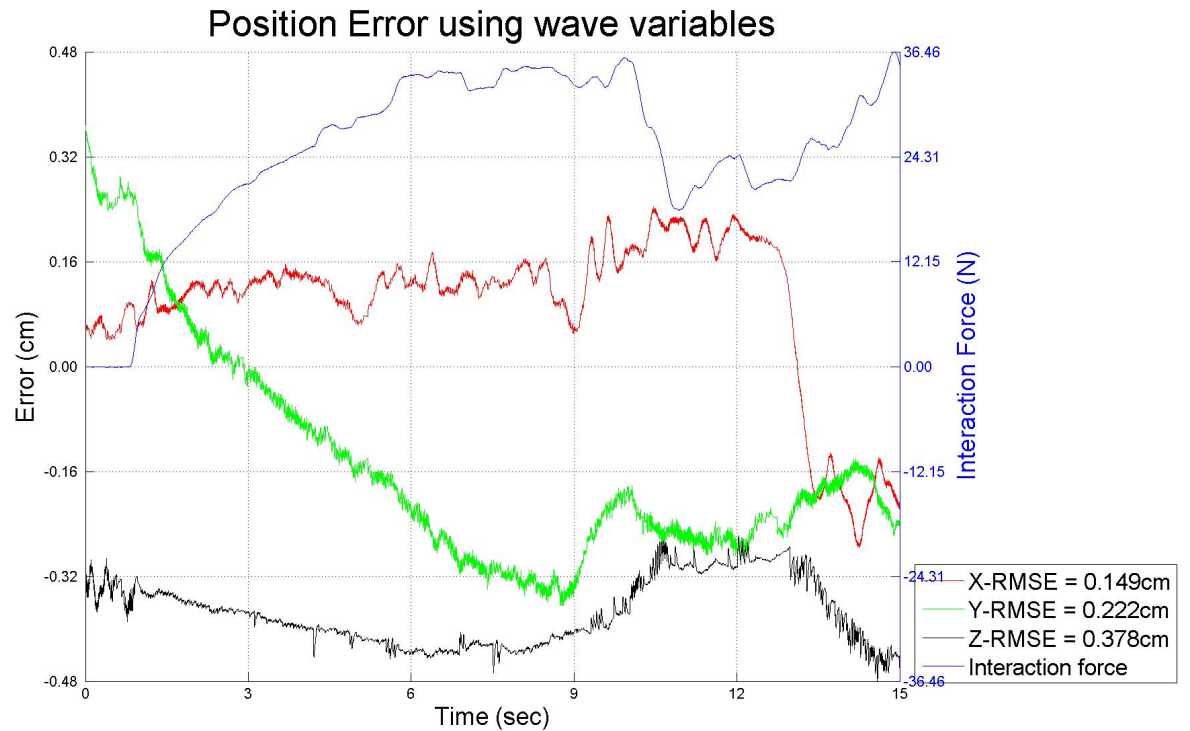


Figure 5.11: Position error using wave variables without time delay, error measured at the controller

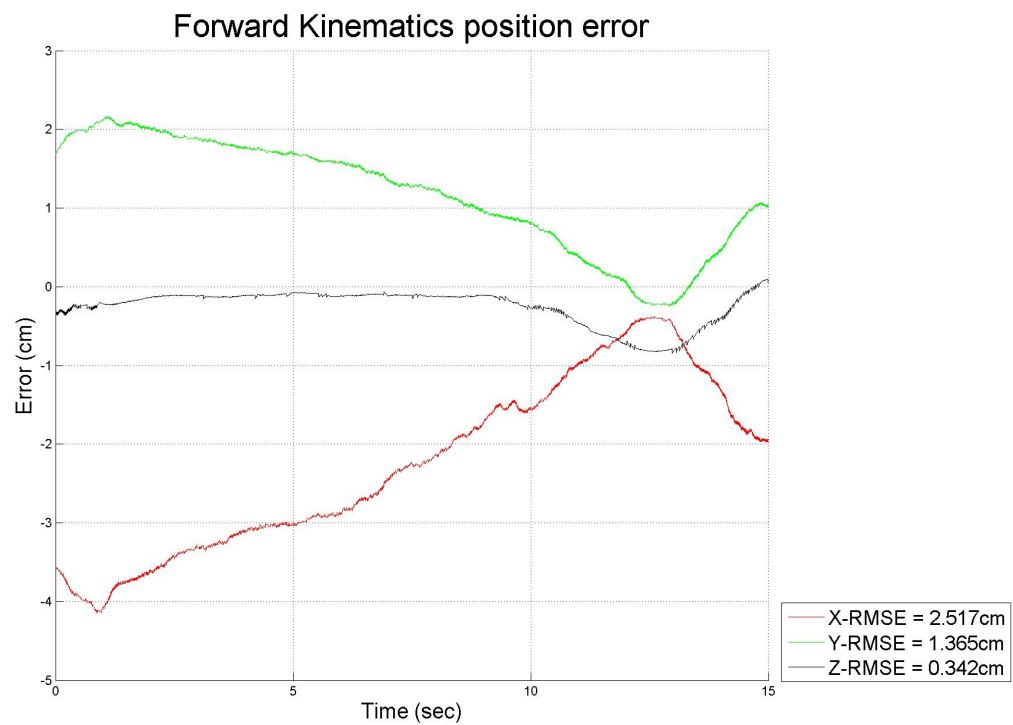


Figure 5.12: Position error using wave variables without time delay, calculated by forward kinematics

### 5.3 Free motion with communication time delay

Experiments with time delays present within the communication medium were performed using Simulink block *Transport Delay* with a time delay value of  $50ms$ . In Figure 5.13 we can clearly see that the slave manipulator is trailing behind the master manipulator. This time delay is known to have the ability to destabilize the system and deteriorate transparency. The results in Figure 5.13 were obtained using the KFB controller while the results in Figure 5.14 were obtained using the PEB controller and the results in Figure 5.15 were obtained by transmitting wave variables.

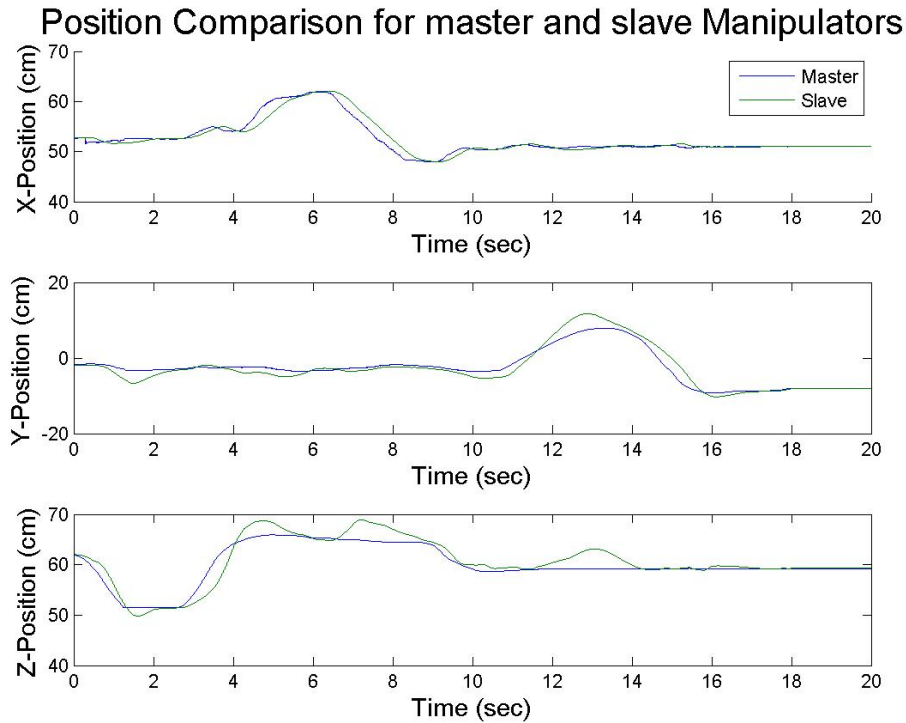


Figure 5.13: Position error of the KFB controller with constant time delay of  $50ms$  present

#### 5.3.1 Discussion 3

In Figure 5.13, it is obvious that the slave's response is delayed. The slave's response often overshoots when the master manipulator stops moving. In Figures 5.14 and 5.15 it is less apparent that a time delay is present since both signals seem to be synchronized with one another. The PEB controller once again demonstrates a clear position drift in the  $x$  direction.

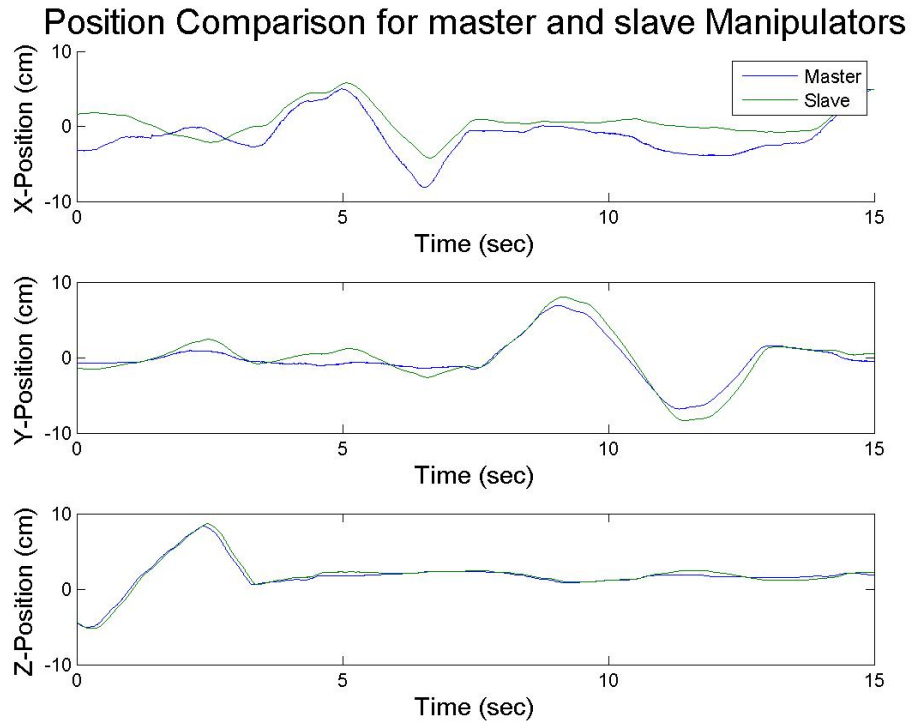


Figure 5.14: Position error of the PEB controller with constant time delay of  $50ms$  present

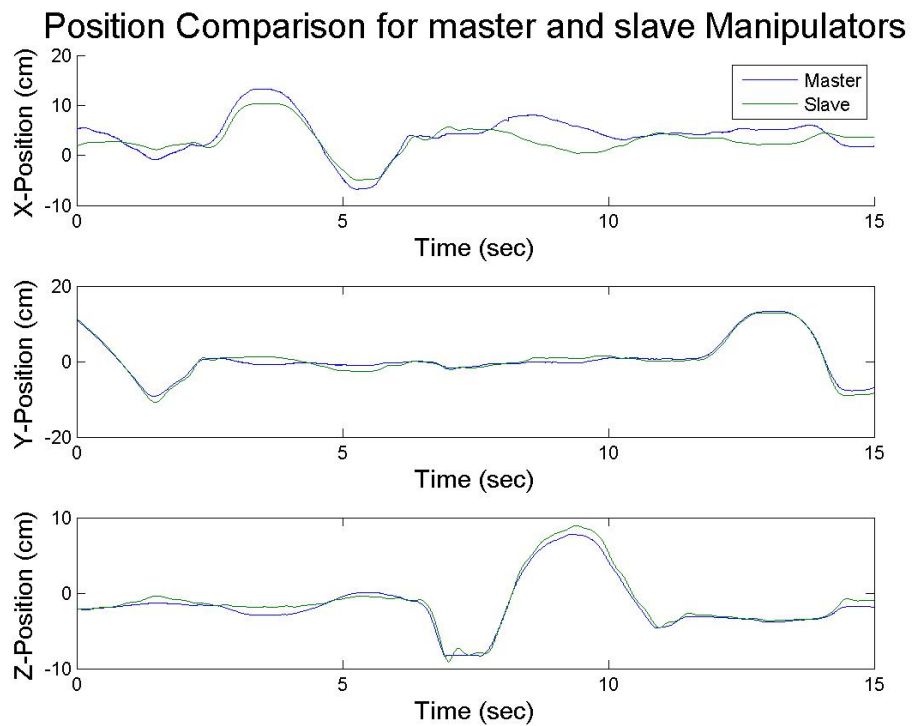


Figure 5.15: Position error using wave variables with constant time delay of  $50ms$  present

## 5.4 Experiment with a compliant environment with communication time delay

This experiment consists of applying a force down on a cardboard box's lid in the presence of time delays within the communication medium. The following graphs show the generalized position error between the master and the slave manipulators and the interaction force measured by the force/torque sensor. The KFB controller was used to obtain the results shown in Figure 5.16. PEB controller was used to obtain the results in Figure 5.17, but the actual error can be seen in Figure 5.18. Also, transmission of wave variables was used to obtain the results shown in Figures 5.19 and 5.20.

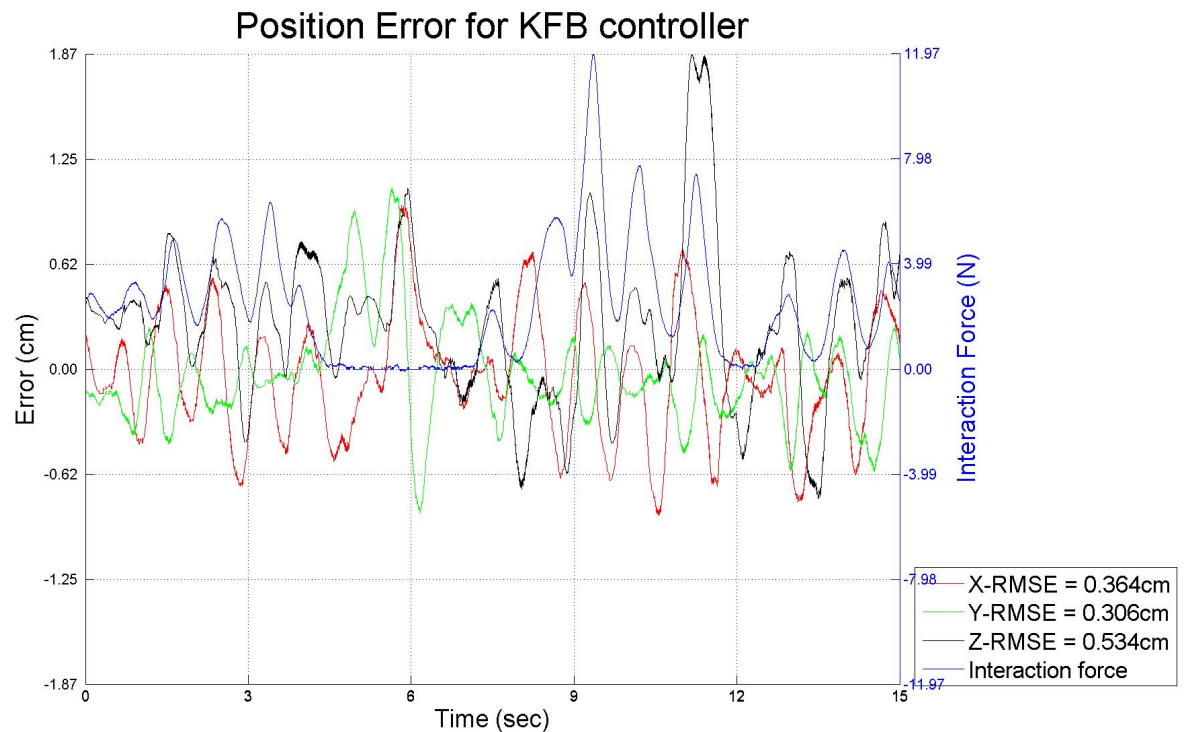


Figure 5.16: Position error of the KFB controller with constant time delay

### 5.4.1 Discussion 4

The KFB controller allowed for accurate delayed force feedback and allowed the operator to be aware of the force he had applied to the box, thus he was not aware of the *current* interaction force. The presence of time delay in the communication medium was not as apparent for the PEB controller, and, resulted in lower root mean square error (RMSE). Position drift is still noticeable but some of the error in Figure 5.18 should be attributed to the fact that the forward kinematics are not aware of the  $50ms$  time delay, thus comparing live error. Roughly speaking since the new desired position has not reached the slave yet, the slave is not trying to track the current position of the master, thus yielding a greater error. It is interesting to see that now the transmission of wave variables yielded a lower RMSE than the PEB controller, Figure 5.18 versus Figure 5.20.

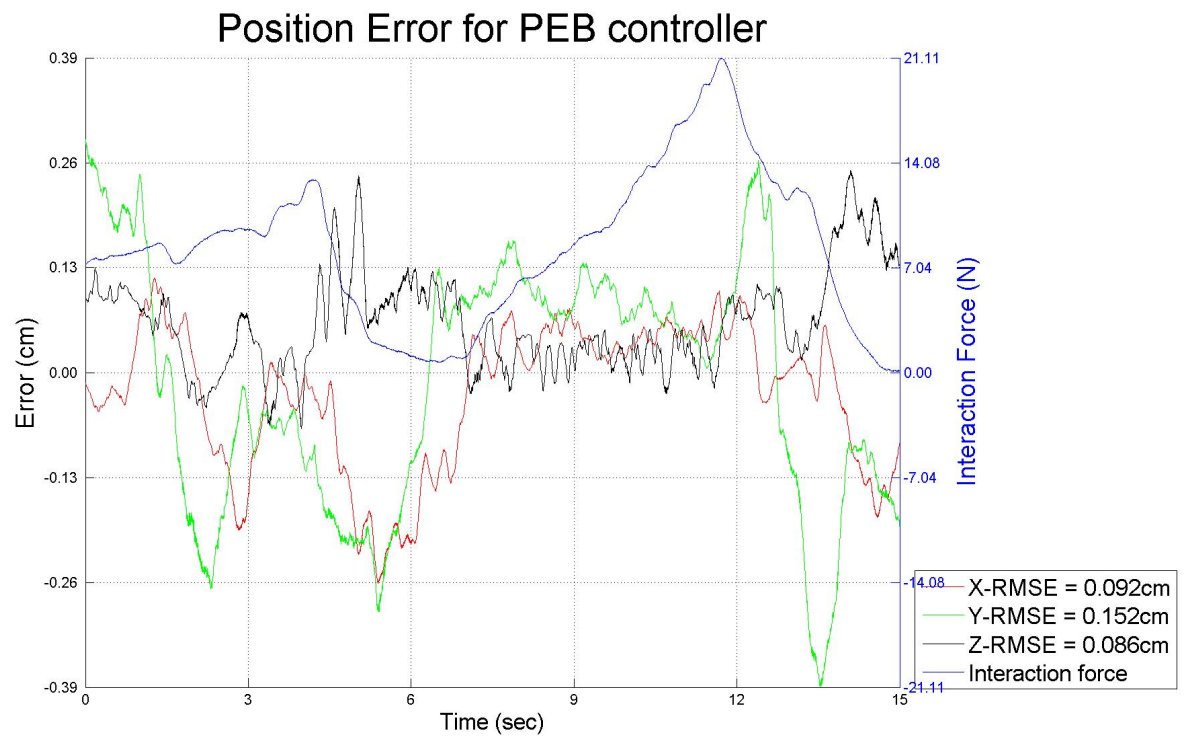


Figure 5.17: Position error of the PEB controller with constant time delay

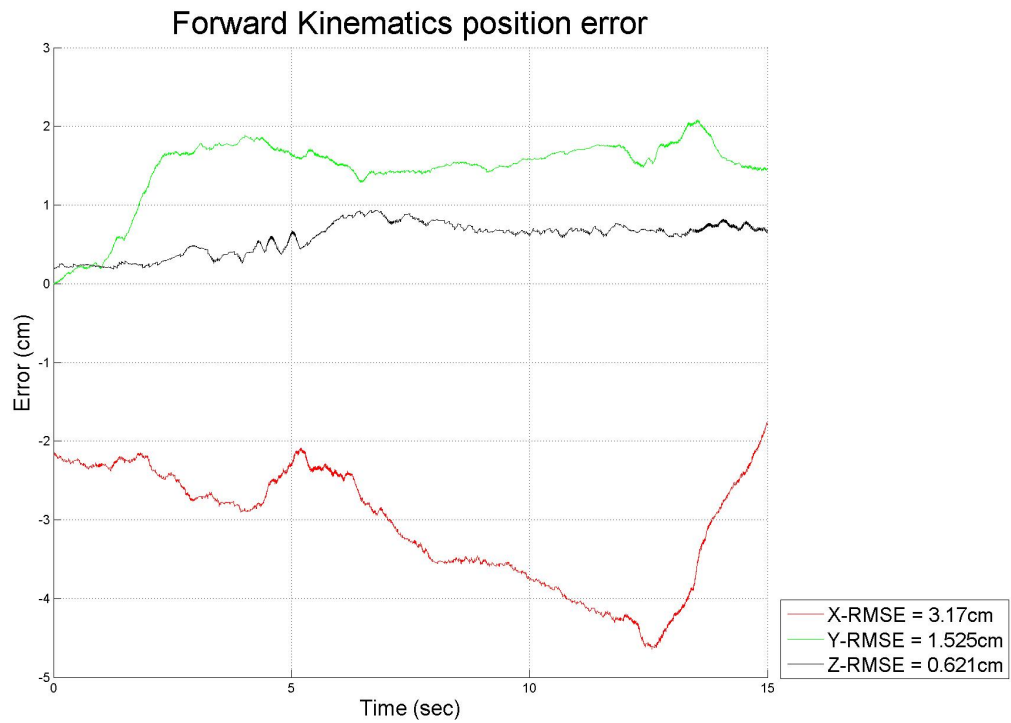


Figure 5.18: Position error of the PEB controller with constant time delay, calculated by forward kinematics

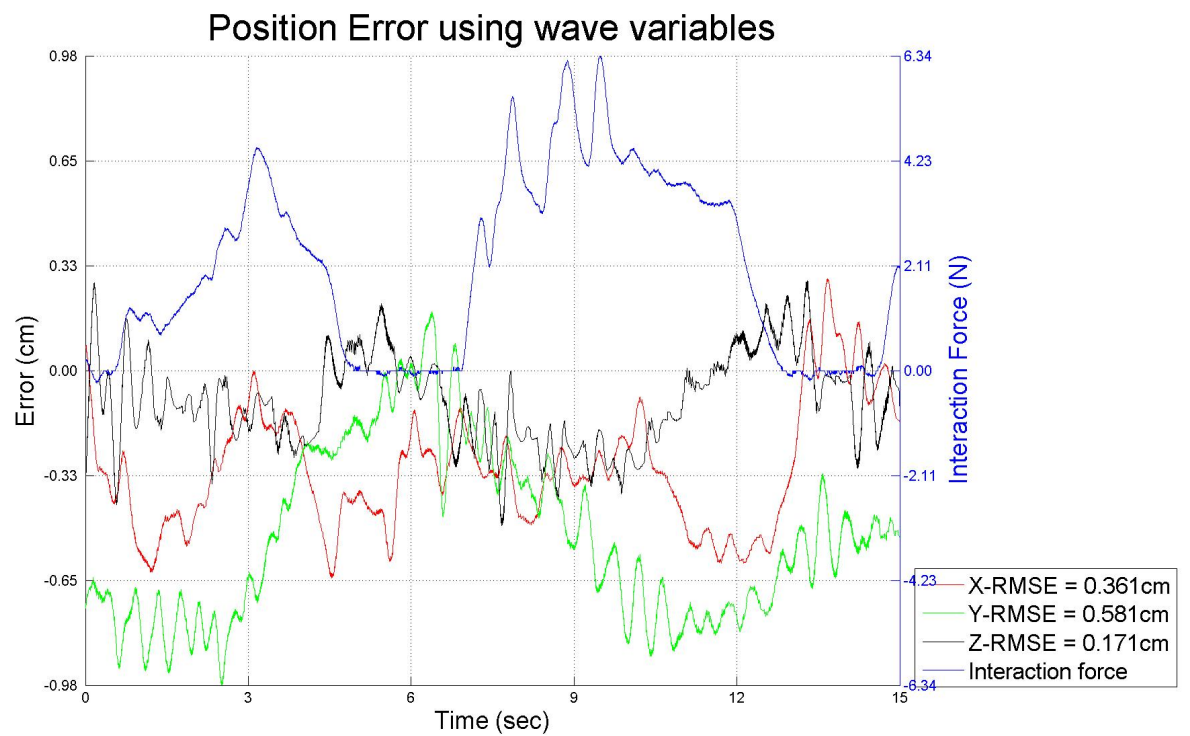


Figure 5.19: Position error using wave variables with constant time delay



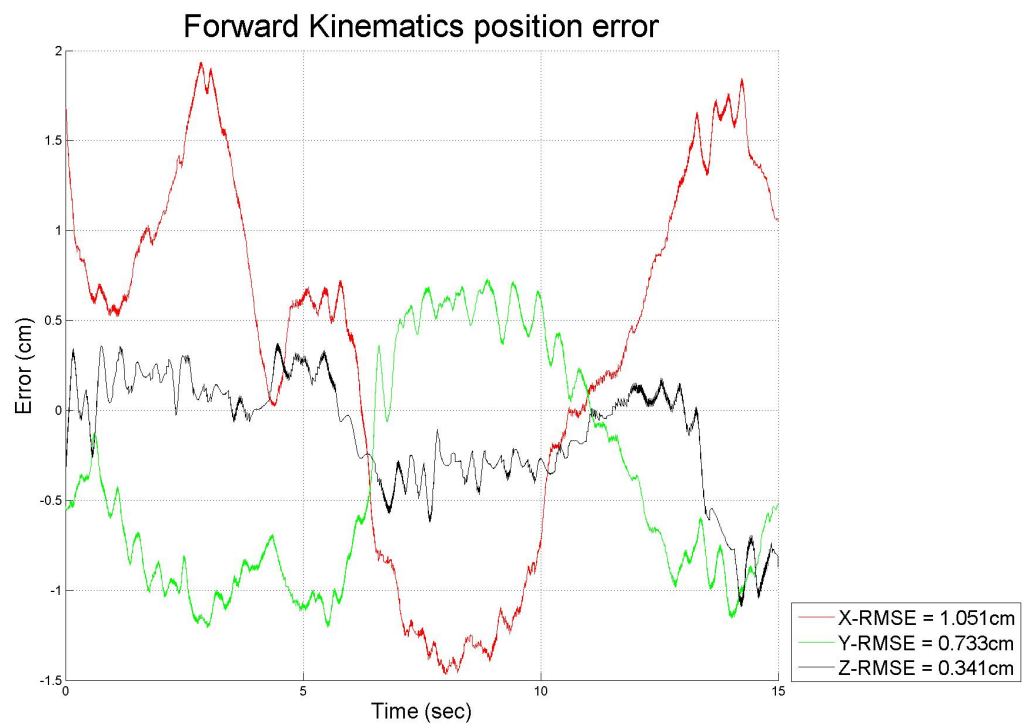


Figure 5.20: Position error using wave variables with constant time delay, calculated by forward kinematics

## 5.5 Experiment with a stiff environment with communication time delay

The last experiment consists of applying a force on a stiff surface, maintain a certain pressure and glide along that surface, with time delay present in the communication medium. The following graphs show the generalized position error between the master and the slave manipulators and the interaction force measured by the force/torque sensor. The KFB controller was used to obtain the results shown in Figure 5.16. PEB controller was used to obtain the results in Figure 5.22, but the actual error can be seen on Figure 5.23 and results obtained using wave variables are shown in Figure 5.24 and 5.25.

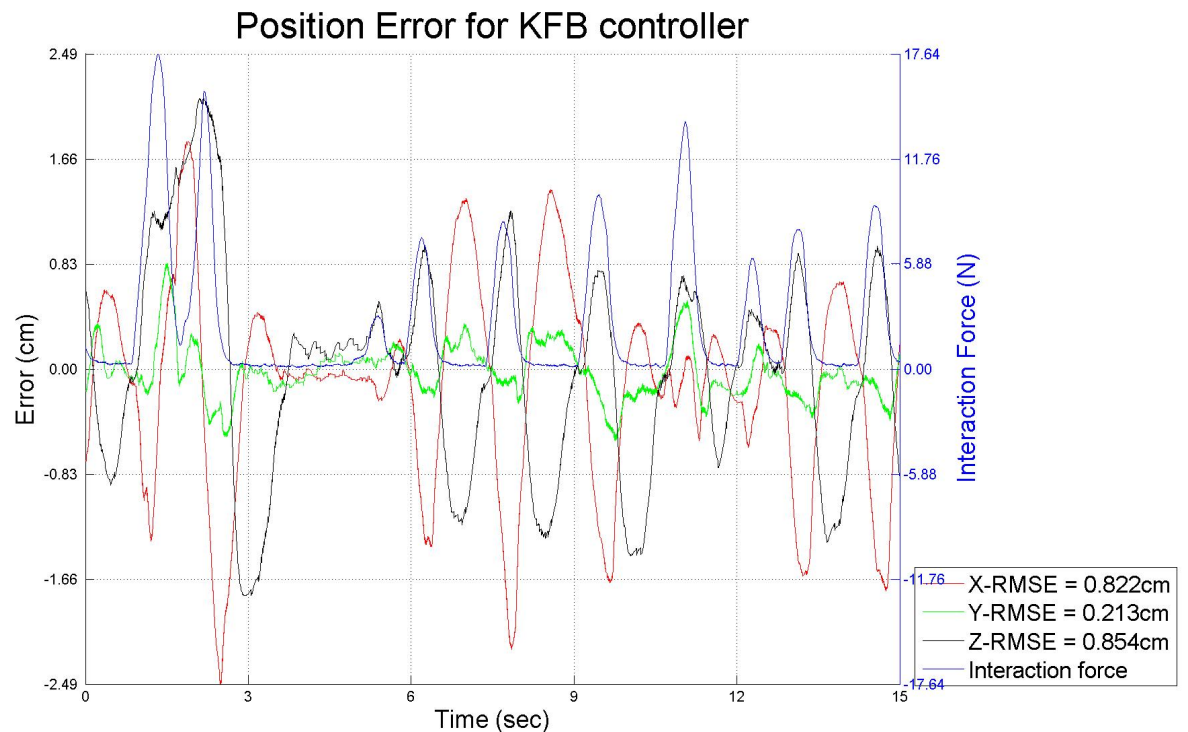


Figure 5.21: Position error of the KFB controller with constant time delay

### 5.5.1 Discussion 5

The KFB controller allowed for accurate delayed force feedback but once again, oscillations occurred due to the non-passive nature of KFB scheme. In Figure 5.21, the operator is not able to apply a constant force to the remote environment. If he were to try, the system would oscillate in a growing fashion until failure of the weakest component. The presence of time delay in the communication medium was not as apparent for the PEB controller, and, resulted in lower mean absolute error. For both the PEB controller and the transmission of wave variables, position drift is still noticeable but some of the error in Figure 5.23 should be attributed to the fact that the forward kinematics are not aware of the  $50ms$  time delay, thus comparing live error. Although the wave variables did not outperform the PEB controller in this experiment, we should keep in mind that the passivity of this controller is guaranteed even in the case of time delays whereas the PEB could have reacted in some unstable fashion.

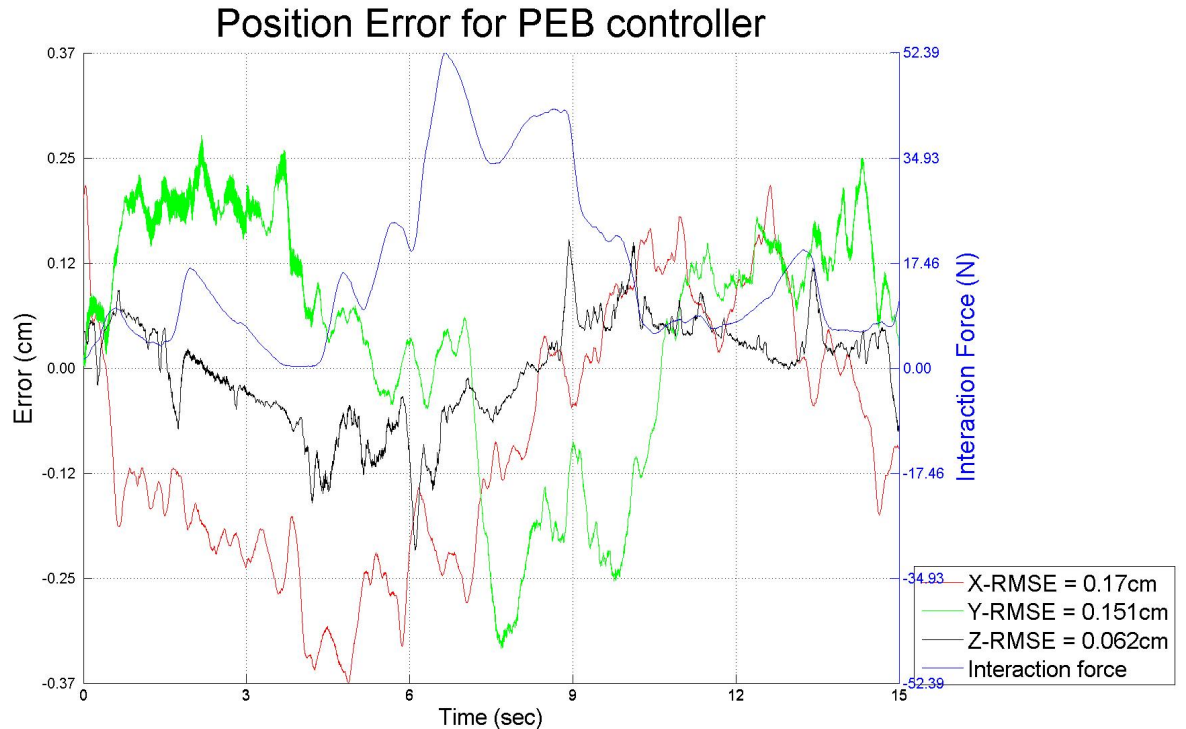


Figure 5.22: Position error of the PEB controller with constant time delay

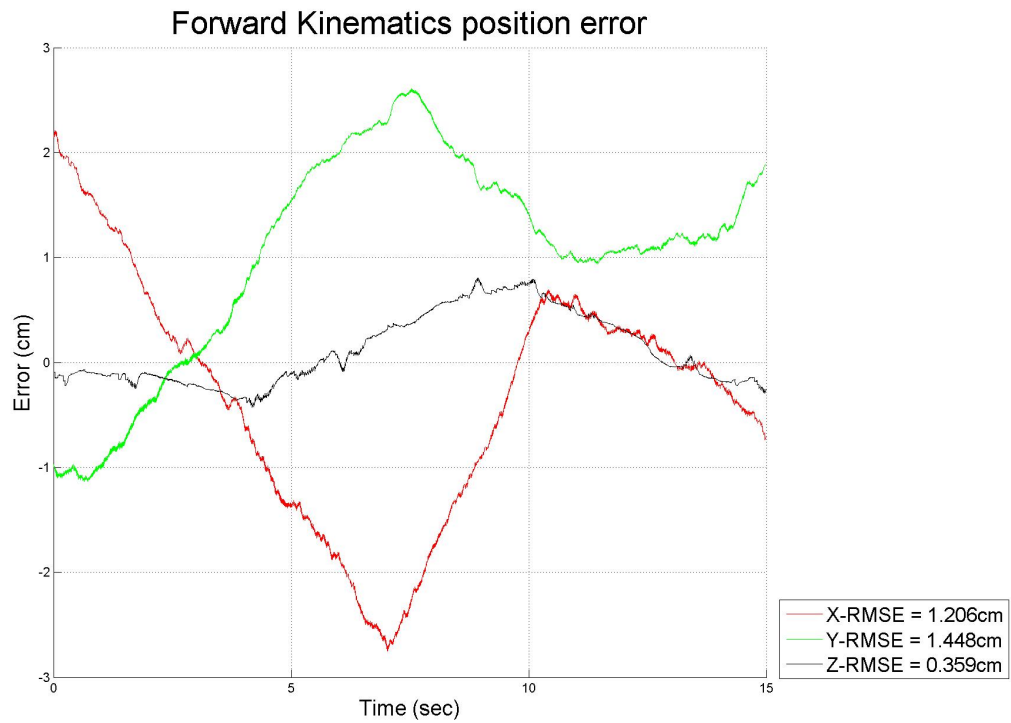


Figure 5.23: Position error of the PEB controller with constant time delay, calculated by forward kinematics

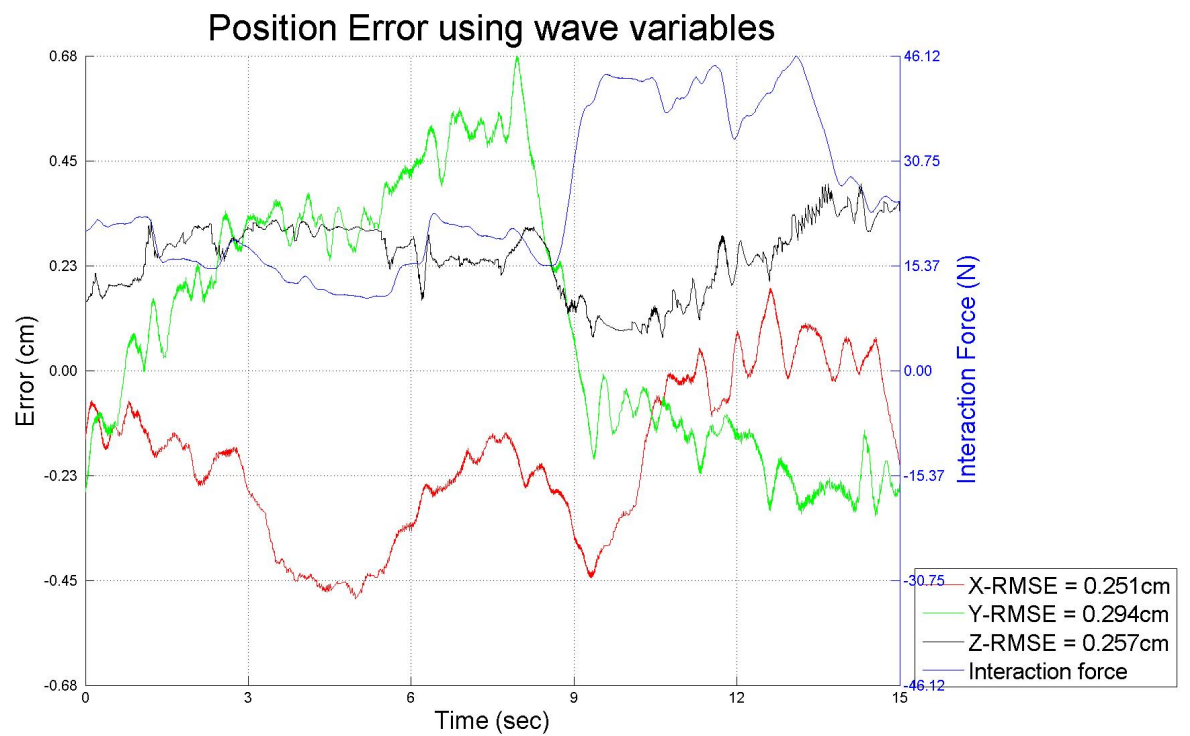


Figure 5.24: Position error using wave variables with constant time delay

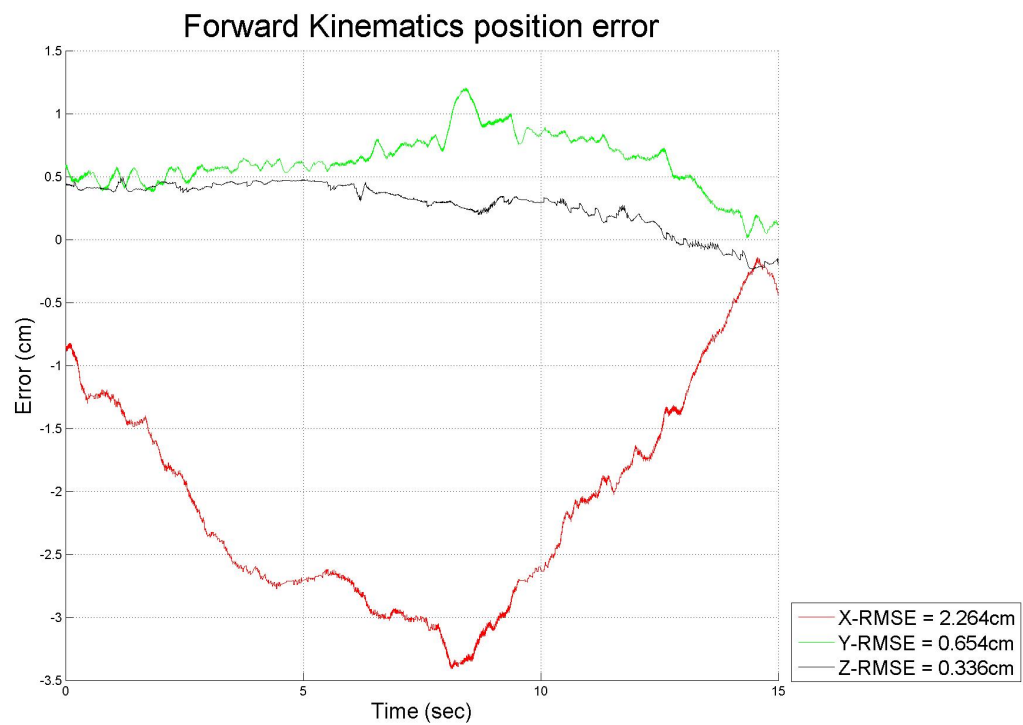


Figure 5.25: Position error using wave variables with constant time delay, calculated by forward kinematics

## 5.6 Problems faced when running the experiments

1. The accuracy of the haptic device Jacobian calculation was affected by the choice of units used for the length of the links.
2. The PEB experiments were, at times, very ‘bouncy’, especially when the CRS manipulator was close to its maximum reach.
3. When the system is first turned ON the haptic manipulator is at its ready position but that ready position is outside its dextrous workspace. It is a problem for the forward kinematics. The Jacobian will also have a problem since the haptic device needs to cross the dextrous workspace boundary surface, [27], which includes terms that approach infinity.
4. The system seems to remain stable even in the presence of time delays and the haptic device’s motors are often operating around or at their maximum continuous current.

The author dealt with these issues in the following way or the following solutions should be considered.

1. It was observed that using *cm* or *mm* would yield different Jacobians while *m* would give a faulty Jacobian. When the haptic device was designed, it was designed to optimize the link lengths in units of *cm*, [34]. It was determined that units of *cm* should always be used for accurate results.
2. The experiments were originally performed with fast filters, maximum group delay of *2ms*, to ensure synchronization with the Jacobian. The final implementation included a second sampling frequency that allowed the Jacobian to be updated less frequently. This made it possible to use slower filters, maximum group delay of *25ms*, and to remove more noise.

3. This issue was resolved by simulating an extra ON single throw switch on the haptic device using the push-button already located on the end-effector. Using a C code with static variables, the program determines the first time the operator releases the push-button. This allows the operator to switch high the ‘ON’ block in Simulink without turning ON the bilateral teleoperation. Once all procedures have been done (removing the bias of the force torque acquisition unit, limping the robot) then the operator can grasp the end-effector, press the push-button, bring it within the dextrous workspace and, only when he releases the push-button will the slave manipulator start to follow the movement of the master manipulator. After this initialization, the push-button can be used for clutching.
4. The capabilities of the human hand are analyzed by [26] through experiments and requirements for haptic devices were determined. The twin pantograph originally designed by [26], was built to meet these requirements and is capable of producing forces shown in Table 5.1 as ‘Original twin pantograph’. It is clear that the haptic device used here, ‘Actual twin pantograph’ in Table 5.1, could provide accurate feedback for a greater range of forces if it had stronger motors or a transmission box attached to its current motors. The electronics used to drive the motors could also be changed in order to provide peak capabilities.

		Force axis		
		<i>i</i>	<i>j</i>	<i>k</i>
Original twin pantograph	continuous	5.0 <i>N</i>	3.3 <i>N</i>	4.1 <i>N</i>
	peak	48 <i>N</i>	21 <i>N</i>	40 <i>N</i>
Actual twin pantograph	continuous	2.25 <i>N</i>	0.5 <i>N</i>	2.6 <i>N</i>
	peak	2.25 <i>N</i>	0.5 <i>N</i>	2.6 <i>N</i>

Table 5.1: Haptic device force capabilities

# Chapter 6

## Conclusions and future work

### 6.1 Conclusion

The goal of this thesis was to implement *Bilateral Teleoperation* using an in-house built parallel robot and the industrial serial manipulator, and to provide some insight on the effects of communication time delays. The forward kinematics and the Jacobian calculation are derived and implemented for the parallel robot (twin pantograph). The forward kinematics, inverse kinematics and Jacobian calculations are also derived and implemented for the CRS manipulator. Experiments were performed using the KFB scheme, the PEB scheme and wave variables approach. Our experiments provided accurate tracking on all three directions and all three orientations. The force feedback, obtained with the force/torque sensor, proved to be indispensable information for the operator. The response of the Kinesthetic Force Based (KFB) scheme was demonstrated to be non-passive. The destabilizing effect of constant time delay was observed to be less significant than anticipated. The inaccurate force feedback of the Position Error Based (PEB) scheme demonstrated the need for better control schemes in order to guarantee stability while providing telepresence. Experiments that included the transmission of wave variables demonstrated their capability to minimize the effect of time delay and the need for well tuned impedance matching of the waves.



## 6.2 Status and future work

A Kinesthetic Force Based (KFB) controller, a Position Error Based (PEB) controller and wave variable transmission are designed and implemented successfully in real-time. All the needed hardware are available for the implementation to take place over a network. The manipulators could be controlled with a different control scheme such as with the 4-channel architecture in order to compare its telepresence versus the KFB scheme, the PEB scheme and the wave variables approach. For now, this system is a great platform to study the effects of time delay on teleoperation and the accuracy of different force feedback methods.

# Bibliography

- [1] J. Vertut and P. Coiffet, *Robot Technology*. Englewood Cliffs, N.J.: Prentice-Hall, 1983, vol. 3B. Teleoperation and robotics : applications and technology.
- [2] G. H. Günter Niemeyer, Carsten Preusche, *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, 2008, vol. Telerobotics, no. 31.
- [3] R. C. Goertz, “Fundamentals of general-purpose remote manipulators,” *Nucleonics*, vol. 10, no. 11, pp. 36–42, 1952.
- [4] R. J. Anderson and M. W. Spong, “Bilateral control of teleoperators with time delay,” *IEEE Trans. Autom. Control*, vol. 34, no. 5, pp. 494–501, 1989.
- [5] S. Munir and W. J. Book, “Internet-based teleoperation using wave variables with prediction,” *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 2, pp. 124–133, 2002.
- [6] S. Stramigioli, “From differentiable manifolds to interactive robot control,” Ph.D. dissertation, Delft University of Technology, Delft, Netherlands, 1998.
- [7] G. Niemeyer, “Using wave variables in time delayed force reflecting teleoperation,” Ph.D. dissertation, Massachusetts Institute of Technology, Massachusetts, United States, 1996.
- [8] M. Minsky, “Telepresence: How the development of teleoperators would have immense benefits in the areas of energy, productivity, and space exploration,” *OMNI magazine*, 1980.

- 
- [9] W. R. Ferrell and T. B. Sheridan, "Supervisory control of remote manipulation," *IEEE Spectr.*, vol. 4, no. 10, pp. 81–88, 1967.
- [10] A. Aziminejad, M. Tavakoli, R. Patel, and M. Moallem, "Stability and performance in delayed bilateral teleoperation: Theory and experiments," *Control Engineering Practice*, vol. 16, no. 11, pp. 1329 – 1343, 2008.
- [11] J. E. Colgate and N. Hogan, "Robust control of dynamically interacting systems," *INT. J. Control*, vol. 48, no. 1, pp. 65–88, 1988.
- [12] M. Spong and M. Vidyasagar, *Robot Dynamics and Control*. New York: John Wiley & Sons, 1989.
- [13] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035 – 2057, 2006.
- [14] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Trans. Robot. Autom.*, vol. 9, no. 5, pp. 624–637, 1993.
- [15] G. Niemeyer and J.-J. E. Slotine, "Stable adaptive teleoperation," *IEEE J. Ocean. Eng.*, vol. 16, no. 1, pp. 152–162, 1991.
- [16] M. Alise, R. G. Roberts, D. W. Repperger, C. A. Moore, and S. Tosunoglu, "On extending the wave variable method to multiple-dof teleoperation systems," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 1, pp. 55–63, 2009.
- [17] N. A. Tanner and G. Niemeyer, "Practical limitations of wave variable controllers in teleoperation," in *Proc. IEEE Conf. Robotics, Automation and Mechatronics*, vol. 1, 2004, pp. 25–30.
- [18] S. Stramigioli, A. van der Schaft, B. Maschke, and C. Melchiorri, "Geometric scattering in robotic telemanipulation," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 588–596, 2002.
- [19] C. Secchi, S. Stramigioli, and C. Fantuzzi, "Transparency in port-hamiltonian-based telemanipulation," *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 903–910, 2008.

- 
- [20] —, “Position drift compensation in port-hamiltonian based telemanipulation,” in *Proc. IEEE/RSJ Int Intelligent Robots and Systems Conf*, 2006, pp. 4211–4216.
- [21] —, “Variable delay in scaled port-hamiltonian telemanipulation,” *Elsevier Science Ltd.*, vol. 18, pp. 357–363, 2008.
- [22] M. Dalsmo and A. van der Schaft, “On representations and integrability of mathematical structures in energy-conserving physical systems,” *SIAM Journal of Control and Optimization*, vol. 37, pp. 54–94, 1999.
- [23] G. Niemeyer and J.-J. E. Slotine, “Towards force-reflecting teleoperation over the internet,” in *Proc. IEEE Int Robotics and Automation Conf*, vol. 3, 1998, pp. 1909–1915.
- [24] P. Berestesky, N. Chopra, and M. W. Spong, “Discrete time passivity in bilateral teleoperation over the internet,” in *Proc. IEEE Int. Conf. Robotics and Automation ICRA '04*, vol. 5, 2004, pp. 4557–4564.
- [25] J. J. Craig, *Introduction to Robotics: Mechanics & Control*. Addison-Wesley Publishing Compagny, 1986.
- [26] L. J. Stocco, “Robot design optimization with haptic interface applications,” Ph.D. dissertation, University of British Columbia, 1999.
- [27] S. A. Ali, “Design of a 6-dof haptic interface for robotic teleoperation,” Master’s thesis, Lakehead University, 2008.
- [28] J.-P. Merlet, *Parallel Robots*. Netherlands: Springer, 2006.
- [29] E. W. Weisstein. Moore-penrose matrix inverse. MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/Moore-PenroseMatrixInverse.html>.
- [30] D. H. Bailey and H. R. P. Ferguson, “A strassen-newton algorithm for high-speed parallelizable matrix inversion,” in *Proc. [Vol.1] Supercomputing '88*, 1988, pp. 419–424.
- [31] Quanser, “Wincon 5.0, user’s guide,” Version 1.1, March 2005.

- 
- [32] P. A. Millman, M. Stanley, and J. E. Colgate, “Design of a high performance haptic interface to virtual environments,” in *Proc. IEEE Virtual Reality Annual Int. Symp.*, 1993, pp. 216–222.
  - [33] A. Aziminejad, M. Tavakoli, R. V. Patel, and M. Moallem, “Transparent time-delayed bilateral teleoperation using wave variables,” *IEEE Trans. Contr. Syst. Technol.*, vol. 16, no. 3, pp. 548–555, 2008.
  - [34] L. J. Stocco, S. E. Salcudean, and F. Sassani, “Optimal kinematic design of a haptic pen,” *J. Microelectromech. Syst.*, vol. 6, no. 3, pp. 210–220, 2001.
  - [35] Quanser, “Wincon 5.0, intallation guide,” Build: 31, January 2007.

# Appendix A

## Bilateral Teleoperation using WinCon

Implementations were made possible thanks to WinCon software. WinCon allows for real-time data acquisition and control with the well-known Simulink interface. Simulink is a simulation software part of Matlab but has the capability to be used in real time with the Real Time Workshop.

After a controller is designed, the user needs to build the C++ code that will allow Simulink to be in control of the data acquisition and control boards (DACB). To build the C++ code the user needs to have the appropriate version of Microsoft Visual C++ (MSVC) installed. For a complete list of software and version numbers installed, see table A.1 and for the mandatory installation order and installation settings see [35].

Required Software	Version #
Windows 2000 pro	Service pack 4
Microsoft Visual Studio 6.0	Service pack 5
Matlab 6.1	Revision 14, Service pack 1
Real-time Workshop	6.1
WinCon	Version 5.0
RTX Real-time	Version 6.1

Table A.1: Installed software and version numbers

## A.1 Single computer set-up

Once the C++ code is compiled, it opens two functions of WinCon; a server and a client. The server window communicates directly with Simulink with the help of the Real-time workshop (RTW). Figure A.1 demonstrates this type of set-up. A key factor to note is that the WinCon server is not operating under the Windows environment but rather under its own pseudo-operating system. This difference disallows the use of dynamic link libraries (DLL) and of many of the more advanced Matlab functions. Therefore, the greatest tool left to the designer is to write his own C-code by using the Simulink S-function or S-function builder blocks. The second function, WinCon Client, can be located anywhere on a local area network, the internet or on the same computer.

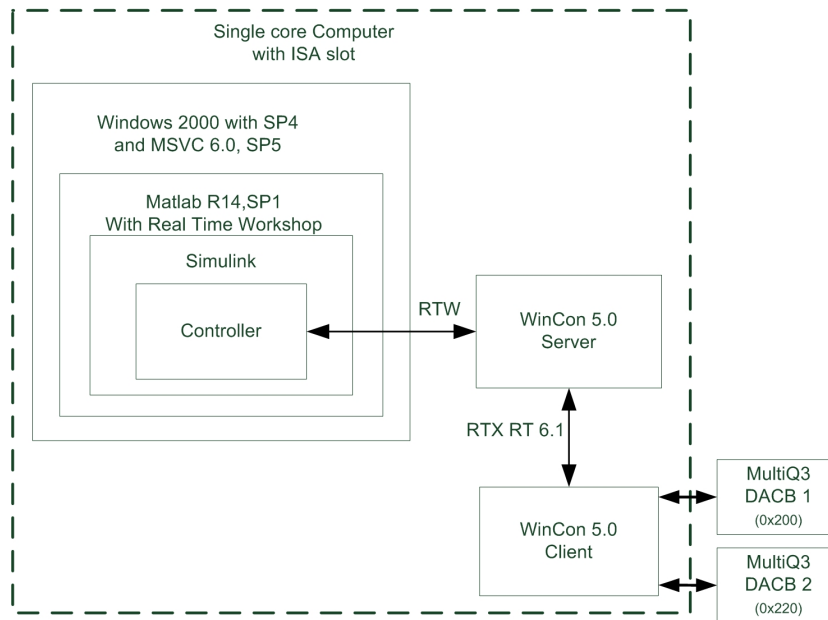


Figure A.1: Software Interaction of WinCon

Assuming WinCon Client is on the same computer for now, signifies that once Matlab has compiled the C++ code, the WinCon code (.wcl) is downloaded from the server to the client. This connection is done with the RTX software. Regardless of whether the client is internal or external the communication from the server to the client is done with the Transmission Control

Protocol and the Internet Protocol (TCP/IP) protocol. The client is in direct link with the data acquisition boards (Quancer MultiQ 3) accessed according to their ISO slot address. With the RTX software the set-up can be augmented to any number of client machines running many different controllers, see [31] for in depth examples.

## A.2 Extending WinCon for Bilateral Teleoperation

One of the many possible arrangements will be discussed here, which is, to the author's knowledge, the best way to implement bilateral teleoperation over a network using the current hardware. First, the master computer needs to have the full WinCon (Server, Client, Controls) installed, while the slave computer should ONLY have Client and Controls installed.

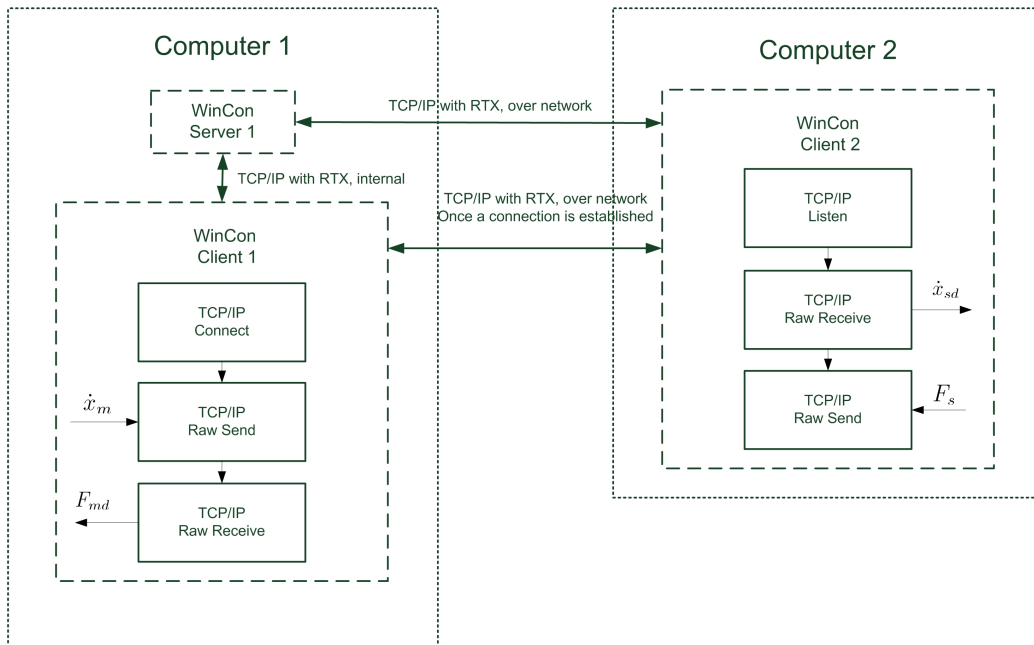


Figure A.2: WinCon used for Bilateral Teleoperation



Each manipulator needs to have its own Simulink controller with some TCP/IP blocks. We will assign the task of plotting graphs and streaming of the IP camera to the most powerful computer, the Pentium 4. Therefore the slave, or CRS, controller is demonstrated under “Computer 2” in Figure A.2. This figure demonstrates that for a PEB controller the velocity and force need to be connected to different TCP/IP blocks for appropriate communications. This is done using 3 blocks. First, the slave controller has to be running *before* the master controller is turned ON. This allows the TCP/IP listen block to start listening for anybody trying to connect to this controller. Once a connection is established, the output of the TCP/IP listen block will distribute the required information for communications, named ‘dev’. The order of the blocks that ‘dev’ connects to decides the order of the communications. Hence connecting the ‘dev’ output of the TCP/IP listen to the input of the TCP/IP Raw Receive and its output to the Raw Send allows the controller to focus on controlling the position of the slave and thereafter transmit the interaction forces.

On the master controller, we need to have the TCP/IP connect block that will initiate a communication with the IP address provided and, if it is listening, connect to it. Once again, we find a send and receive block but in reverse order such that it will match the blocks on the slave controller. Figure A.2 demonstrates the interconnectedness of this set-up.

### A.2.1 Choice of Transmission Blocks

Although the control Engineer has the option of using simple Send and Receive blocks with built in buffering and order correction, it is recommended to use the raw data blocks. This way the engineer can implement his own methods, example [18, 23], and is still able to demonstrate the passivity of the buffering method. It is the authors believe that such extension is possible with the current hardware and only requires a second RTX licence. In addition, RTX is able to use a network interface card (NIC) installed under Windows but the author recommends the addition of a separate, dedicated, NIC that can operate with less time delay. This dedicated NIC set-up needs to be installed in a special matter but can be used directly from the WinCon Server or Client hence removing the communication from the WinCon to Windows down to the NIC.

# Appendix B

## Hardware modifications made to the system

### B.1 Modifications to the emergency stop button

The original set-up included two emergency stop (e-stop) buttons, one for each manipulator. This meant that the operator could only stop one manipulator at a time given that one hand is always holding the end-effector of the haptic device. The A645 control box for the CRS manipulator made it possible to simplify things by simply having one e-stop. By using the System Input/Output (SYSIO) port, we added a normally closed single pole single throw (SPST) contact between pin 19 and 20. This contact was included in the housing with the mushroom head e-stop of the haptic device e-stop, Figure B.1.

### B.2 Modifications to the motor control boards

The author noticed a discrepancy between the waist's motor discussed in [27] and the physical motors. The motors were thought to be Maxon F2260.890 but actually were Maxon F2260.883. These motors have a torque constant of  $78.3mNm/A$  as opposed to  $250mNm/A$ . We actually needed to drive the motors with 3.2 times more current to reproduce the torque. This meant changing the maximum current output of the LMD18200 from  $1A$  to  $3A$ . In

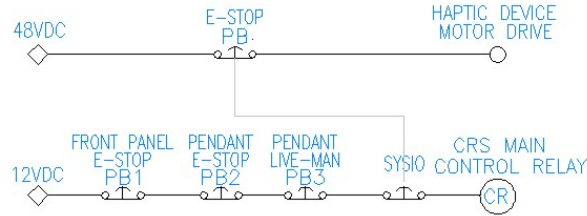


Figure B.1: The emergency stop circuit

order to minimize the changes, only three resistors were changed in order to change some voltage divider ratios. From Figure B.2, we first changed R1 such that  $R1:R2=1:1$  in order to change the peak current of the LMD18200. Secondly, we changed R11 and R12 from the original 2:1 to 3:2 in order to get a higher duty-cycle PWM signal. These changes allowed to drive each waist motor with  $3A$  but created a heating problem. The LMD18200 were now capable of producing  $25W$  of heat. The original heat sinks were only capable of  $4W$  dissipation at  $70^{\circ}C$  and would eventually fail. Given the small space that was provided for heat sinks it was decided to swap the heat sinks for the Wakefiel 401k without mounting holes. This heat sink has a greater thermal dissipation than we need,  $30W$  dissipation at  $80^{\circ}C$ . This gives the operator plenty of confidence and removes any worry of frying components if high currents are demanded for extended periods.

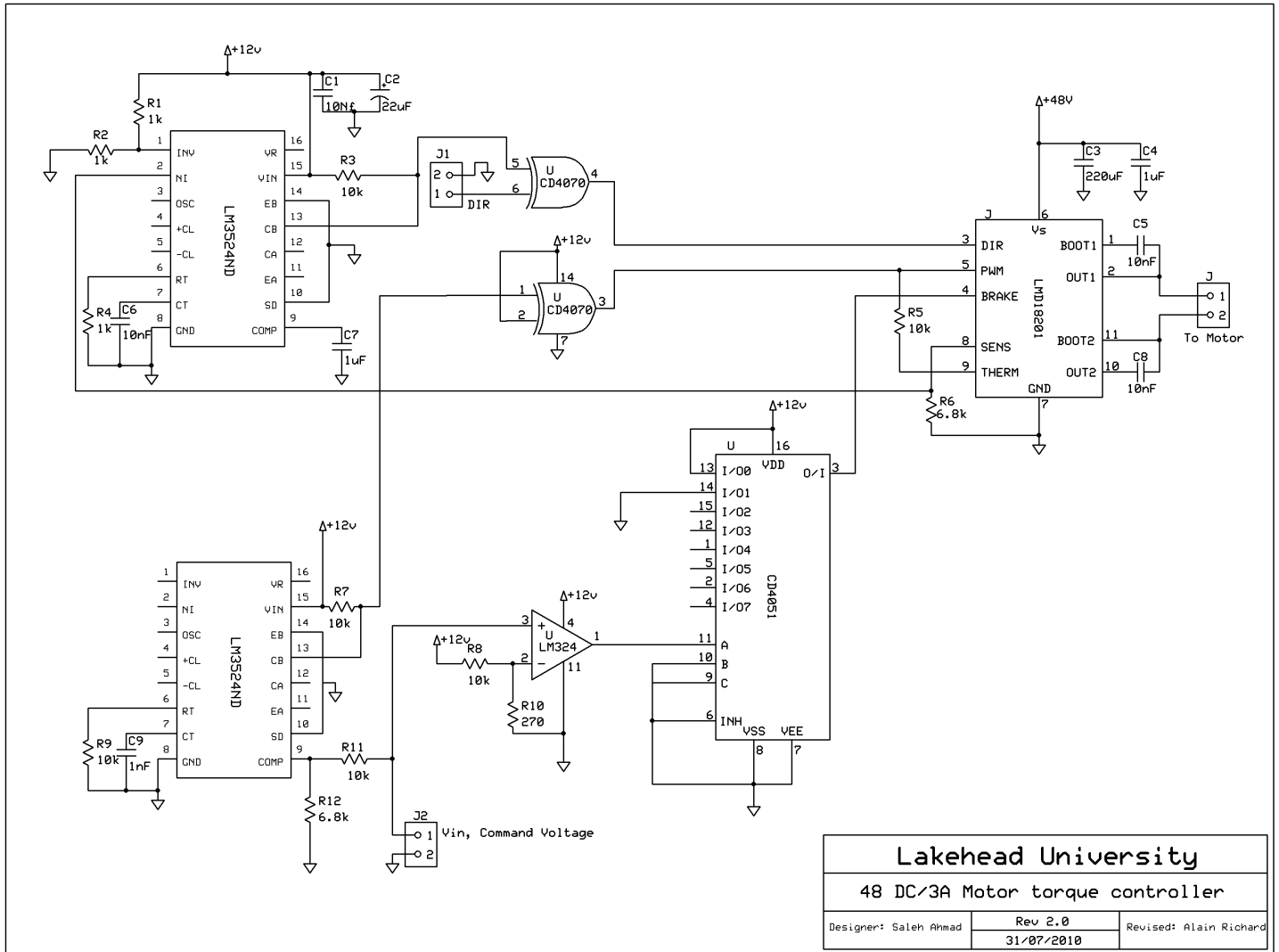


Figure B.2: Modified 90 watt motor drive