# REACTION TORQUE OBSERVER BASED FORCE FEEDBACK GRIPPER WITH FORCE LOCK

Rathnayake Mudiyanselage Maheshi Ruwanthika

(148032B)



Degree of Master of Science

Department of Electrical Engineering

University of Moratuwa

Sri Lanka

September 2015

# REACTION TORQUE OBSERVER BASED FORCE FEEDBACK GRIPPER WITH FORCE LOCK

Rathnayake Mudiyanselage Maheshi Ruwanthika

(148032B)



Thesis submitted in partial fulfillment of the requirements for the degree Master of Science

Department of Electrical Engineering

University of Moratuwa

Sri Lanka

September 2015

**DECLARATION**

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works.

Signature: Date:

Rathnayake Mudiyanselage Maheshi Ruwanthika



I endorse the declaration by the candidate.

Dr. A. M. Harsha S. Abeykoon

**ACKNOWLEDGEMENTS**

This thesis is a partial requirement for the completion of Master of Science degree in University of Moratuwa. This text is a compilation of research work that has been carried out at the Control and Robotics Laboratory, Electrical Engineering Department, Faculty of Engineering, University of Moratuwa, SriLanka.

This research was performed under the supervision of **Dr. A. M. Harsha S. Abeykoon**. I would like to show him my gratitude for his continuous guidance and support to fulfill this task.

Many thanks and gratitude to **Prof. Sudath Rohan Munasinghe,** Electronic and Telecommunication Engineering Department and **Dr. W. D. Asanka S. Rodrigo,** Electrical Engineering Department, Faculty of Engineering, University of Moratuwa. As the review panel, their valuable comments, encouragements and discussions in the progress review meetings helped me to achieve this task.

I deeply thank to my postgraduate colleagues Mr. W. A. S. P. Abeysiriwardhana, Mr. M.A. V. J. Muthugala and Mrs. M. P. Menikdiwela at the Department of Electrical Engineering, Faculty of Engineering, University of Moratuwa for their helping hand during difficult tasks.



All members of Electrical Engineering Department, University of Moratuwa are also gratefully acknowledged. They constantly helped me during my research activity.

I would like to express my sincere gratitude to all those who have assisted me in my life specially my parents Mr. R. M. Gunawardhana and Mrs. L. Karunawathie. Last but not least, I should thank Dr. A. M. Harsha S. Abeykoon for his guidance.

Finally, I would like to thank everybody who was important to the successful realization of thesis, as well as expressing my apology that I could not mention personally one by one.

Rathnayake Mudiyanselage Maheshi Ruwanthika.
University of Moratuwa
September 2015

## ABSTRACT

Haptic information is a bilateral information of the law of action and reaction. Most of the researchers have done experiments on bilateral teleoperation but still they haven't given much attention on force lock during bilateral teleoperation even though it's important in the latest technology. This research proposes an enhanced force limiting gripper to avoid object deformation when it is in contact with the slave due to excessive forces imposed by the master operator in bilateral control. The force lock protects the gripped object on the slave and the attainment of the force limit is notified to the master operator via a small vibration. Master operator is expected to experience a spring effect if the operator presses his leaver towards the force increasing direction. The continuous copying of the slave position as the reference to the virtual spring controller's spring equilibrium point allows smooth releasing. In addition to vibration the loss of reaction force coming from the slave environment could also be sensed by the master operator. Releasing logic of force lock can be determined by the operator. In this proposed system sensor-less sensing is used. Disturbance Observer (DOB) is used to estimate disturbances and Reaction Torque Observer (RTOB) estimates reaction forces.

The reaction from real world includes not only position and force information but also environmental impedance. The griped object impedance variation with the motion parameters of the actuator should also be analyzed. In this research, the behavior of environmental object impedance has been studied with the changes of the different motion parameters of the actuator, like applied force, velocity, position and depth on the object 3D space. This idea is exemplified using a rubber balloon and a rubber sponge which is often modeled using simple linear equations.

The proposed system is tested using hardware setup and results prove the applicability of RTOB and DOB for force feedback gripper for haptic teleoperation. The results of variable impedance model as function of its motion parameters suggests nonlinear impedance variations against the motion parameters of the environmental object.




**Key words**; Haptic teleoperation, Disturbance Observer, Reaction Torque Observer, Force lock, variable impedance model.

**TABLE OF CONTENTS**

**TABLE OF FIGURES**

## TABLE OF TABLES



## LIST OF ABBREVIATIONS

| Abbreviation | Description |
|---|---|
| DOB | Disturbance Observer |
| RTOB | Reaction Torque Observer |
| DOF | Degree of Freedom |
| FSR | Force Sensing Resister |
| SPI | Serial Peripheral Interface |
| QEI | Quadrature Encoder Interface |
| ppr | pulses per revolution |

# CHAPTER 1

# 1 INTRODUCTION

Human have five sensory organs which are eye, ear, nose, tongue and skin sensitive to sight, sound, smell, taste and touch respectively. All sensory organs except skin act unilaterally. Those identify (act) the stimulus from the environment and make aware the brain about the sense. But human skin identifies the touch while reaction is exerting on the touched object. Therefore human skin is bilateral.

The technology has been developed to store sound and visual sensors of the human. It also can be reproduced in a remote place. The sense of the nose (smell) and tongue (taste) cannot normally be transmitted nor stored in an electronic means successfully. The fifth sense, the touch can be transmitted and reproduced using the bilateral control concept. The target of bilateral control is the transmission of sense of touch information by electronic means from a remote location. The sense of touch is termed as “haptic”.

## 1.1 Haptic Teleoperation



The target of haptic teleoperation is the transmission of haptic information by electronic means from a remote location. The term “tele” is a Greek word to denote “at a distance”. The term “teleopertation” means operating a machine at a distance. Teleoperation is used to combat the distance, to improve accessibility in hazardous environments [1]. Technical advancements in teleoperation are useful in ultimate environments, such as disaster site, undersea, space exploration, mine excavations, atomic furnaces, etc. [2]. If human operator wants to improve a manual task execution, such as robotic surgery again teleoperation is useful [3]. A teleoperation system consists of five interacted elements:

- a human operator
- haptic interface or master device
- remotely located teleoperator
- remote surrounding environment
- a communication channel

A human operator issues commands through a haptic interface or master actuator to remotely located telioperator or salve actuator which is interacting with its

surrounding environment via a communication channel [4]. In haptic teleoperation the reaction from the remote environment is transmitted to human operator via communication channel so that he senses remote environment. Figure 1.1 shows the concept of haptic teleoperation.

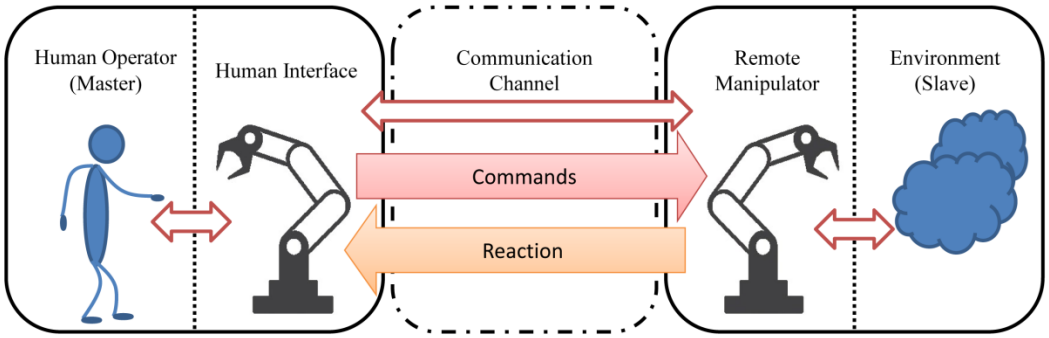



Figure 1.1 Concept of haptic teleoperation

The bilateral control system is a popular and successful concept for transmission of haptic perception.

## 1.2 Bilateral Control

When a person wants to feel an object by touch, he directly touches the object. If the object is to be touched is dangerous or it is placed far away, we need a device to prevent the damage or combat the distance. Bilateral control system provides the solution. The main idea of the bilateral control is sensing an environment remotely while controlling the remote manipulator based on the operator's intension. The word "bilateral" means controlling both ways. The operator who is giving commands is named as the master and the follower which is placed on the remote environment is named as the slave. Figure 1.2 shows the concept of bilateral control.



The slave side is controlled through position and force information from the master side ($X_m, F_m$), and the master side is controlled by position and force information from the slave side ($X_s, F_s$) [5]. The operator should feel as if he is physically present at the remote environment ("telepresence").

If bilateral control is a successful solution for remote force sensing it should follow the natural law of action and reaction. If scaling has not been used between master and slave, both master and slave system track position of each other while reaction

coming from the slave environment is equal in magnitude and opposite in direction to that of master. Figure 1.3 shows the bilateral teleoperation system.

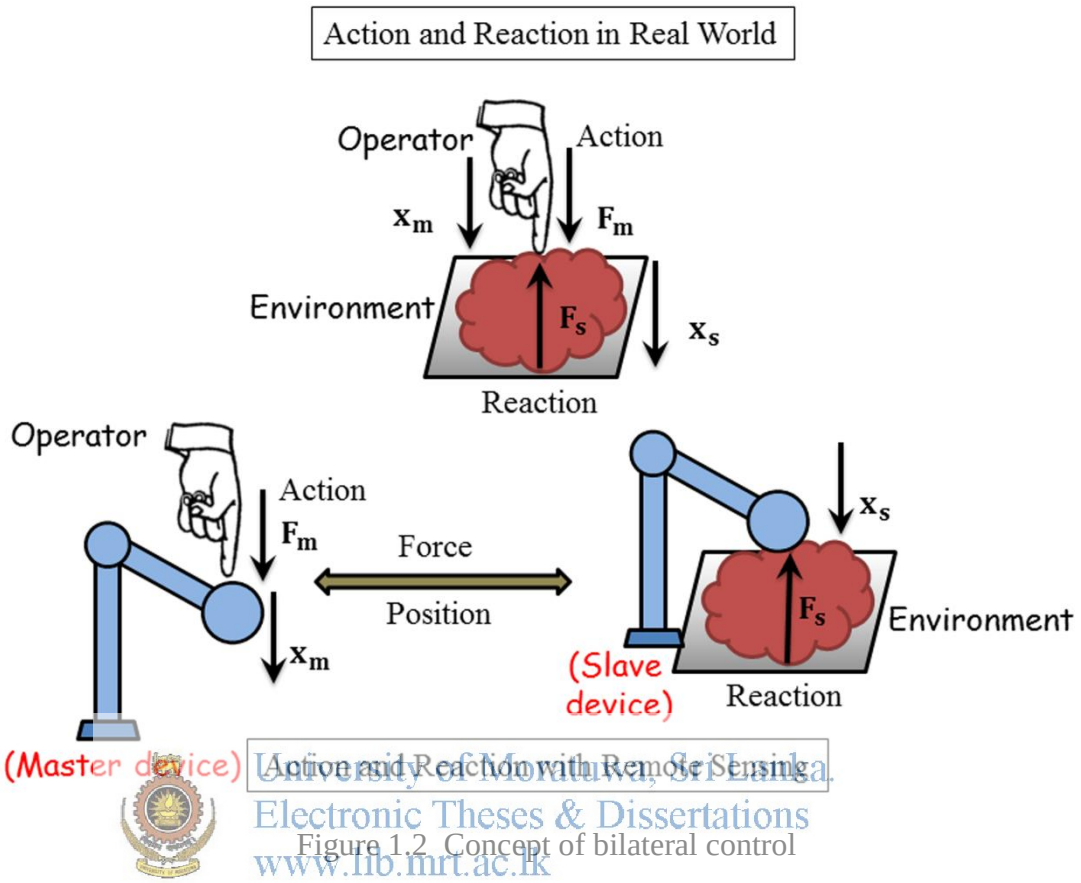





Figure 1.2 Concept of bilateral control

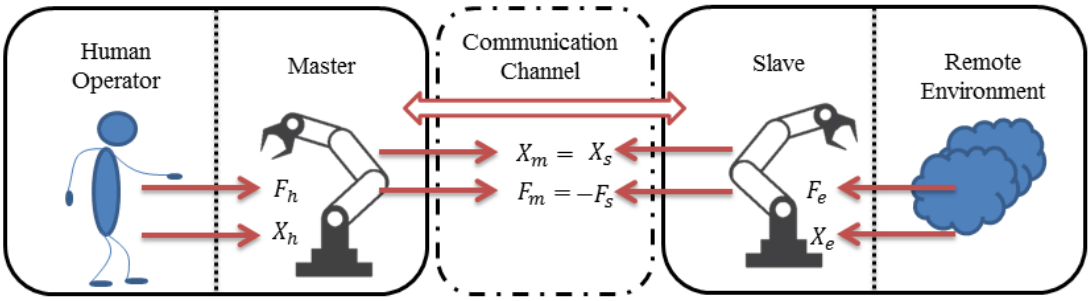



Figure 1.3 Bilateral teleoperation system

Most of the teleoperated systems consist of robotic arms with grippers as their actuator. Gripper interacts with the remote object. Selection of grippers depends on the task that it is used, environment, robot arm configurations, and control conditions [6]. Complex gripper design requires complex control mechanism and simple gripper design may cause poor grasping of the working object.

Every object has a predefined force limit which can bear up without damaging it. When object are handling with grippers the force exerted through gripper should be lower than the force limit of the gripped object. Therefore there are applications of force limiting grippers for teleoperation.

### 1.3 Force Limiting Grippers

Force limiting grippers are used to ensure that the gripped object will not be damaged by the excessive force applied and to maintain static grip force sufficient to firmly hold the object. Throughout the past few decades researches on compliance force grippers have been conducted [7 -12]. The compliance force has been achieved through gripper mechanical design [12], gripper material [9],[11], and using various types of force sensors like pressure sensors [8], strain gauges [11], Force Sensing Resister (FSR) [13], and piezoelectric sensors [14] etc. The researches mentioned above are for unilateral robot arms. Therefore there is no grip force feedback to the operator.

Grip force control during teleoperation is important in applications like robot – assisted minimally invasive surgery (MIS) [15],[20] where surgeon perform operation at a distance through instruments inserted into the body through small incisions as well as in hazardous object handling and pick and place in industry. It is important to maintain compliance force continuously on slave to protect gripped object and to avoid slipping from gripper arms while force feedback to the master operator allows force sensation from the slave environment in real time. Although researches on grip force sensing during bilateral teleoperation could be found [16-17], compliance force lock during bilateral teleoperation is yet to be tested. On those studies gripping force sensing has been achieved through kinesthetic, tactile sensors and stain gauges.



Tactile sensors based on piezoelectric materials could be used at the expense of a limited force range in sensing application and piezoelectric grippers provide limited displacement. Although flexible gripper based on inflatable rubber pockets driven by a pneumatic actuator could grasp up to 20 kg, have the disadvantages of being large, heavy and need of an air compressor.

The gripped object behavior should be analyzed with the different motion parameters of the gripper fingers because object impedance might varies depending on the impact exert on it by the gripper fingers. The environmental object should be modeled for further analysis. Researches on object modeling could be found on haptic databases for skill preservation.

### 1.4 Haptic Database

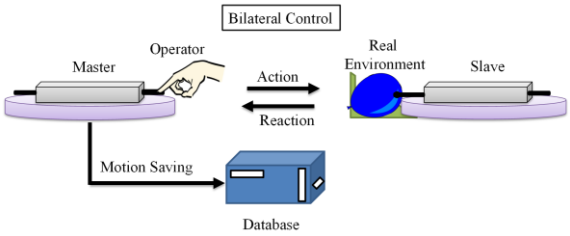
Limitations of information recording facilities of characters and/or pictures which were used in ancient time were expanded to auditory information with the invention of the radio in 18th century. Furthermore in mid-19th century, the invention of video tape recorder facilitated static and dynamic visual information storage. With the rapid development of communication technologies, internet enables transmission of text documents, sounds, and visual information worldwide. That information is captured by human auditory and visual organs of ears and eyes. Although the human body has haptic organs, the information acquired using conventional methods and visual information does not contain haptic information like force [18].

In recent years the concept of haptic information storage and reproduction has been proposed [19]. Preservation of human haptic information in a haptic database provides facility to reproduce it at any time at anyplace. Expert skills of motion reproduction capability decline with the aging of people and decrease the number of successors especially in artistic machining, artistic pottery, etc. By using haptic information robots are expected to work in production lines as well as in open environments. Further haptic database can preserve once skills on to the next generation. This is called as motion copying system. The motion copying system includes human haptic information in terms of position and force information which are recorded during motion saving mode. Recording haptic information is done by bilateral control of action and reaction [18] using motion saving system.



Bilateral controller with disturbance observer [21] gives the vivid force sensation from the real environment. Therefore real environment force and position information also should be recorded successfully for preservation and reproduction. The Force Sensation Recorder in [22] proposes the preservation and reproduction of force sensation from the real environment. The FSR abstracts the position and the

force information including the environmental information like environment impedance from the real environment based on the acceleration control architecture. This information is extracted during the abstraction mode of the force sensation recorder. The motion copying system stores the haptic information of human and FSR stores the information of the environment. Therefore at the motion loading stage of the motion copying system has virtual master and slave follows the haptic data recorded in the database whereas force sensation recorder has virtual environment and master feels the environmental model as he moves the manipulator. Figure 1.4 shows the concept of motion copying system and Figure 1.5 shows the concept of force sensation recorder.

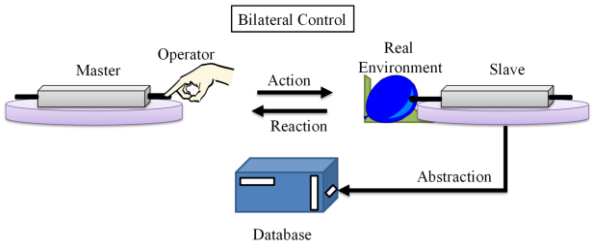



(a) Motion saving system

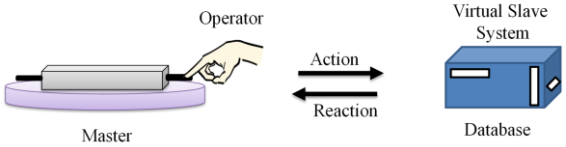



(b) Motion loading system

Figure 1.4 Motion copying system.




(c)Abstraction Mode




(d) Reproduction Mode

Figure 1.5 Force sensation recorder

## 1.5 Objectives

Force limiting teleoperated grippers which have been used in production lines, agriculture, care taking, assistant, entertainment, etc do not provide force feedback to the human operator because those are unilateral grippers. But force feedback is essential for real world haptic teleoperation because touch sense to be successfully transmitted law of action and reaction should be satisfied. The author is interesting to study the gripper operation with a force lock for haptic teleoperation using bilateral control which is based on acceleration architecture. Introduction of Disturbance Observer (DOB) and Reaction Torque Observer (RTOB) for the force lock is carried out. Since the study involve the force lock for bilateral teleoperation the transition from bilateral control to force lock and vice versa should not panic the master operator. Therefore author is aiming to achieve smooth griper operation during transition stages while maintaining the stability. The effect of communication delay during bilateral teleoperation is not addressed in this study. The gripped object impedance varies depending on the gripper finger motion parameters like compression depth, compression velocity and compression acceleration. Therefore author is also intending to study object impedance variation as a function of its motion parameters.



## 1.6 Originality

The applications of bilateral teleoperation like MIS have many examples for compliance force control [20]. But the transmission of touch sensation while grasped object is being locked without damaging it by exceeding the bearable force limit of the object has not been studied. Therefore this study proposes a novel method of force feedback gripper with a force lock to be used with bilateral teleoperation via sensorless sensing technology of DOB and RTOB. The introduced force lock in which force limit is set by the operator experience, allows operator freedom to release his hand during force lock mode. The proposed system is tested with 1 Degree of Freedom (DOF) rotary motor actuators.

The changes in gripper finger motion parameters like applied force, applied velocity and deformed depth exert different impacts on gripped object. The object should be modeled to study its behavior with motion parameters. The object models have been

studied for objects with high stiffness during construction of haptic databases in [19], [23] and [24]. Therefore behavior of environmental object impedance has not been studied during above studies. Therefore the study is further extended by author to low environmental impedance objects like balloon and rubber sponge to study environmental object model variation with different motion parameters. Previous studies have not considered property change of environmental model as actuator moves along the 3D space on environmental surface. In studies of [22] and [25] have been used sponge as an environment and modeled it as a function of stiffness and viscosity for considerable force range and they have not considered the environmental impedance behavior for different velocities and deformed depth and surface position on the 3D environment. This study proposes an impedance model as a function of its motion parameters and highlights the importance of considering the different motion parameters of actuator on environmental impedance changes for construction of real environmental object model.

## 1.7 Thesis Organization

The structure of the dissertation is shown in Figure 1.6. A summary of each chapter is mentioned below according to the chapter sequence.



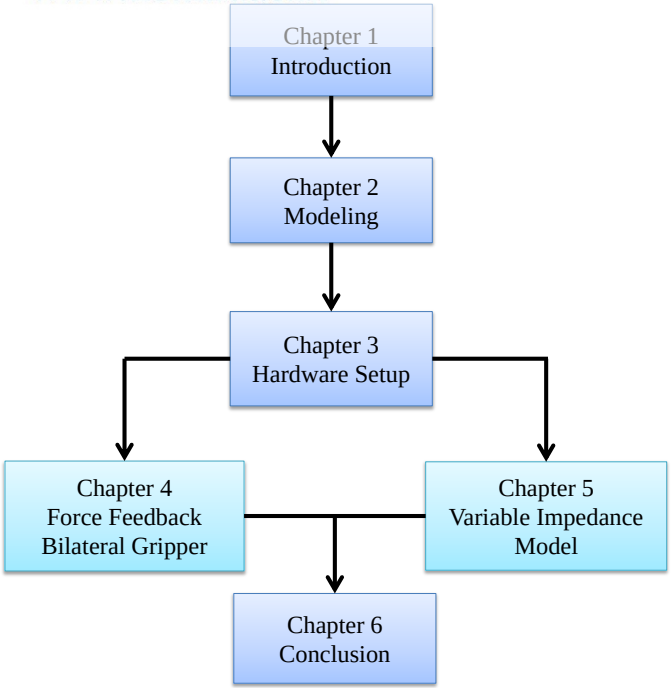



Figure 1.6 Outline of the thesis

**Chapter 1**

This chapter describes the background of the thesis, haptic teleoperation, bilateral control, force limiting grippers, available methods for force estimation, environmental object modeling for haptic databases, objectives and originality of the research.

**Chapter 2**

Modeling of basic control block diagrams which were used for chapter 4 and chapter 5 studies are presented in this chapter. It includes modeling of DC motor, Disturbance Observer and Reaction Torque Observer.

**Chapter 3**

This chapter explains about the hardware experimental system used to implement the force feedback bilateral gripper functions and the variable impedance model of the environmental object.



**Chapter 4**

The modeling of force feedback gripper which consists of bilateral control, force control, spring control, vibration notification and release function is considered in this chapter. The simulation results and experimental results are presented in order to verify the proposed novel method of grip force control.

**Chapter 5**

The environmental object model and its parameter estimation for different motion parameters are included in this chapter. The experimental results are presented in order to verify the environment impedance variation with different motion parameters of the actuator.

**Chapter 6**

The summary and the conclusion of the thesis are presented in chapter 6.

# CHAPTER 2

# 2 MODELING

## 2.1 DC Motor Modeling

The DC motor is a simple machine which converts electrical energy to the mechanical energy. Electrical equivalent model of a DC motor is shown in Figure 2.1.

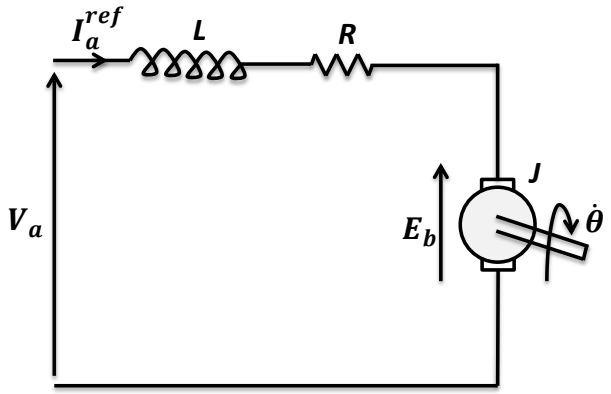



Figure 2.1 Electrical model of a DC motor.



A major fraction of the torque generated in the rotor of the motor is available to drive an external load. DC motors are widely used in numerous control applications, including robot manipulators, disk drive, machine tools, servo actuators and high torque applications etc [26]. DC motor is very popular especially in precise motion control applications, as the modeling is much simpler than all the other motors [27]. The notations used in DC motor modeling are listed below.

| | | |
|---|---|---|
| $R$ | : | armature resistance |
| $L$ | : | inductance of the armature winding |
| $V_a$ | : | supply voltage |
| $I_a^{ref}$ | : | armature current |
| $E_b$ | : | motor back emf |
| $J$ | : | motor inertia |
| $\dot{\theta}$ | : | angular velocity |
| $T_m$ | : | motor torque |
| $K_t$ | : | motor torque constant |
| $K_e$ | : | motor back emf constant |

$T_l$ : load torque

$T_f$ : static friction

$T_{int}$ : interactive torque

$T_{ext}$ : external torque

$B$ : viscous friction coefficient

$s$ : Laplace operator

$\theta$ : angular displacement

Equation (2.1) can be obtained after applying the Kirchhoff's voltage law to the DC motor model.

$$V_a = L_a \frac{dI_a^{ref}}{dt} + R_a I_a^{ref} + E_b \tag{2.1}$$

The back emf, $E_b$ is proportional to the angular velocity and relationship is represented in (2.2).

$$E_b = K_e \dot{\theta} \tag{2.2}$$

By substituting (2.2) in (2.1) it can be rewritten as in (2.3).

$$V_a = L_a \frac{dI_a}{dt} + R_a I_a + K_e \dot{\theta} \tag{2.3}$$



The torque produced by the DC motor is proportional to the armature current as shown in (2.4) assuming a constant field DC motor.

$$T_m = K_t I_a^{ref} \tag{2.4}$$

Applying (2.4) in (2.3) gives (2.5),

$$V_a = L_a \frac{d(T_m / K_t)}{dt} + R_a \frac{T_m}{K_t} + K_e \dot{\theta} \tag{2.5}$$

The dynamic equation of a rotary motor can be represented as (2.6).

$$J \frac{d\dot{\theta}}{dt} = T_m - T_l \tag{2.6}$$

The generated motor torque $T_m$ serves load torque $T_l$ and the motor acceleration. The load torque $T_l$ can be represented as a combination of reaction torques of the external mechanical load $T_{ext}$, interactive torque $T_{int}$, static friction $T_f$ and viscous friction $B\dot{\theta}$. When motor is not under acceleration major fraction of motor torque is consumed by the external load. Static friction is the main frictional element of a

motor when comparing to other frictional effects and it is effective when the motor is about to rotate. Viscous friction is always depending on the angular velocity of the rotor [28],[43]. Therefore load torque can be represented as (2.7) and motor generated torque can be modified by substituting (2.7) into (2.6) as shown in (2.8).

$$T_l = T_{ext} + T_{\text{int}} + (T_f + B\dot{\theta}) \tag{2.7}$$

$$T_m = J\frac{d\dot{\theta}}{dt} + T_{ext} + T_{\text{int}} + (T_f + B\dot{\theta}) \tag{2.8}$$

In Laplace domain equation of (2.4) is shown in (2.9) whereas (2.10) shows Laplace domain equation of (2.8) after rearranging.

$$T_m(s) = K_t I_a^{ref}(s) \tag{2.9}$$

$$\dot{\theta}(s) = \frac{T_m(s) - [T_{ext}(s) + T_{\text{int}}(s) + T_f(s) + B\dot{\theta}(s)]}{Js} \tag{2.10}$$

Equations (2.9) and (2.10) are used to develop the model of DC motor as shown in the control block diagram given in Figure 2.2. For a single motor servo control system interactive torque can be considered as zero. The combination of static friction $T_f$, viscous friction $B\dot{\theta}$, reaction torque of the external mechanical load $T_{ext}$ and interactive torque $T_{int}$ is also known as the disturbance torque of a motor [29].



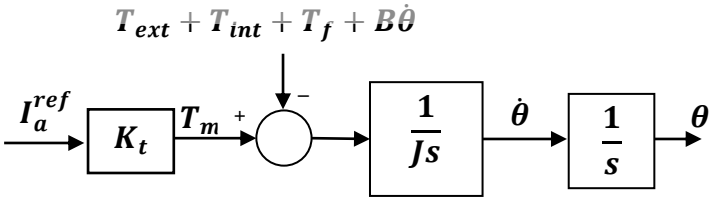



Figure 2.2 DC motor model

## 2.2 Disturbance Observer

Robotic manipulators are often subject to different types of unknown disturbances such as external loads, motor parameter changes, etc. If such disturbances are not accounted for calculations, performance of the robot may tend to degrade and may even cause the instability of control system [30]. Controlling a servomotor under unknown disturbances is a challenging problem. An approach to suppress these disturbances is to use Disturbance Observer (DOB). Due to the difficulty and cost of

direct measurement, estimation has become a popular method of measuring the quantity of the disturbance torque acting on a shaft of DC motor rather than measuring directly [31]. This estimation then may be used to compensate for the disturbance torque acting on the shaft thus improving the system's robustness to external torques and load changes. The magnitude of the disturbance can be estimated and then can be used to improve the performance of the manipulator. Estimation of the external disturbance torque is done by using DOB [32].

The notations used in addition to DC motor model which are introducing in DOB model are listed below.

| | | |
|---|---|---|
| $\ddot{\theta}$ | : | angular acceleration of the motor |
| $J_n$ | : | nominal inertial of motor |
| $\Delta J$ | : | change in motor inertia |
| $K_{tn}$ | : | nominal motor torque constant |
| $\Delta K_t$ | : | change in motor torque constant |
| $T_{dis}$ | : | disturbance torque |
| $\hat{t}_{dis}$ | : | estimated disturbance torque |
| $g_{dis}$ | : | cut-off frequency of the low pass filter |



The dynamic equation of a rotary motor which is shown in (2.6) can also be represented as (2.11).

$$J\ddot{\theta} = T_m - T_l \tag{2.11}$$

The load torque $T_l$ shown in (2.7) and generated motor torque in (2.4) simplifies (2.11) to (2.12) as shown below.

$$J\ddot{\theta} = K_t I_a^{ref} - (T_{ext} + T_{\text{int}} + T_f + B\dot{\theta}) \tag{2.12}$$

Parameters $J$, $K_t$ are subjected to variations and estimation errors. Therefore these can be re-written in terms of nominal values and variations. The inertia of the motor $J$ can be changed significantly from its nominal value with the mechanical configuration of the system. It can be represented as (2.13).

$$J = J_n + \Delta J \tag{2.13}$$

Furthermore, due to the torque pulsation, and due to the flux distribution variation by the rotor position or sometimes may be due to calculation errors the motor torque constant can also be changed from its nominal value. It can be represented as (2.14).

$$K_t = K_{tn} + \Delta K_t \tag{2.14}$$

By substituting (2.13) and (2.14) to (2.12) and rearranging (2.15) can be derived.

$$J_n\ddot{\theta} = K_{tn}I_a^{ref} - ( T_{int} + T_{ext} + T_f + B\dot{\theta} + \Delta J\ddot{\theta} - \Delta K_t I_a^{ref}) \tag{2.15}$$

The term within the parenthesis in (2.15) is called as total disturbance torque to the system $T_{dis}$.

$$\begin{aligned} T_{dis} &= T_l + \Delta J\ddot{\theta} - \Delta K_t I_a^{ref} \\ &= T_{int} + T_{ext} + \left(T_f + B\dot{\theta}\right) + (J - J_n)\ddot{\theta} + ( K_{tn} - K_t)I_a^{ref} \end{aligned} \tag{2.16}$$

Introducing $T_{dis}$ to (2.15) gives (2.17),

$$J_n\ddot{\theta} = K_{tn}I_a^{ref} - T_{dis} \tag{2.17}$$

By rearranging (2.17), $T_{dis}$ can be calculated as (2.18).

$$T_{dis} = K_{tn}I_a^{ref} - J_n\ddot{\theta} \tag{2.18}$$

When angular acceleration and motor current are known parameters, it is possible to calculate the disturbance torque $T_{dis}$ according to (2.18) from the block diagram given in Figure 2.3. This implemented arrangement is known as the disturbance observer based on acceleration architecture.



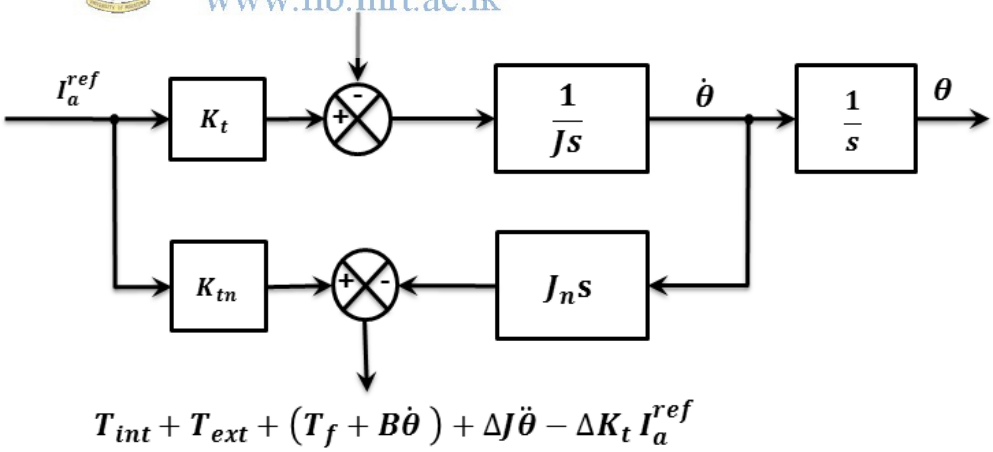



Figure 2.3 Determination of disturbance torque

However, practically the angular acceleration cannot be measured directly with available sensors. So, it is normally calculated by differentiating the angular displacement twice which is measured from the encoders. Furthermore, a low pass

filter should be used at the output of the disturbance torque calculation because of the differentiator used in estimating the angular acceleration. If first order low pass filter is used, we can express the estimated disturbance torque as in (2.19).

$$\hat{t}_{dis} = \frac{g_{dis}}{(s + g_{dis})}\{K_{tn} I_a^{ref} - J_n \ddot{\theta}\} \tag{2.19}$$

The modified block diagram after adding low pass filter is shown in Figure 2.4.

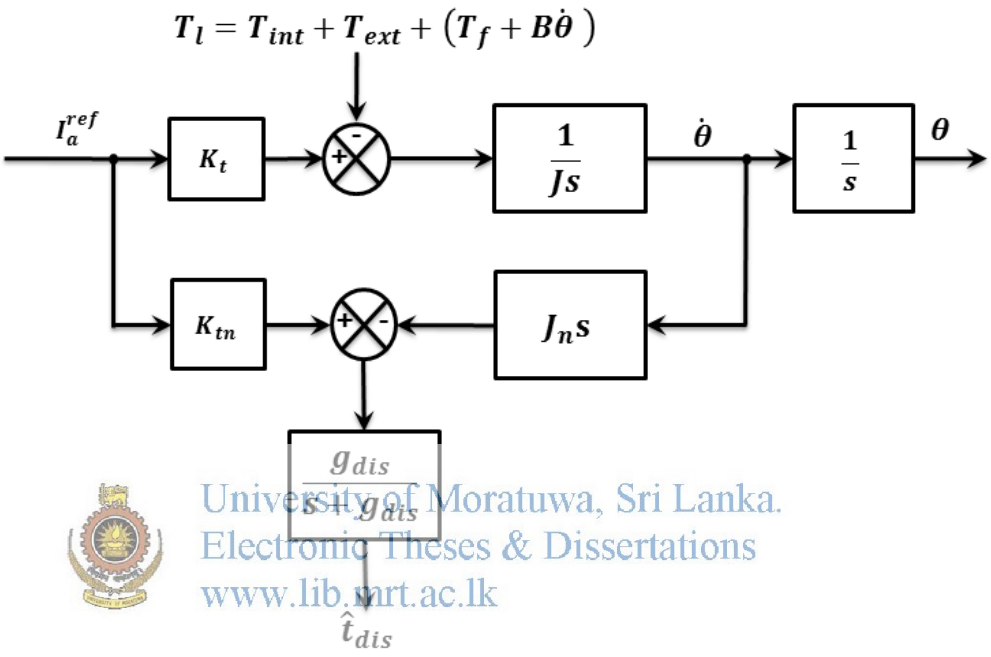




Figure 2.4 Block diagram of the disturbance observer with low pass filter

The disturbance torque is directly calculated from the acceleration of the motor and the real torque current. Disturbance observer observes the disturbance force in the system without using force sensors. It is designed such a way that it is possible to estimate the disturbance from the observer [33], [34].

The acceleration is not directly detected by ordinary sensors in Figure 2.4. The acceleration signal is calculated through the differentiation of the velocity of the encoder. A certain time delay presents in this calculation. Therefore, block diagram in Figure 2.4. can be rearranged as in Figure 2.5. as a practical way of estimating disturbance torque excluding above time dely. Then we can express the estimated disturbance torque as in (2.20).

$$\hat{t}_{dis} = \frac{g_{dis}}{s + g_{dis}}\left(K_{tn} I_a^{ref} + J_n g_{dis}\dot{\theta}\right) - J_n g_{dis}\dot{\theta} \qquad (2.20)$$

The disturbance observer is introduced to identify and to suppress the effect of disturbance torque for robust motion control.

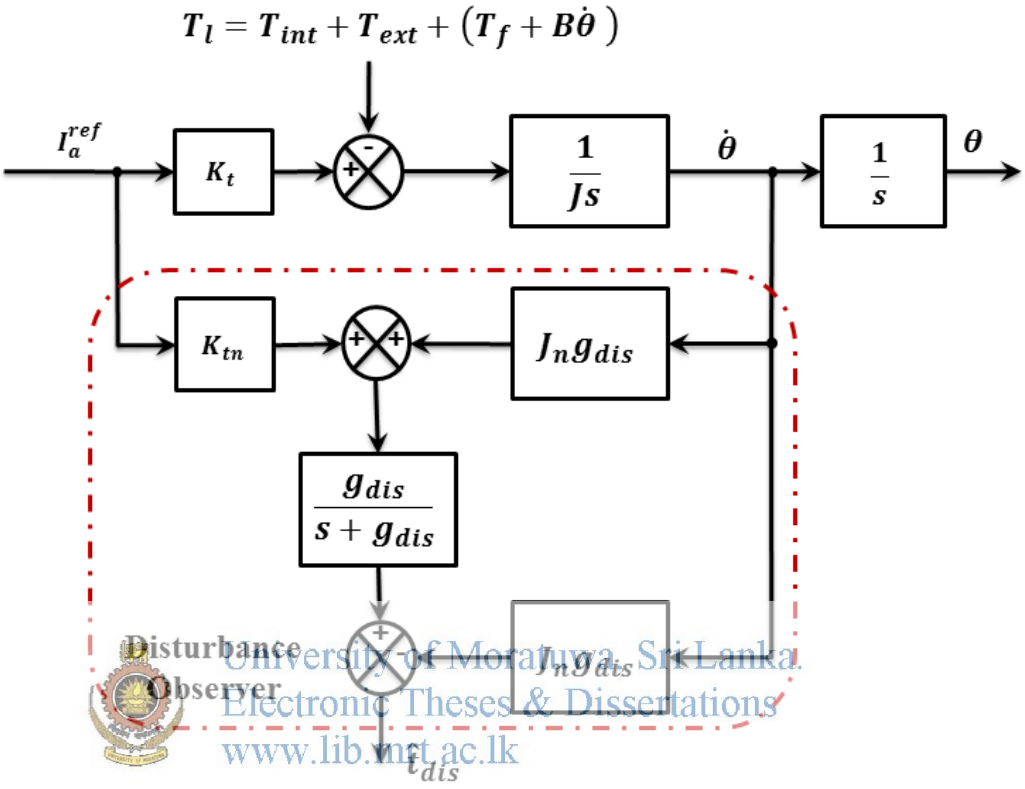




Figure 2.5 modified block diagram of DOB to exclude differentiation time delay

The disturbance observer estimates the disturbance torque on the control system. The feedback of the estimated disturbance torque compensates the low frequency component of the disturbance which is below the cut off frequency $g_{dis}$ of low pass filter. Introducing a disturbance current, it is possible to compensate for the disturbance. Disturbance observers are highly useful when the system requires higher bandwidth because most of the torque sensors have a lower bandwidth [36]. Disturbance observer is not only estimates the disturbance torque but also it can modify as a reaction torque estimator [37].

## 2.3 Reaction Torque Observer

The reaction torque observer (RTOB) behaves as a virtual force sensor. In the actual application, the DOB is furthermore modified for reaction torque estimation as

Reaction Torque Observer (RTOB). This is achieved by identifying internal parameter variation and friction component in the system previously. The calculation of estimated external torque is shown in (2.21).

$$\hat{t}_{ext} = \frac{g_{rec}}{s + g_{rec}} \left( K_{tn} I_a^{ref} + J_n g_{rec} \dot{\theta} - \begin{pmatrix} T_{\text{int}} + T_f + B\dot{\theta} + \\ \Delta J\ddot{\theta} + \Delta K_t I_a^{ref} \end{pmatrix} \right) - J_n g_{rec} \dot{\theta} \qquad (2.21)$$

Where $g_{rec}$denotes the cut-off frequency of RTOB low pass filter. The modified control block diagram of the DOB as RTOB is shown in Figure 2.6. The DOB calculates and estimates reaction torque as quickly as possible by increasing the cutoff frequency in the stability range.

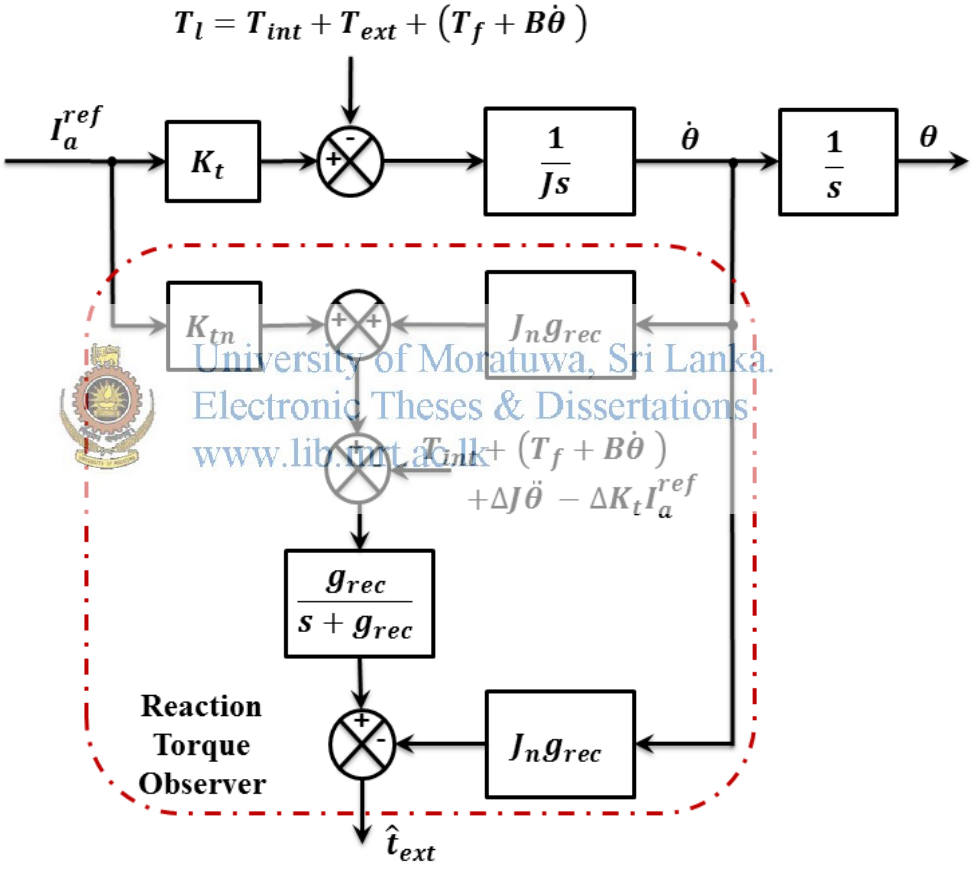





Figure 2.6 Estimation of reaction torque using RTOB

To estimate only the external torque, the torque exerted on the servo system due to friction component should be compensated [43] and the effects due to the self-inertia variation and the self-motor constant variation should be eliminated [44]. For a single motor control system $T_{int}$ can be assumed as zero [38].

# CHAPTER 3

# 3 HARDWARE SETUP

## 3.1 Hardware Experimental System

Figure 3.1 shows the hardware setup up of the system. The microcontroller mbed NXP LPC 1768 has been installed to implement the algorithm [39]. A current sensor module with differential amplifiers is used for current measurements and 2500 ppr incremental encoders are used to measure the positions. SD card reader is attached with SPI protocol for data storage. The detailed information of components is discussed in the following sub sections.

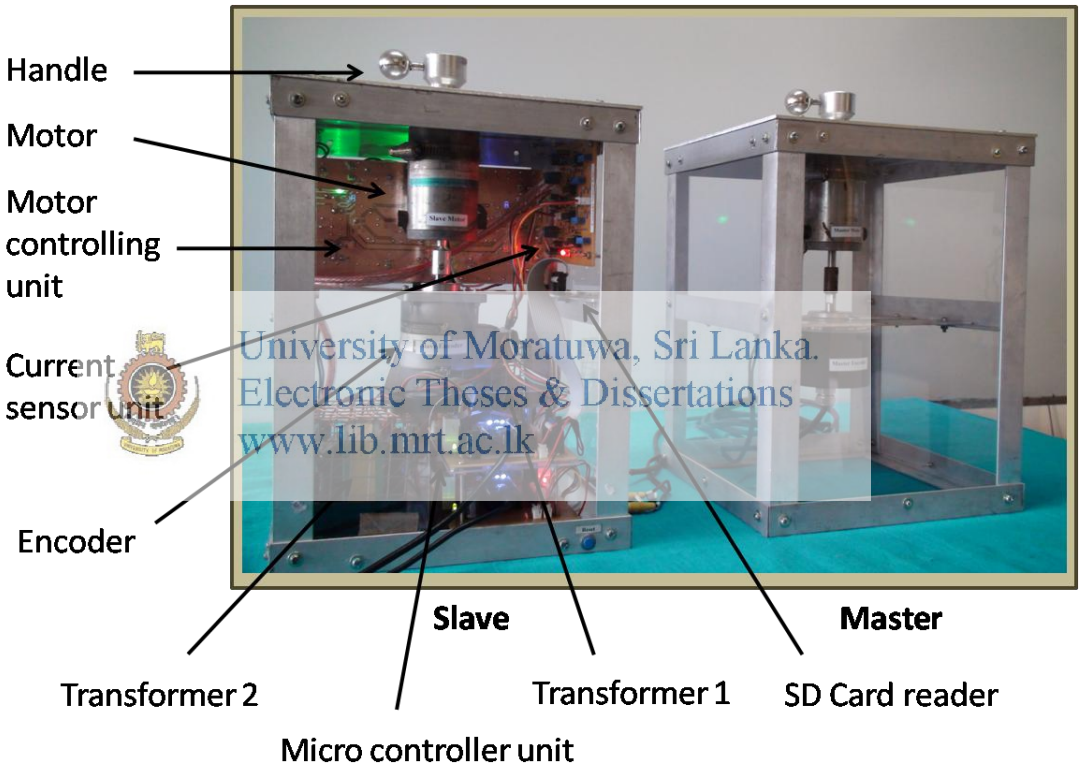




Figure 3.1 Hardware setup

## 3.2 Hardware setup circuits

Simulator Hardware setup basically comprises of 5 different modules of electronic circuit units. They are,

1. Power supply unit.
2. Main mbed microcontroller interfacing module with SD Card reader.
3. Auxiliary mbed microcontroller interfacing module.

4. Current sensor module with differential amplifiers.

5. DC motor driver.

The purpose of having 5 different modules is to make those units easy to analyze, easy to produce, easy to troubleshoot and easy to fix. The hardware combination is shown in below Figure 3.2.

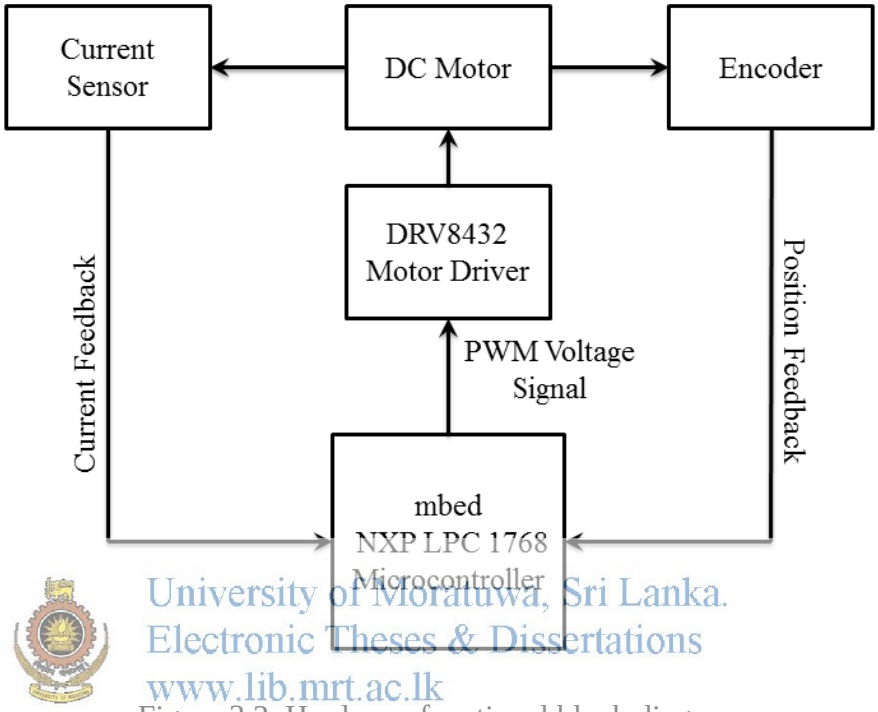





Figure 3.2 Hardware functional block diagram

### 3.2.1 Power supply unit

Power requirement of the total unit is supplied through two transformers which are supplied 230VAC as primary side input and then the secondary outputs are rectified and smoothened to get desired regulated DC power supply. Transformer 1 is used to supply power for the DC motors while transformer 2 power up the auxiliary circuits. Refer bellow list for further details.

Transformer 1– Supply for motors

| | | |
|---|---|---|
| Master and slave motors | : | 32VDC |

Transformer 2 – Auxiliary supply

| | | |
|---|---|---|
| Encoder (Master) | : | 12VDC |
| Encoder (Slave) | : | 12VDC |
| Current sensor unit | : | 8 VDC |
| mbed Microcontroller unit | : | 15 VDC |

Motor driving unit : 15 VDC

Cooling Fan (microcontroller board) : 12 VDC

The power flow diagram is shown in Figure 3.3 and Figure 3.4 shows the power supply circuit of the system. Feeding areas of the transformers are as follows.

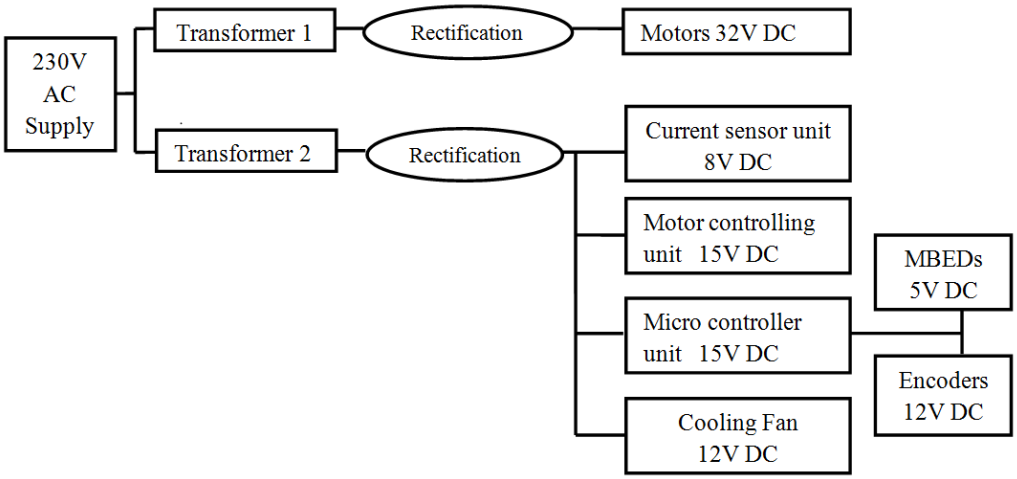



Figure 3.3 Power flow diagram

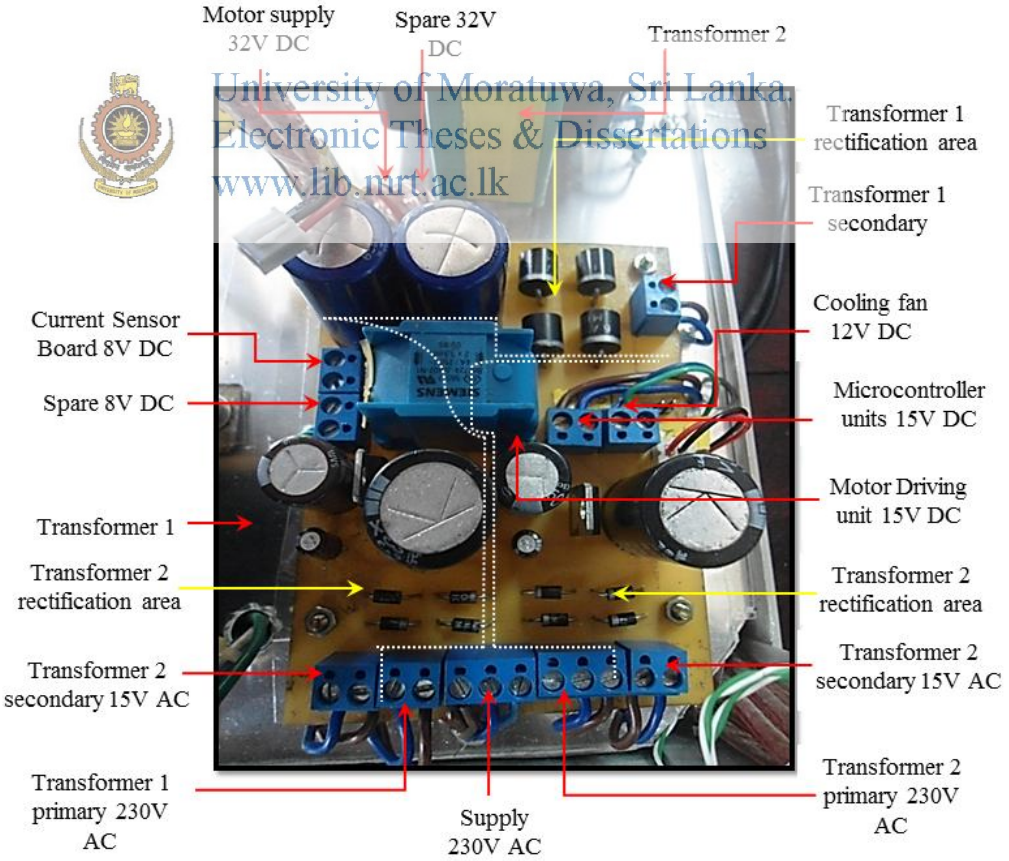





Figure 3.4 Power circuit configuration

### 3.2.2 Main mbed microcontroller interfacing module with SD card reader

The following Figure 3.5 describes the circuit module arrangement for main mbed microcontroller unit in this setup. This operates with 15VDC 1A supply which power up the microcontroller, two encoders and SD card module. The main microcontroller processes itself and issues command to the motor driver through the interface link. Position feedback is getting from encoder interfaces from both motors whereas current feedback is getting form current sensor interface from both motors.

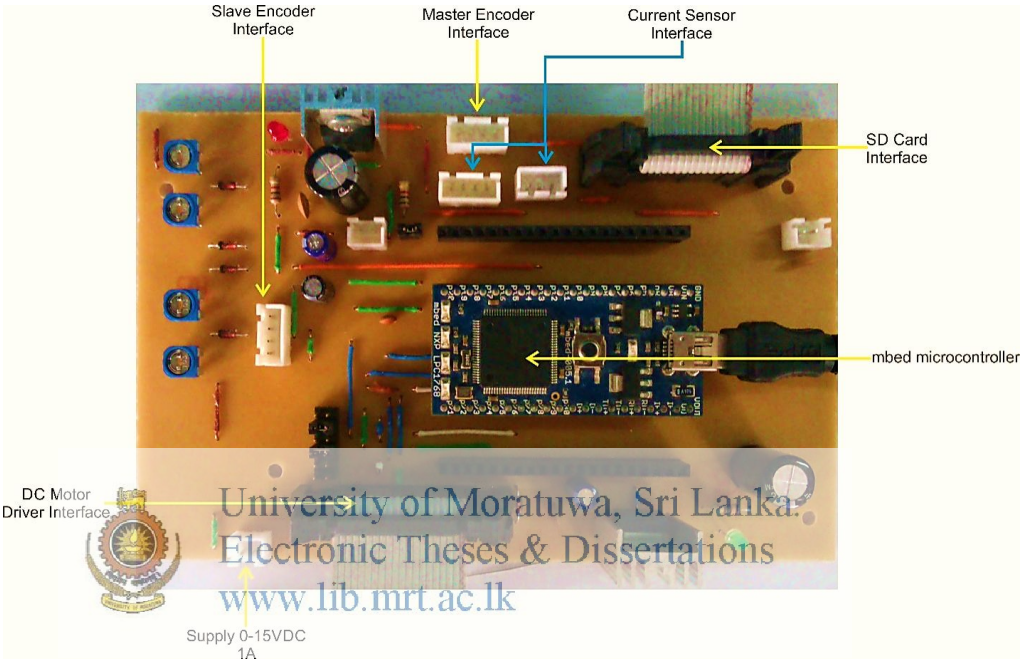




Figure 3.5 Microcontroller circuit

The SD card module is used to store the data for further analysis and record results. Since data writing process takes a considerable time, it was done by a parallel thread to reduce the cycle time of the algorithm because high cycle time reduces the accuracy of the algorithm parameters hence degrading the performances. Figure 3.6 shows used SD card module. This module was connected to the microcontroller using Serial peripheral Interface (SPI) connection.

Figure 3.6 SD Card module

### 3.2.3 Auxiliary mbed microcontroller interfacing module

Two mbed microcontroller units hsvr been used in the hardware setup. The auxiliary mbed microcontroller module used to send the master encoder position data to the main microcontroller. To sending data here it used 100Mbps Half Duplex Ethernet link. Printed circuit board arrangement is similar to the main mbed microcontroller inserted circuit module except SD card interface and signal link to motor driver.

### 3.2.4 Current sensor module with differential amplifiers

Accurate current sensing of the motors is essential because this setup is designed with internal sensing mechanism of DOB and RTOB for external force estimation which utilizes current feedback from the motors as one of the two controller inputs. Therefore very sensitive current sensor, even to detect miliampere range changes in current is a must. This current sensor module comprises with two sections, first section is current transducer and the other part was differential amplifier to amplify transducer signal before fed into the microcontroller. Basically the current sensors are giving one output from each sensor, upon that it has to be divide into two depending on the direction of current flows. The current measurement method used is shown in Figure 3.7.



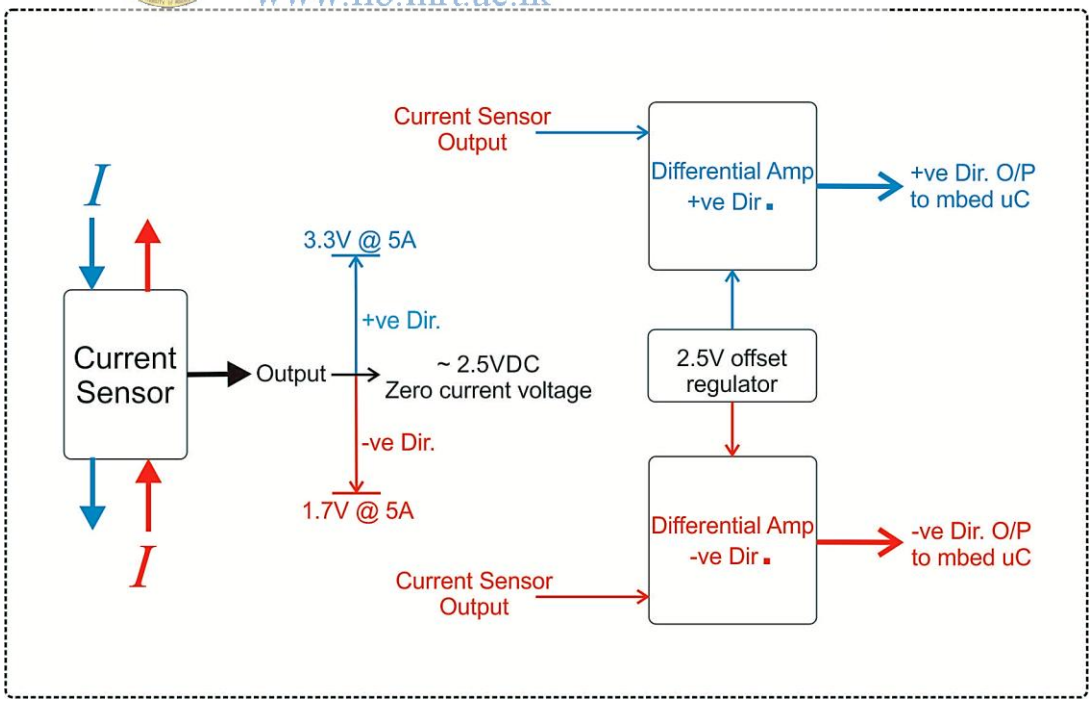



Figure 3.7 Method of current measurement from the current sensor

Two identical ACS712t 5A current transducers which are capable of measuring instantaneous current through a conductor have been used in this research. Refer Figure 3.7. Blue color direction taken as the +ve direction while red color direction taken as –ve direction. When current is flowing in the +ve direction the sensor output increases from zero current voltage which is set at 2.5V DC, with the gradient of 185mV/A, to 3.3V DC which corresponds to +ve 5A. Then that voltage difference up to 800mV is fed to differential amplifier to amplify current sensor output between 0-3.3V DC. The -ve direction operation is also same as the +ve direction, the only difference is voltage varies in between 2.5V DC and 1.7V DC which correspond to zero current and –ve 5A respectively. The Current measuring circuit is shown in Figure 3.8 while the current sensors are shown in Figure 3.9.

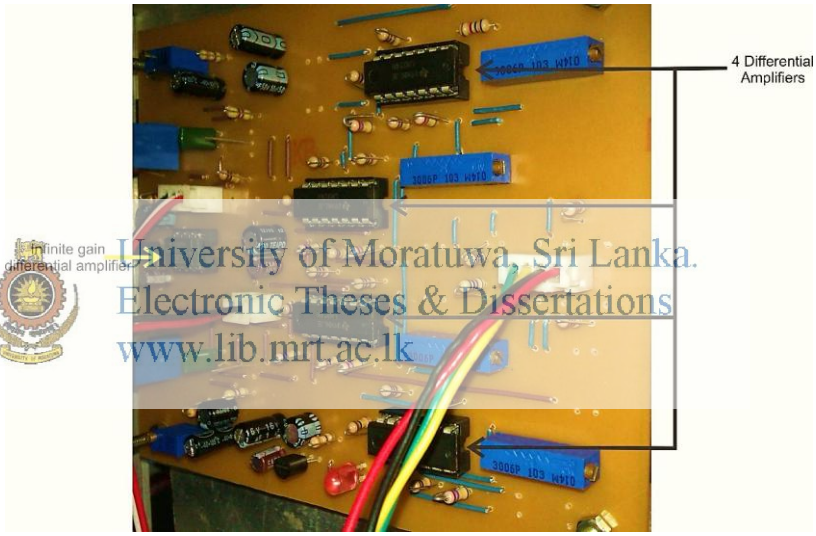




Figure 3.8 Current measuring circuit

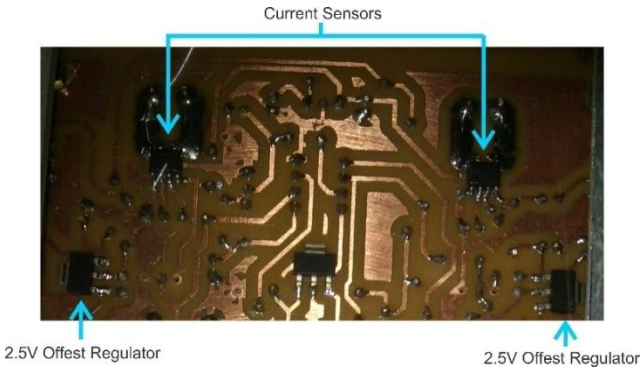



Figure 3.9 ACS712t 5A current sensor

Figure 3.10 shows the output response of the current sensor circuit. In order to achieve high precision, well-tuned PID controller has been included. Therefore, the known current command is applied to the motor and measured the current response of the current sensor. Then PID constant values are tuned so as to give current response same as the current command. The proportional gain, the derivative gain and integral gain values are 8, 0.015 and 1.5 respectively. According to the Figure 3.11, it shows that the current controller is well tuned because the current response is nearly same as the current command.

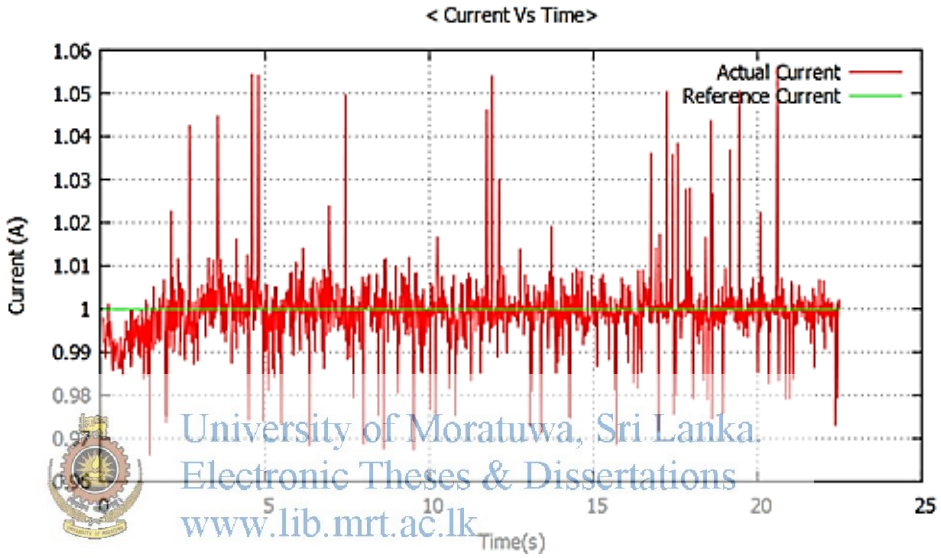




Figure 3.10 Output response of the current measuring circuit

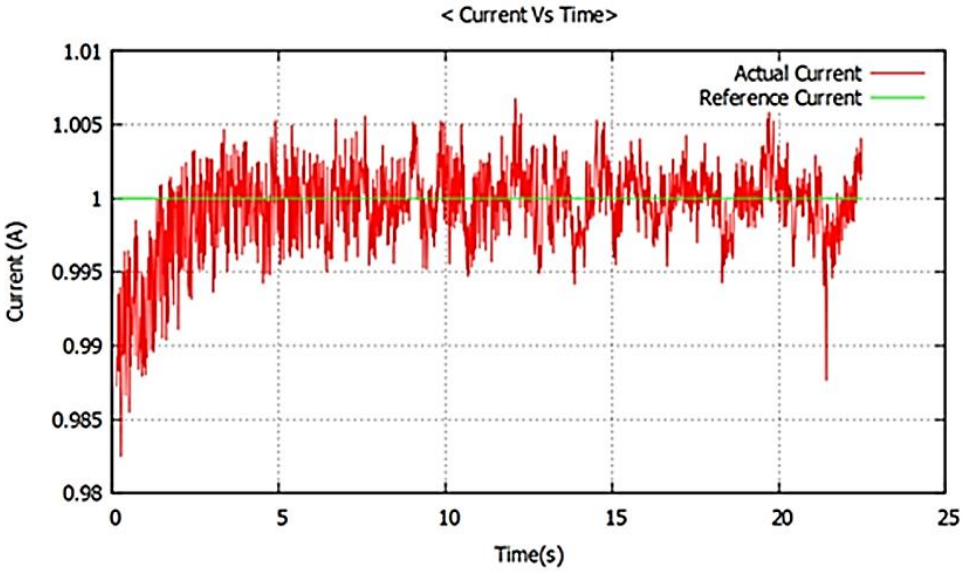



Figure 3.11 Output response of the current with tuned PID

### 3.2.5 DC motor driver

Motor control board constructed for the research is shown in Figure 3.12. DRV 8432 dual full bridge PWM motor driver is selected as the driver IC. It is possible to use one DRV 8432 driver for two DC motors; however it is not suitable when motors are drawing high currents. Two drivers have been implemented in the motor control board for two DC motors of the master and the slave actuators.

DRV 8432 provides high performance and integrated dual full bridge drivers with high protection system for safeguarding the device against multiple fault situations. This driver package consists of thick heat slug on the top of the driver in order to dissipate heat through the heat sink. In this circuit, two 12 V supplies are used; one for DRV drivers and other for power up the H bridge.


Figure 3.12 Motor control board with DRV 8432 driver

In normal operation, inductance in the motor is sufficient to provide low di/dt. However, when the motor is in short circuit condition there may be very high current which may exceed the maximum current limit of the circuit due to low impedance path and very high di/dt. Therefore, with the use of external inductances, current will rise at a slower rate. So with the use of suitable inductor in series the rate of current rise can be controlled. One of the major reasons for choosing DRV 8432 is that it provides very high efficiency at relatively very high switching frequencies such as 500 kHz. In addition to operate at high frequencies, it is possible to get wide range of

input and output duty cycle values. Figure 3.13 shows the output duty cycle verses input duty cycle curve.

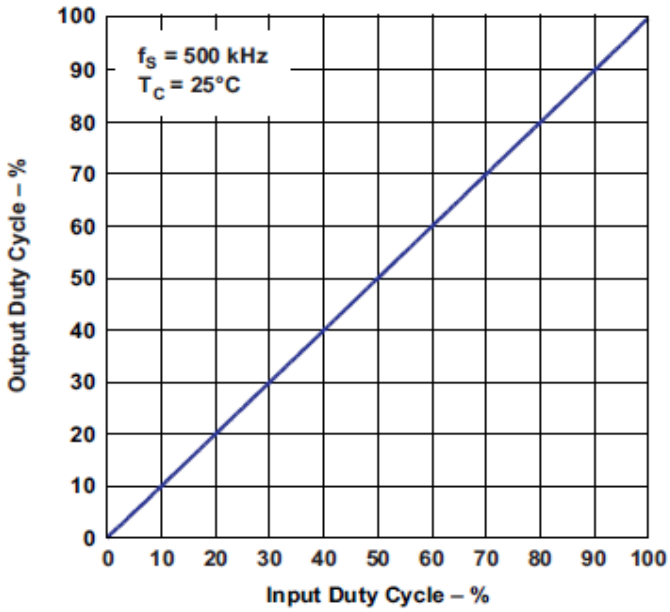



Figure 3.13 Output duty cycle versus input duty cycle curve

### 3.3 Control board



DC motor is controlled by Mbed LPC 1768 microcontroller of 96 MHz. Pin configuration of Mbed is shown in Figure 3.14. The Mbed is the one of online platform for fast, low risk prototyping of microcontroller based systems. It is cost effective and low power device. It has powerful 32 bit ARM Cortex-M3 processor running at 96 MHz, 512 kB of flash memory and 64 kB of SRAM. LPC 1768 is mounted on the Mbed board which uses a 40 pin DIP with a 0.1 inch pitch. Programs can be written in C++ and then compile and download them to run on the Mbed NXP LPC1768 microcontroller. Further, there is no software required to install and almost everything even the compiler is online. Complier supported browsers include Internet Explorer, Firefox, Safari or Chrome running on Windows, Mac or Linux platforms. The compiler and libraries are easy to use but powerful enough to work on complex real world applications. Downloading programs are as simple as using a USB flash drive and no external programmer is required.

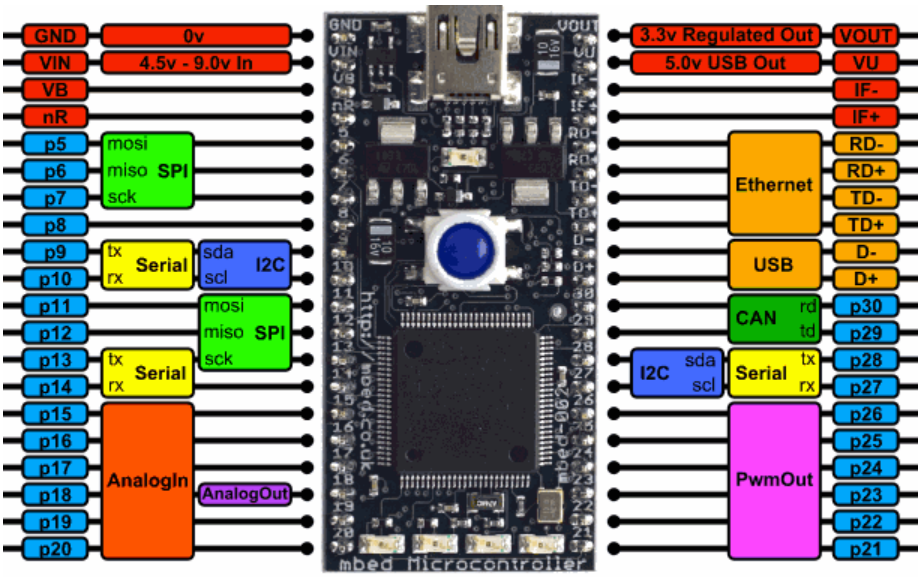



Figure 3.14 Mbed pin configuration

### 3.4 DC motors

Two identical DC motors of same manufacturer have been used for both master and slave units which is shown in Figure 3.15. The motors are brushed type permanent magnet DC motors. Dimensions of the motor are shown in Figure 3.16. Motors used with coupled encoders to measure the position, velocity and acceleration of both master and slave. Specification of the motor is given in the Appendix I.



Figure 3.15 Electrocraft E240 DC motor

## 3.5 Encoder

Positions of the manipulators are tracked by using two incremental rotary encoders. These encoders are subjected to quadrature encoder interface (QEI) so that the resolution is increased. Master and slave units have encoders of different manufacturers.

### 3.5.1 Master encoder

5000 pulses per revolution (ppr); optical incremental rotary type encoder has been used with the master motor to achieve required high accuracy. The encoder used is shown in Figure 3.16 and the specification of the master encoder is given in the Appendix II. Using quadrature encoder interface techniques this resolution has been increased to 20,000 ppr.



Figure 3.16 Master encoder

### 3.5.2 Slave encoder

2500 ppr optical incremental rotary type encoder has been used with the slave motor to achieve required high accuracy. The encoder used is shown in Figure 3.17 and the specification of the slave encoder is given in the Appendix III. Using quadrature encoder interface techniques this resolution can be increased four times for each cycle.


Figure 3.17 H25D BEI encoder

### 3.6 Coupling Motor and Encoder

Motor and the encoder are connected with an aluminum coupler as shown in Figure 3.18. Coupler has been designed such that, it can bear the connection between motor and the encoder even in minor nonalignment conditions. Specification of the coupler is given in the Appendix IV.

Figure 3.18 Motor-encoder coupler



### 3.7 Hardware modification of QEI module

Mbed microcontroller is not capable of detecting signals of high resolution encoders at higher speed. Hardware modification shown in Figure 3.19 and Figure 3.20 could be used to facilitate high speed motor encoder [40]. The rotary encoder must be connected to nodes p1.20 and p1.23. These nodes are also used to connect LED2 and LED4, respectively, on the mbed module. The encoder pulls the port pins up through 330 Ω series resistors to +3.3V. 3.3V supply was given to the encoder's common terminal from the pin 40. When the encoder breaks contact, the port pins are pulled down passively by 1kΩ resistors to ground QEIHW.h library should be involved instead of QEI.h with this hardware modification.

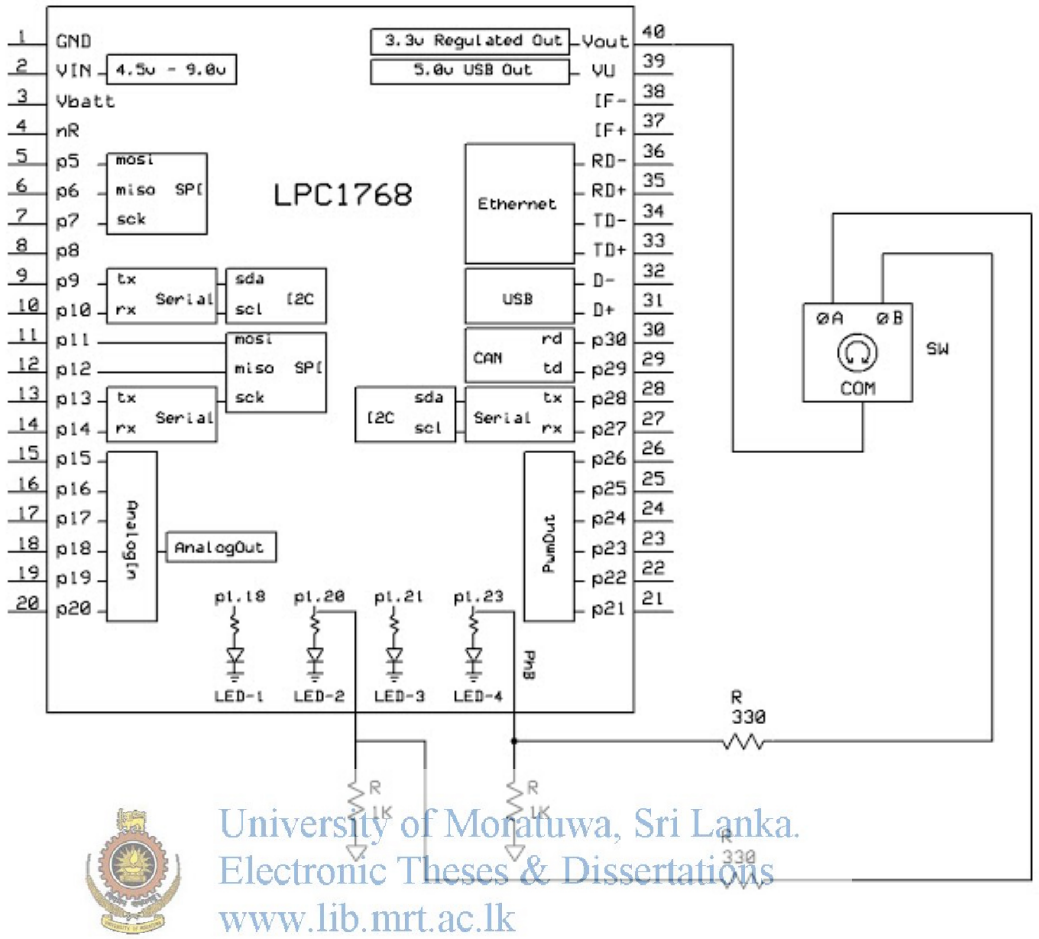





Figure 3.19 Hardware modification circuit diagram

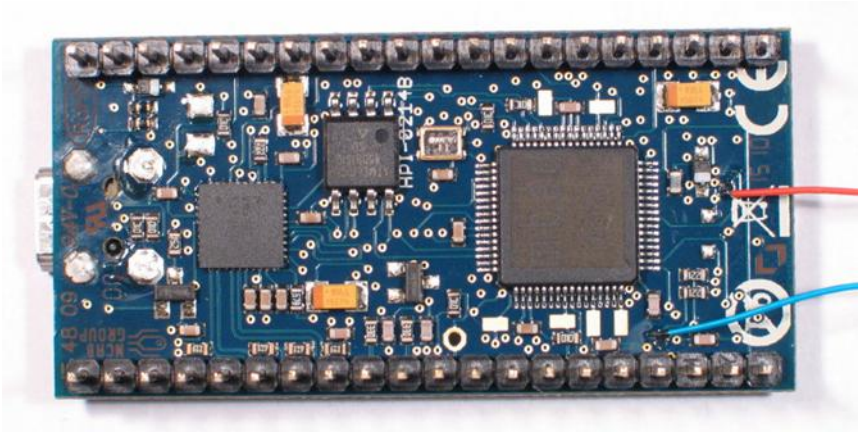


Figure 3.20 Hardware modification of QEI Module

# CHAPTER 4

# 4 FORCE FEEDBACK BILATERAL GRIPPER

## 4.1 Introduction

Each object has a pre-defined force limit which could be exerted on it externally without damaging the object. In this proposed system a force lock is introduced on the slave side to protect the object from the excessive force applied through the master. Also a force limit feedback is sent to the master through a vibration notification when the reaction force on the slave is equal to the predefined tolerable force limit of the object. Proposed system includes a virtual spring controller on master side while force lock is engaged on the slave side. The lock release condition could be defined by the operator which should be satisfied when releasing the gripped object. Master and slave follow bilateral control when it is not running in force lock mode. The proposed system is applied to 1 DOF bilateral hardware system which consists of identical two rotary motor and therefore torque lock is applied instead of the force lock. Thus, the main functions available during bilateral force feedback gripper could be listed as follows.



- Bilateral control
- Torque control
- Spring control
- Vibration notification

The modeling of each function is described in the forthcoming sections in this chapter. The results are shown at the end of the chapter for the proposed torque feedback bilateral gripper.

## 4.2 Bilateral Control

Bilateral control is one of the technologies for the teleoperation of a master–slave system. In bilateral control, realization of the law of action and reaction in a master slave system is the most important. If the transmitting of information is based on acceleration control, it is easy to transform the information in acceleration dimension into force and position dimensions, because acceleration times mass equals to force, and the second-order derivative of position equals to acceleration. Moreover, since a

DOB is implemented in the bilateral control system in this research, the transmission of information is based on acceleration control, which leads to secure the robustness [41].

Since proposed system is using identical 1 DOF two rotary motors the theory of bilateral control without scaling is applied. In bilateral control, as the aim for the artificial realization of law of action and reaction between the action torque that is applied to a master system by the operator and the reaction torque that is applied to a slave system from the real environment, the objective function of torque between master and slave systems is represented as (4.1). The aim of position for the master and slave systems to track each other is also represented as (4.2).

$$T_m + T_s = 0 \tag{4.1}$$

$$\theta_m - \theta_s = 0 \tag{4.2}$$

The notations $\theta$, $T$ and subscript $m$, $s$ denote angle position response, torque response, master and slave respectively. Since the dimension of torque information is different from the dimension of position information, the sum and differential in them are not able to be calculated simply in one axis since force and position are said to be orthogonal. Thus, the aims of torque control and position control are standardized in common dimension of acceleration. The equations (4.1) and (4.2) are transformed to acceleration dimension as in (4.3) and (4.4) considering equal inertias in both manipulators because position control and torque control are now in common dimension of acceleration.



$$\ddot{\theta}_m + \ddot{\theta}_s = 0 \tag{4.3}$$

$$\ddot{\theta}_m - \ddot{\theta}_s = 0 \tag{4.4}$$

$\ddot{\theta}$ denotes the acceleration. If (4.3) and (4.4) are needed to achieve simultaneously $\ddot{\theta}_m, \ddot{\theta}_s$ should be equal to zero separately. The transmission of the torque sensation between master and slave systems in rigid real time with attaining of the law of action and reaction is achieved by proper design of DOB based on acceleration control. Therefore (4.3) modified to (4.5) which represents torque servoing based on the acceleration dimension which is called the common mode between the master and the slave systems.

Equation (4.4) represents the acceleration information obtained through second derivative of position information. Therefore the information delays essentially. Thus, (4.6) is expressed by arrow sign. It means that the difference in the acceleration dimension should converge to zero as quickly as possible. Equation (4.6) is known as differential mode between the master and the slave systems.

$$\ddot{\theta}_m + \ddot{\theta}_s = \ddot{\theta}_{com} = 0 \quad (4.5)$$

$$\ddot{\theta}_m - \ddot{\theta}_s = \ddot{\theta}_{diff} \rightarrow 0 \quad (4.6)$$

In order to achieve (4.5) and (4.6) the system should be controlled with a low sampling time. Figure 4.1 shows the conventional block diagram of bilateral controller which satisfies (4.5) and (4.6) [42].

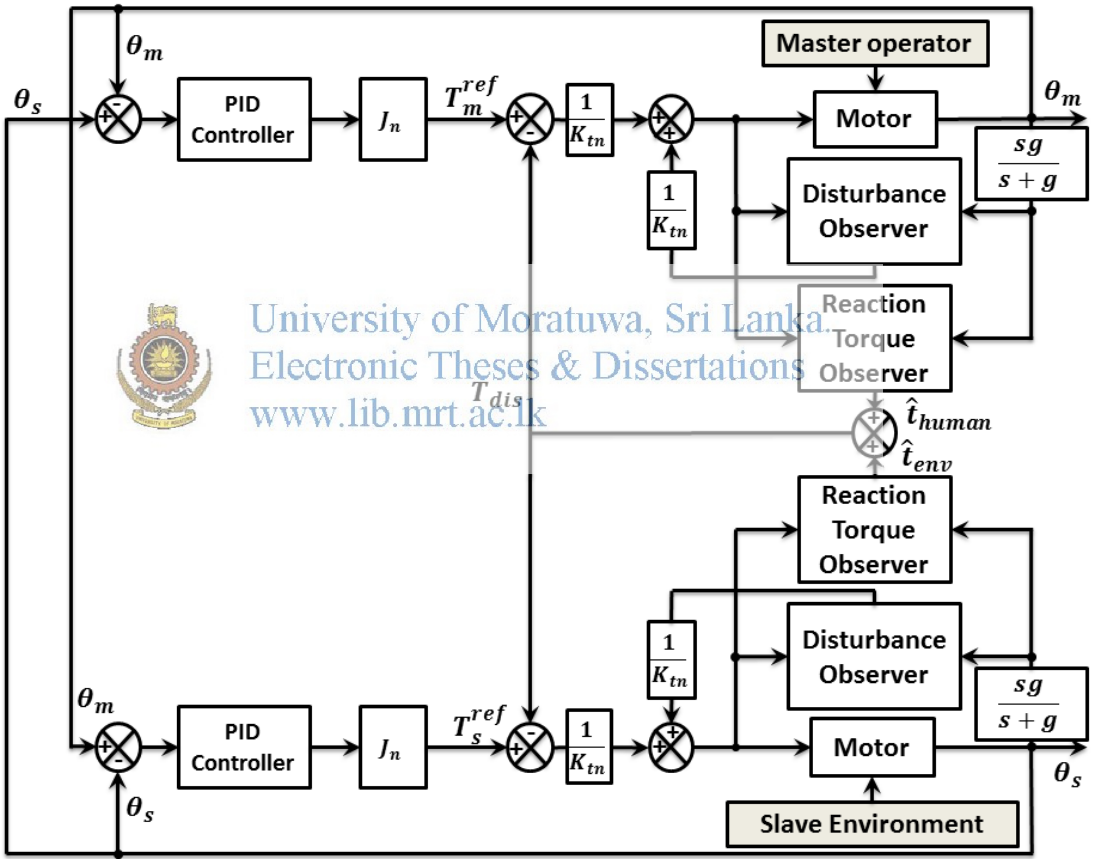




Figure 4.1 Block diagram of bilateral controller

Master and slave sides are symmetrical. $J_n$ and $K_{tn}$ are the motor nominal inertia and the nominal torque constants of DC motors. Master is fed with the position response of slave $\theta_s$ as the reference and master position response $\theta_m$ as the feedback. Slave is fed with the position response of master as the reference and slave position response

as the feedback. Two PID controllers are used to compensate the position error. When tuning the PID first introduce proportional gain starting from zero and increase until master and the slave track positions of each other. As the proportional gain is increased some vibrations are experienced and introduce derivative gain to reduce vibrations. Finally introduce integral gain to reduce the steady state error of position error which cannot be compensated by increasing the proportional gain value. Disturbance Observer is used to estimate the disturbance force on the master and the slave and corresponding current required for compensating the disturbance is fed back to the master and the slave. Reaction Torque Observer is used to estimate the reaction torques acting from the master and the slave environment and torque error is fed to both master and slave control blocks to align the torque response of both master and slave.

### 4.3 Torque Control

The torque controller protects the gripped object from the excessive torque imposed by the master operator. The torque limit of the object should be specified previously by the operator. As for the simulation purpose, the reaction of the environmental object can be modeled as mass, damper and spring combination as in (4.7).



$$F = M\ddot{\theta} + B\dot{\theta} + k\theta \tag{4.7}$$

Where $F$, $M$, $B$, $\dot{\theta}$, and $k$ denote reaction of the object, object mass, damping coefficient, velocity response and spring coefficient.

The master and the slave follow bilateral control until slave environment reaction reaches the specified torque limit of the object. The switching of control strategy from bilateral control to torque controller is activated when the reaction torque from the slave environment is equal or exceeding the specified torque limit. Thus, slave manipulator is locked at that maximum torque limit until the master operator satisfies the object release conditions and the master and the slave no longer follow the bilateral control. Figure 4.2 shows the control block diagram of the torque controller.

The operator defined torque limit $T_{limit}$ is taken as the reference torque and the estimated reaction torque of the slave environment via the RTOB of slave actuator, $\hat{t}_{ext}$ is taken as the feedback. The torque error $T_{err}$ in (4.8) is fed to the PID controller. The converted output of the PID controller provides current reference $I_a^{ref}$

to the slave motor. The disturbances are also estimated via DOB and compensating current $I_{dis}$ is fed to the slave motor current input.

$$T_{err} = T_{limit} - \hat{t}_{ext} \tag{4.8}$$

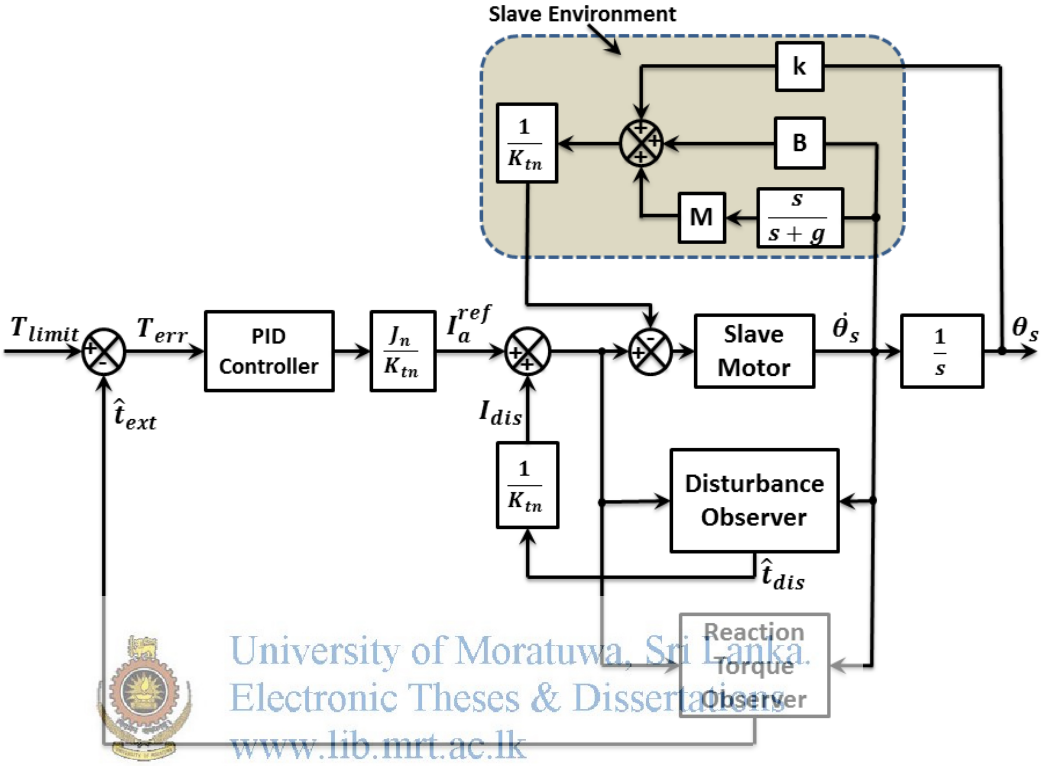





Figure 4.2 Block diagram of torque controller

## 4.4 Spring Control

By the time of torque lock activation the objective equations (4.1), (4.2) of bilateral control are violated because slave torque is locked at the predefined torque limit $T_{limit}$. Therefore master operator feels the loss of reaction torque coming from the slave environment as a sudden torque reduction causing a forward position step which is undesirable although it provides a torque limit achievement notification. Therefore a virtual spring controller is introduced on to the master manipulator. The spring controller used in the simulation of force feedback gripper using MATLAB m code is shown in Figure 4.3. In this scenario master operator experiences position step due to loss of slave environment. The magnitude of the position step can be reduced by increasing the spring constant of the virtual spring. This position step is undesirable in practical applications because it gives a shockwave to the master

operator. Therefore for the practical application block diagram of Figure 4.3 is modified as shown in Figure 4.4 to facilitate the sense of continuous torque increasing to the operator if he further presses his leaver towards torque increasing direction.

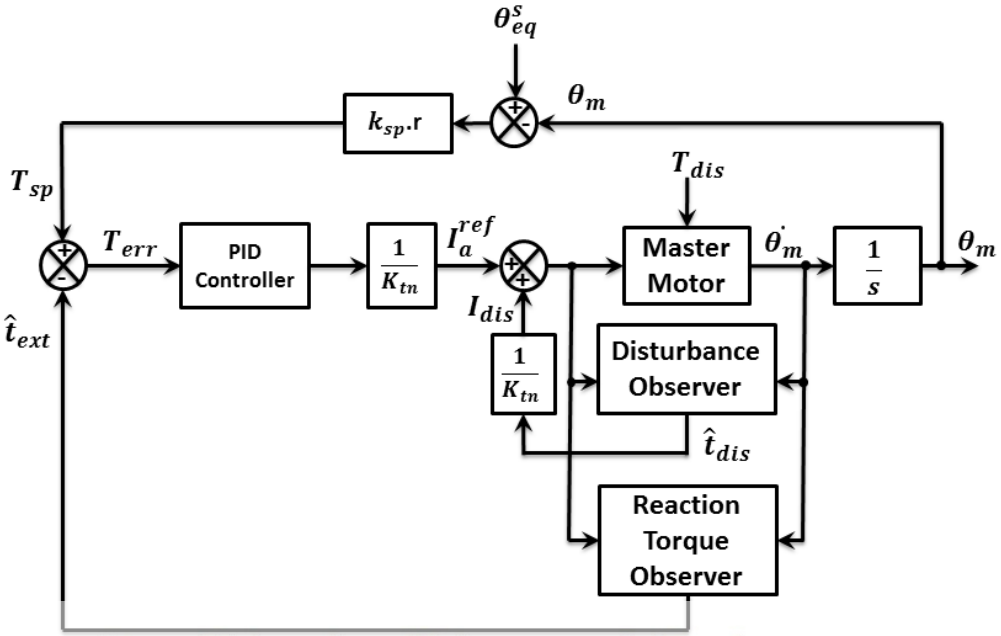



Figure 4.3 Spring controller with loss of reaction torque notification



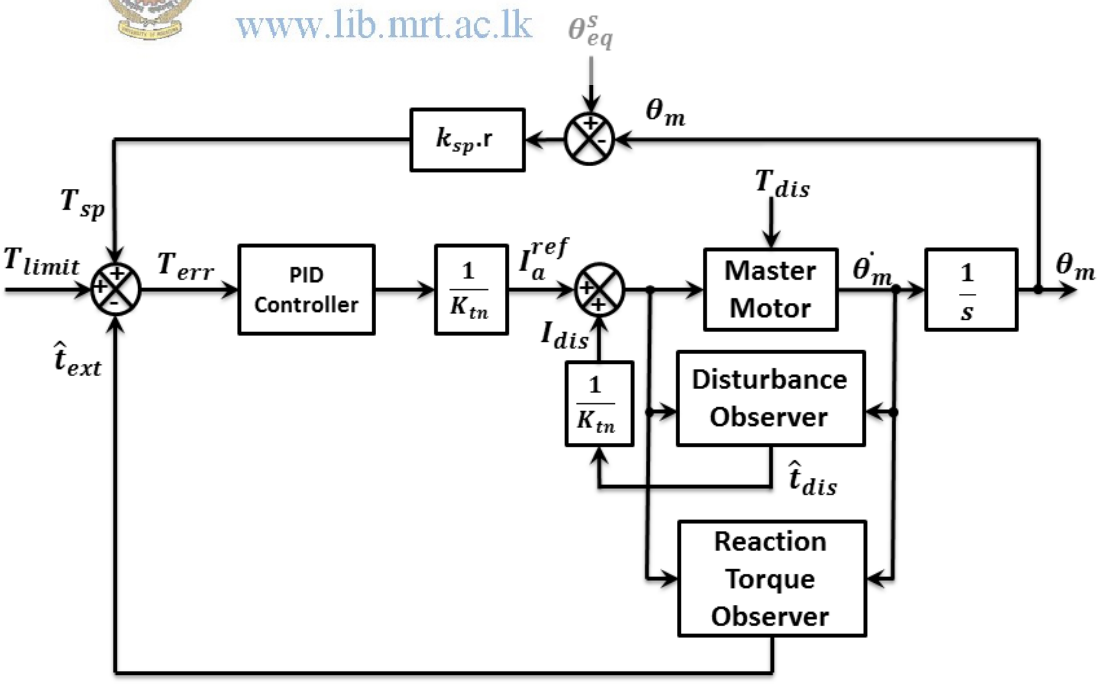



Figure 4.4 Spring controller with the sense of continuous torque increasing

This would not be having any effect on the gripped object on the slave. The spring torque $T_{sp}$ is produced as in (4.9) around the equilibrium position $\theta_{eq}^{s}$ which is continuously copied from the slave position $\theta_s$ during torque is locked.

$$T_{sp} = k_{sp}(\theta_{eq}^{s} - \theta_m).r \quad (4.9)$$

The notations $k_{sp}$, $r$ denote virtual spring coefficient and length of the slave actuator handle. Reference torque $T_{ref}$ for both scenarios is presented as (4.10) and (4.11).

$$T_{ref} = k_{sp}(\theta_{eq}^{s} - \theta_m).r \quad (4.10)$$

$$T_{ref} = T_{limit} + k_{sp}(\theta_{eq}^{s} - \theta_m).r \quad (4.11)$$

The intension of virtual spring controller is to equalize the positions in the lock mode.

## 4.5 Vibration Notification

A vibration alert is generated on the master operator side to notify the torque limit attainment on the slave side. This is available only for 500ms. A square wave signal is added on to the slave equilibrium position which is copied from the slave position during torque lock is active. It is used as the reference to the master position controller. Here also $T_{limit}$ is used as a part of torque reference to avoid untying the operator hand from negative step reaction feedback through reaction torque observer of master side due to slave environment reaction loss in bilateral control. Figure 4.5 shows the control block diagram of the vibrator.



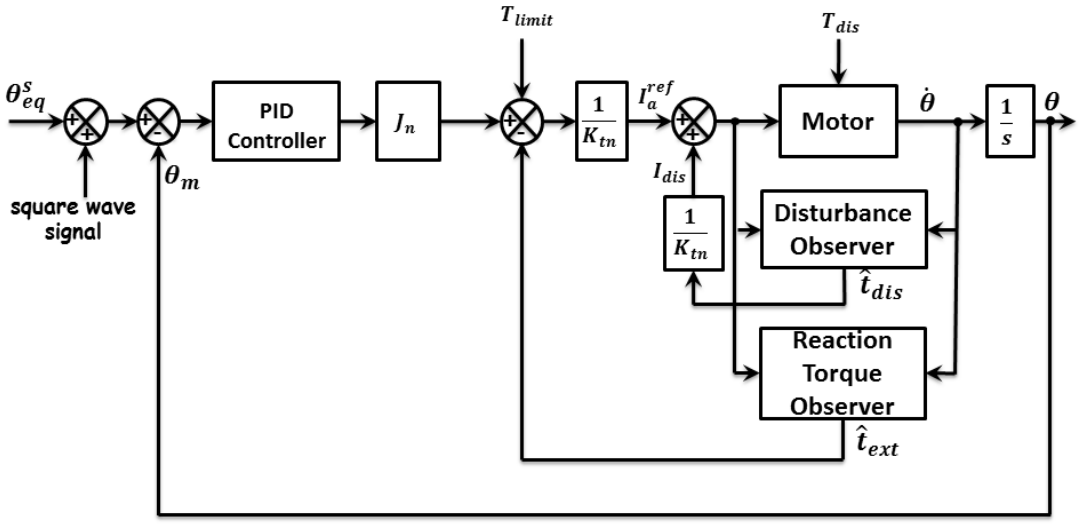



Figure 4.5 Block diagram of vibration signal generator

## 4.6 Object Release

Object release condition can be specified by the user because it provides operator flexibility. The release conditions can be defined as shown in Table 4.1 which determines whether operator has free hand with active force lock or hand is on the master actuator during whole process of bilateral motion and torque grip.

Table 4.1 Possible combinations of object release

| Condition | Operator free hand |
|---|---|
| $x_{eq}^{s} - x_m < 0$ | X |
| $\hat{t}_{ext,m} < T_{limit}$ | X |
| $(x_{eq}^{s} - x_m < \mathbf{0})$ & $(\hat{t}_{ext,m} < T_{limit})$ | X |
| $\hat{t}_{ext,m} < 0$ | ✓ |
| $(x_{eq}^{s} - x_m < \mathbf{0})$ & $(\hat{t}_{ext,m} < 0)$ | ✓ |

Figure 4.6 shows the supervisory control block diagram of the propose system including above functions.



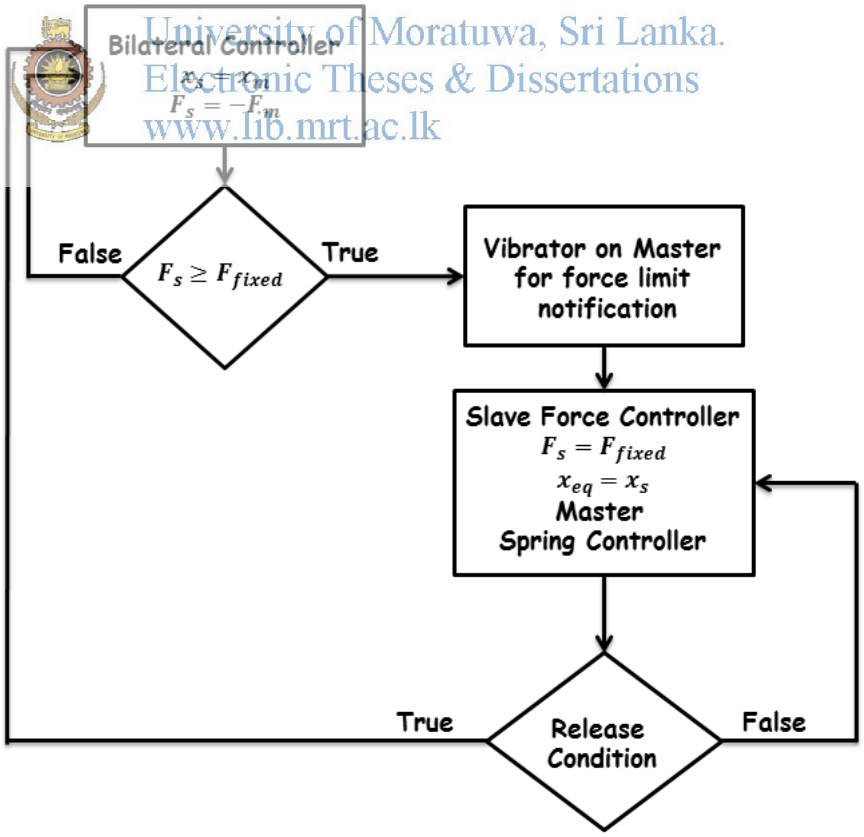



Figure 4.6 Supervisory control block diagram

## 4.7 Results

The results of force feedback gripper to be used with bilateral teleoperation obtained from both MATLAB m code simulation and experimental hardware setup are shown in forthcoming sub sections.

### 4.7.1 Simulation results

The parameters used in the simulation are listed in Table 4.2. The force limit of the object is considered as 9N and simulation is carried out using parameters of identical two linear motors. Therefore results are shown in linear scale. Refer Appendix V for MATLAB m code.

Table 4.2 Simulation parameters

| **Parameter** | | **Value** |
|---|---|---|
| $K_{fn}$ | Force constant | 24 N/A |
| $M_n$ | Motor shaft weight | 0.40 kg |
| $K_P$ | Proportional gain of bilateral controller | 350 |
| $K_d$ | Derivative gain of bilateral controller | 25 |
| $K_{P,sp}$ | Proportional gain of spring controller | 300 |
| $K_{d,sp}$ | Derivative gain of spring controller | 25 |
| $K_{P,f}$ | Proportional gain of force controller | 3.3 |
| $K_{d,f}$ | Derivative gain of force controller | 1.1 |
| $g$ | Filter constant of DOB | 150 rad/s |
| $B$ | Viscous friction coefficient | 5 $N/ms^{-1}$ |
| $M$ | Mass of object | 0.1 kg |
| $k$ | Spring constant of object | 250 N/m |
| $K_{sp}$ | Spring constant of spring controller | 100 N/m |
| $T_s$ | Sampling period | 100 µs |
| $F_{limit}$ | Object force limit | 9 N |



It is assumed that at the point of starting of simulation both master and slave grippers are fully opened and those spans are set to be 0.15m. Figure 4.7 shows the applied force profile by the master. Within the first 5s master applies the force to overcome the friction of the system and during this period both master and slave move 0.05m

obeying the rules of the bilateral control as position responses shown in Figure 4.9 and Figure 4.10. A modeled object which is having 0.1m diameter is placed in between the slave gripper arms. Therefore slave handle is in contact with the object when it is moved 0.05m. Now master increases his force gradually to reach 10N at 10s and maintain same force for 2s and reduce his force gradually to zero and again increase to 12N and gradually reduce and at 19s reverse the force to satisfy the force lock release conditions. Figure 4.8 shows slave force response.

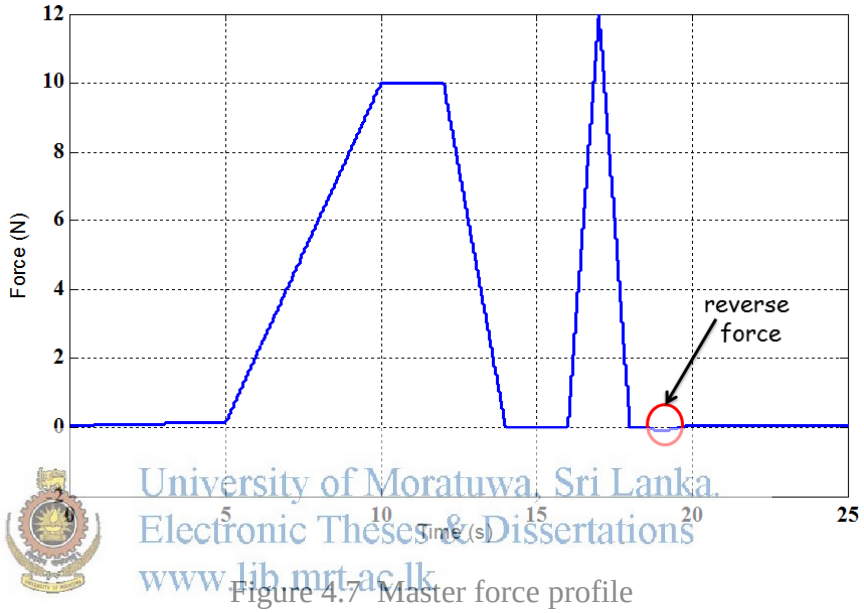





Figure 4.7 Master force profile

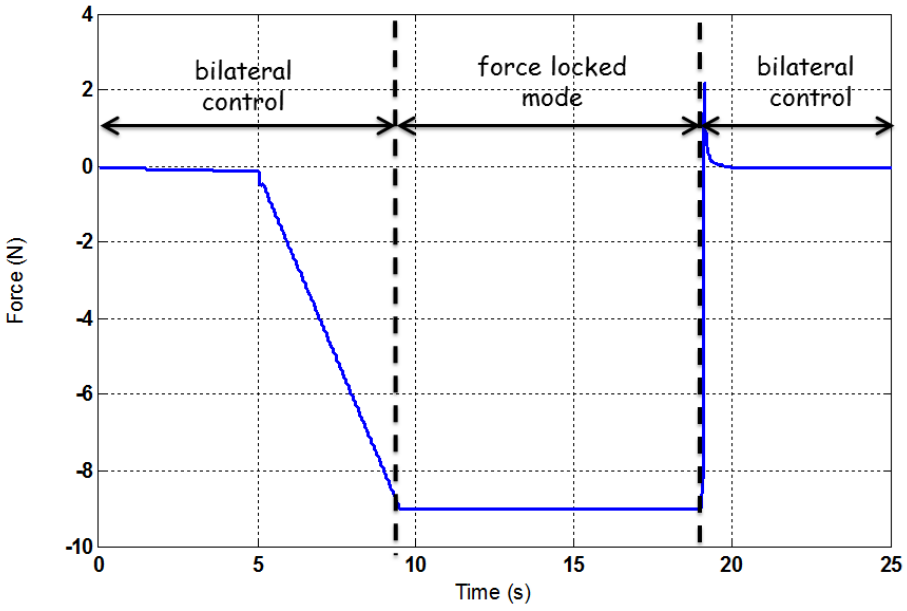



Figure 4.8 Slave force response

By observing slave force response, it can be observed slave force have reached 9N around 9.5s. Now force controller is activated and therefore slave force response remains at 9N, although the master changes his force profile without reversing the force. When master reverses his force and master position response is less than the slave position response, the object is released and bilateral control is switched back. Figure 4.9 shows the master position response and Figure 4.10 shows the slave position response.

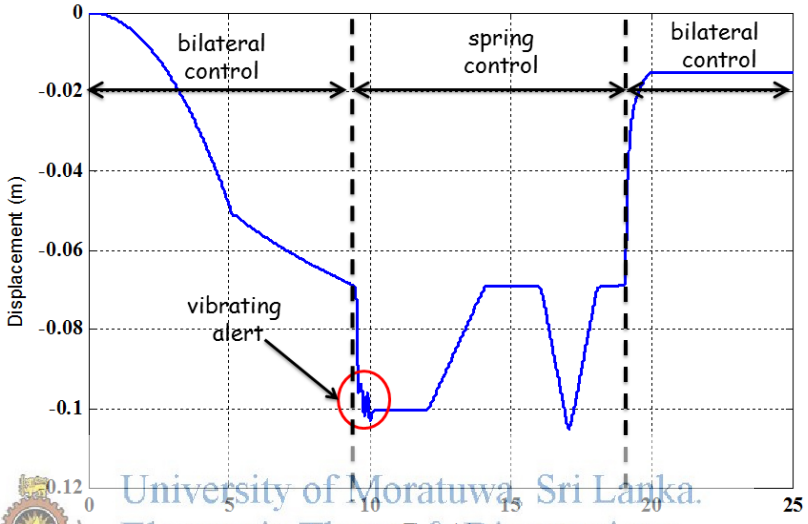





Figure 4.9 Master position response

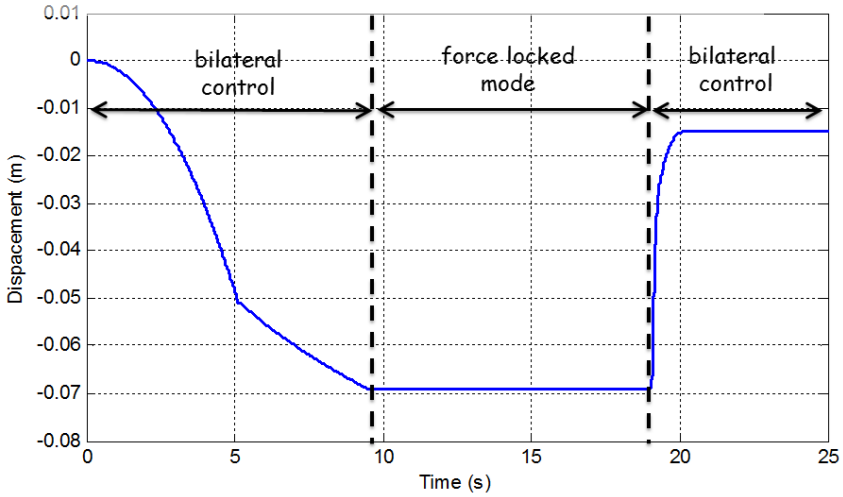



Figure 4.10 Slave position response

It can be observed that until the object is being contacted by the gripper finger, the system is accelerating from 0s to 5s. After slave is in contact with the object, due to

force applied on master, object is compressed until it reaches its force limit. With the activation of the force controller, the slave position response remains unchanged due to the object property until master applies the releasing force. The spring controller which is attached to the master during force locked period on the slave is controlling the master position. The vibrating alert on master is also shown. When force lock release condition is satisfied switching of control strategy from force control and spring control to bilateral control shows positions jump back. Because at this transition slave side force step of 9N and master side small reverse force provides a force error nearly equal to 9N to the bilateral controller. When the step size is increasing it will cause system instability after force lock is being released due to position and force jump backs.

Figure 4.11 shows master applied force and slave force response in same figure where operating regions are marked. Figure 4.12 shows master and slave position responses in same figure where regions are marked. Master and slave follow bilateral control when it is not running in force lock mode. When the object is released both master and slave follows bilateral control and settle down according to the applied force on the master.



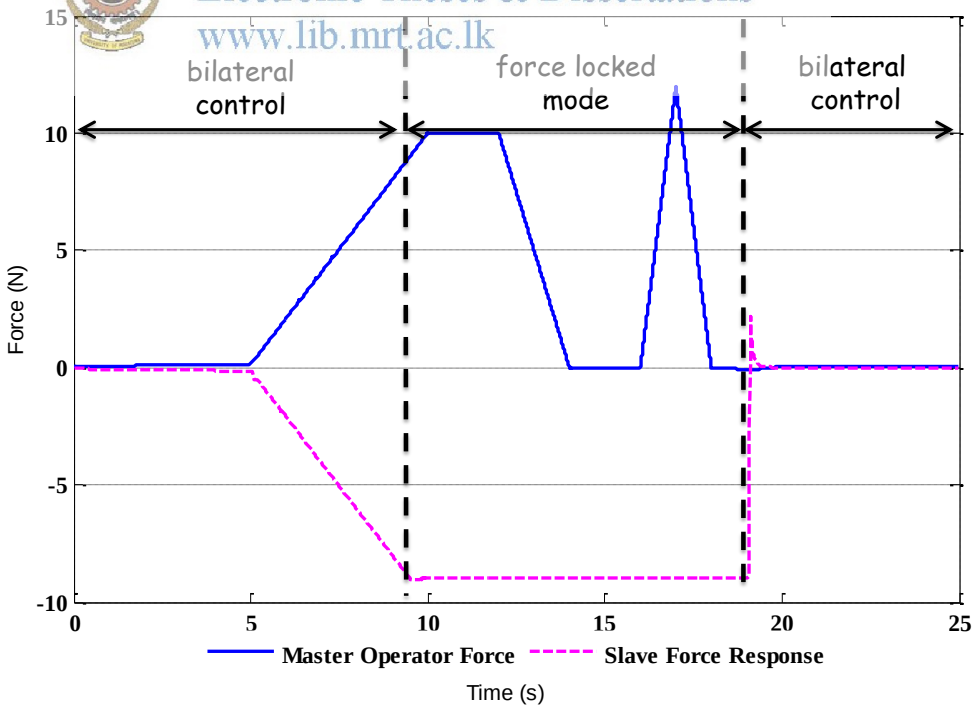



Figure 4.11 Master force profile and slave force response on it

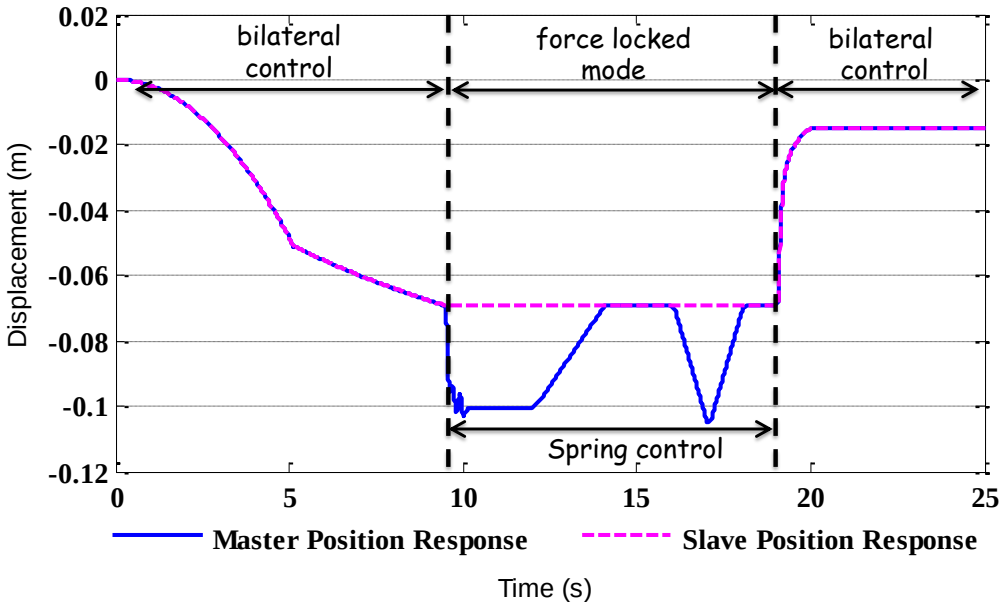



Figure 4.12 Master and slave position response

### 4.7.2 Experimental results

The parameters used in the experiment are listed in Table 4.3. Since experiment is carried out using two identical rotary motors, torque limit is considered as 0.1Nm which correspond to 2.5N because length of the handle is 0.04m. Therefore results are shown in circular dimensions. Refer Appendix VI for c code written in the mbed microcontroller.



Table 4.3 Parameters used in the experiment

| **Parameter** | | **Value** |
|---|---|---|
| $K_{tn}$ | Torque constant | 0.135 Nm/A |
| $J_n$ | Motor inertia | 0.00091 kgm$^2$ |
| $K_p$ | Proportional gain of bilateral controller | 900.0 |
| $K_d$ | Derivative gain of bilateral controller | 10.0 |
| $K_i$ | Integral gain of bilateral controller | 10.0 |
| $T_{limit}$ | Torque limit of torque controller | 0.10 Nm |
| $Kp,T$ | Proportional gain of torque controller | 650.0 |
| $K_{d,T}$ | Derivative gain of torque controller | 1.10 |
| $K_{i,T}$ | Integral gain of torque controller | 60.0 |
| $k_{sp}$ | Spring coefficient of spring controller | 15.0 N/m |

| Parameter | | Value |
|---|---|---|
| $K_{tn}$ | Torque constant | 0.135 Nm/A |
| $K_{p,sp}$ | Proportional gain of spring controller | 550.0 |
| $K_{d,sp}$ | Derivative gain of spring controller | 10.0 |
| $K_{i,sp}$ | Integral gain of spring controller | 5.0 |
| $r$ | Length of handle | 0.04 m |
| $g_{dis}$ | Cut-off frequency of DOB | 100.0 rad/s |
| $dt$ | Sampling time | 150 µs |

Figure 4.13 and Figure 4.14 show the results of bilateral controller. A balloon has been used as the slave environmental object. In Figure 4.13 both master and slave follow same positions according to (4.2). The torque command exerted by the master operator and the torque response from the balloon obey the law of action and reaction according to (4.1). The applied torque by the master operator is positive and torque response from the slave is negative but equal in magnitude as in Figure 4.14. Shaded area represents the free motion of both master and slave before contacting the slave actuator handle with the slave environment.



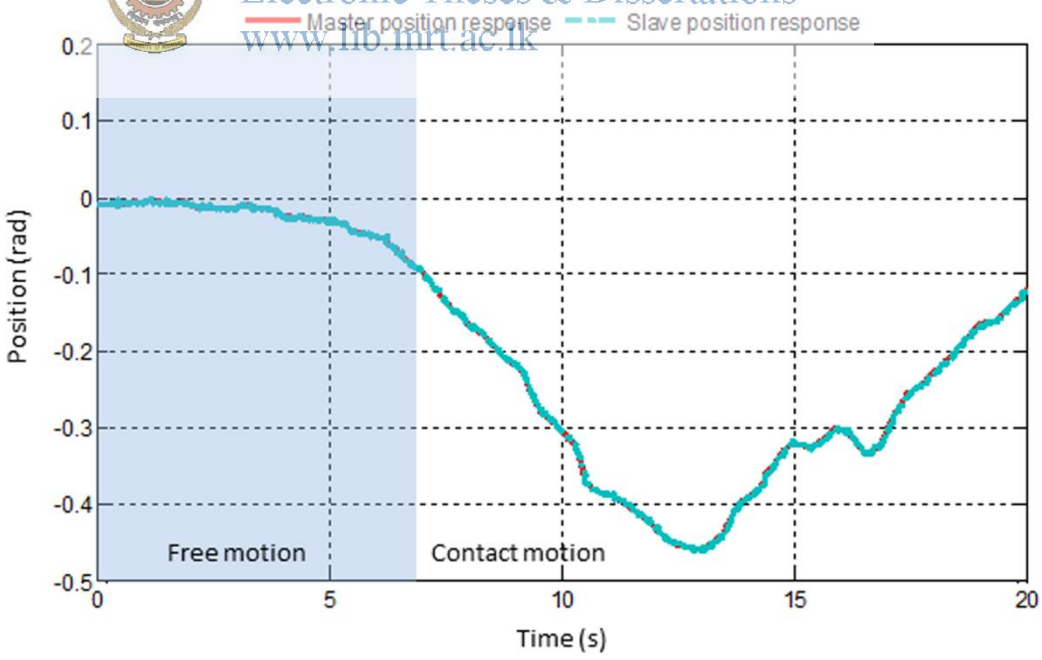



Figure 4.13 Master and slave position responses in bilateral control

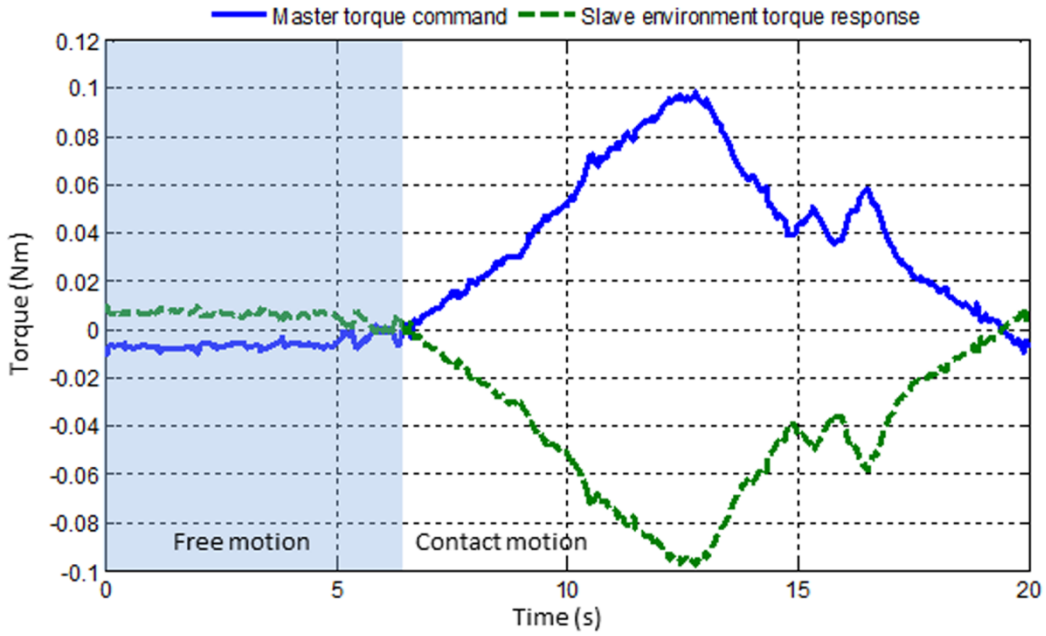



Figure 4.14 Torque command by the master operator and torque response from the slave environment in bilateral contol

Figure 4.15 and Figure 4.16 shows the results of torque lock. The limit of 0.10 Nm is used on the slave environment object. The results are taken using Aluminium block as the slave environmental object. The locked object at the torque limit is released when the release condition $(x_{eq}^{s} - x_m < \mathbf{0})$ & $(\hat{t}_{ext,m} < T_{limit})$ is satisfied by the master operator. Figure 4.15 shows the force applied by the master operator and the response from the slave environmental object. Near 19s the reaction torque from the slave environment reaches the torque limit and hence torque lock is activated. The torque remains constant in the slave side near 36s until master operator satisfies the release conditio $(x_{eq}^{s} - x_m < \mathbf{0})$ & $(\hat{t}_{ext,m} < T_{limit})$n. During the torque lock master operator is free to increase his holding torque with virtual spring. The vibration notification is shown at the time of force lock engaging and it available only for 500ms. The magnitude of vibration signal is sufficient to feel by the operator and it is small to cause system instability. The vibration notification signal or increasing of holding torque beyond defined force limit on the master side has not affected the torque lock on the slave side. The shaded areas in following figures represent bilateral control before torque lock engage and after torque lock release.

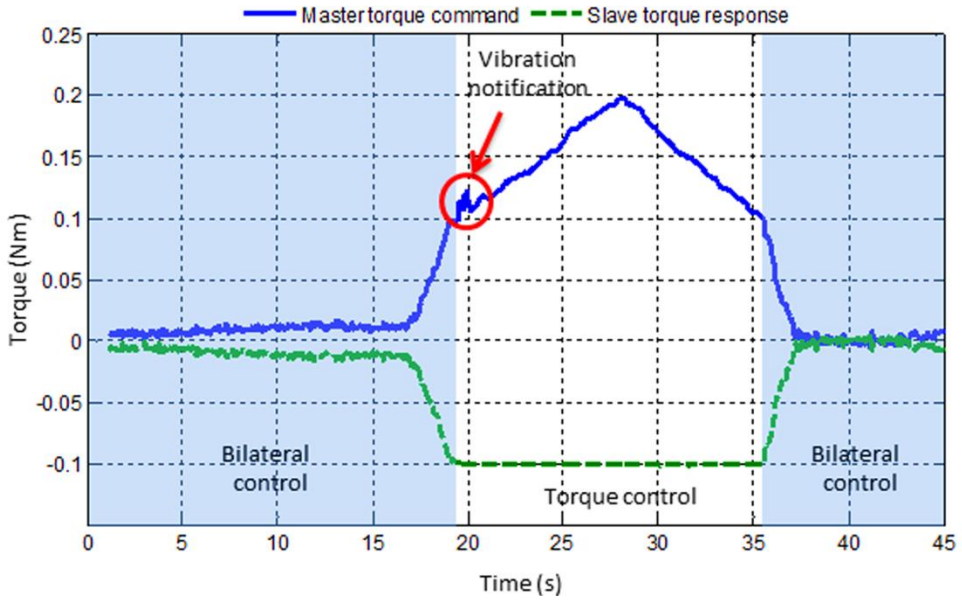



Figure 4.15 Master and slave torque responses in bilateral gripper with torque lock

Figure 4.16 shows the master and slave position response corresponding to applied torques in Figure 4.15. Both master and slave follow same positions during bilateral control nearly first 19s. When the torque lock is active slave actuator maintains constant position where master position is changing with the master operator holding torque adjustments. The vibration notification also can be observed. When the release condition $(x_{eq}^{s} - x_m < \mathbf{0}) \,\&\, (\hat{t}_{ext,m} < T_{limit})$ satisfied by the master operator again master and slave follow same positions obeying bilateral control



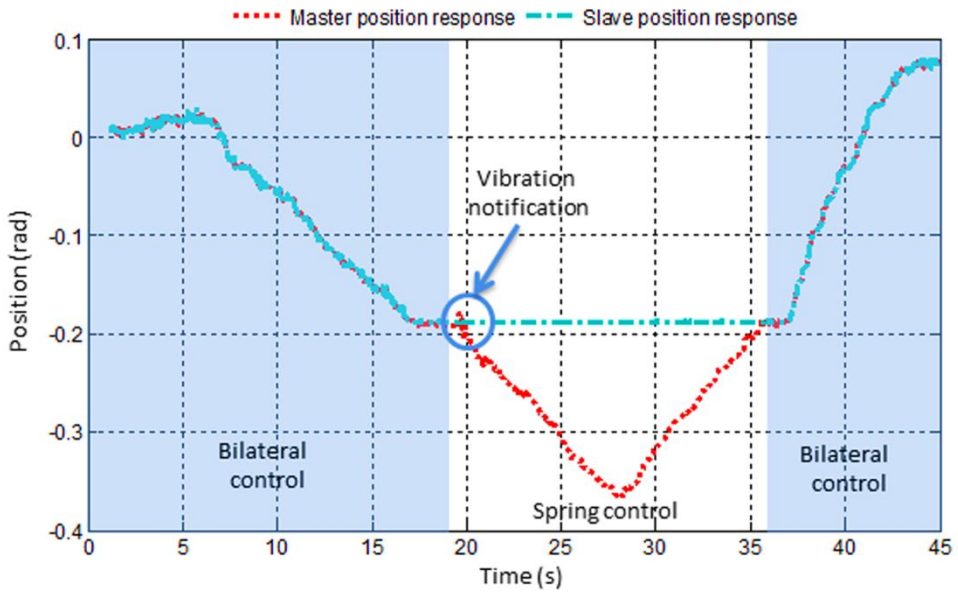



Figure 4.16 Master and slave position responses in bilateral gripper

Figure 4.17 represents torque variation and position responses on the same figure for the comparison. The slave torque response has been inverted.

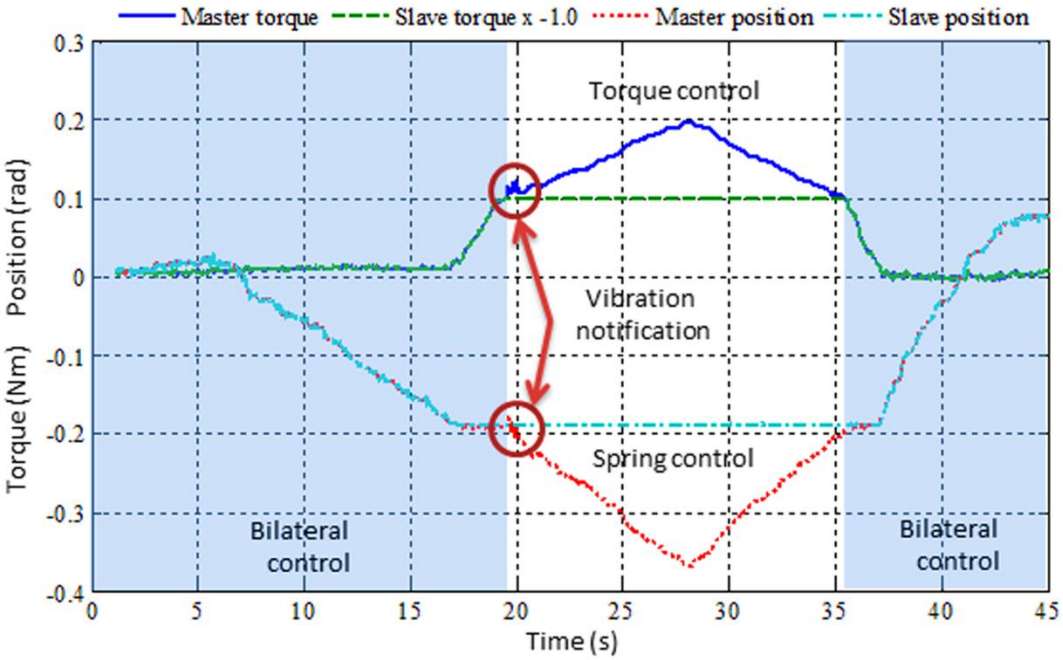



Figure 4.17 Master and slave torque and position responses with torque lock for comparison

# CHAPTER 5

# 5 VARIABLE IMPEDANCE MODEL

## 5.1 Environmental Object Model

The environment which is the object interested in the study is modeled as a simple linear mass, damper and spring model and environment reaction for operator force can be represented as (5.1) and, (5.2) denotes its Laplace representation. The relationship between environment reaction and position denotes the environmental impedance as (5.3). These parameters are expected to be calculated using the 3D data during the experiment.

$$F_e = m_e\ddot{x} + b_e\dot{x} + k_e x = F_m + F_b + F_k = F_{op} \quad (5.1)$$

$$F_e = (m_{(e)}s^2 + b_{(e)}s + k_{(e)}).x \quad (5.2)$$

$$F_e = Z_e.x \quad (5.3)$$

The notations which have been used in the environmental object modeling are listed below.

$F_e$ : Environment reaction force

$m_e$ : Environmental mass

$b_e$ : Environmental damping coefficient

$k_e$ : Environmental spring coefficient

$F_m$ : Force from the mass

$F_b$ : Force from the damper

$F_k$ : Force from the spring

$F_{op}$ : Operator force

$x$ : Position change of the environment

$\dot{x}$ : Compression velocity

$\ddot{x}$ : Compression acceleration

$Z_e$ : Environmental impedance

$s$ : Laplace operator

The environment model in (5.1) assumes constant $m_{(e)}$,$b_{(e)}$, and $k_{(e)}$. This chapter proposes an impedance model as a function of its motion parameters as (5.4).

$$F_e = m_{(\ddot{x})}\ddot{x} + b_{(\dot{x})}\dot{x} + k_{(x)}x \quad (5.4)$$

It is proposed that $m_e, b_e$ and $k_e$ are function of compression depth $x$, compression velocity $\dot{x}$ and compression acceleration $\ddot{x}$ as $m_{(\ddot{x})}, b_{(\dot{x})}$ and $k_{(x)}$. Therefore it is possible to identify and estimate variable environmental impedance through this modeling. Figure 5.1 shows the model of proposed variable impedance model.

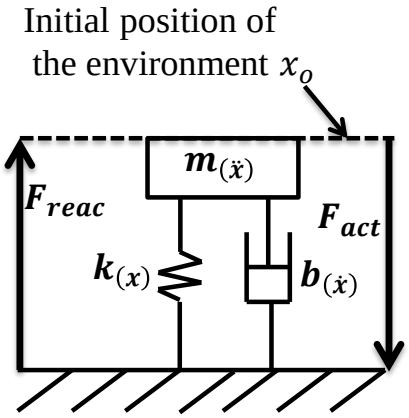



Figure 5.1 Environmental model of mass, spring, damper.

## 5.2 Estimation of Environmental Object Impedance



The estimation of $k_{(x)}, b_{(\dot{x})}$ and $m_{(\ddot{x})}$ is done as follows. The spring coefficient $k_{(x)}$ variation with compression depth $x$ is estimated using a position controller. A series of position commands, from 4mm to 24mm with steps of 4mm, similar to Figure 5.2 is applied on environment surface. The data is acquired when actuator maintains a stationary position. Because $x$ is stable $\dot{x}, \ddot{x}$ is approximated to be zero and reaction force from the environment obeys (5.5).

$$F_e = k_{(x)}x = F_k \quad (5.5)$$

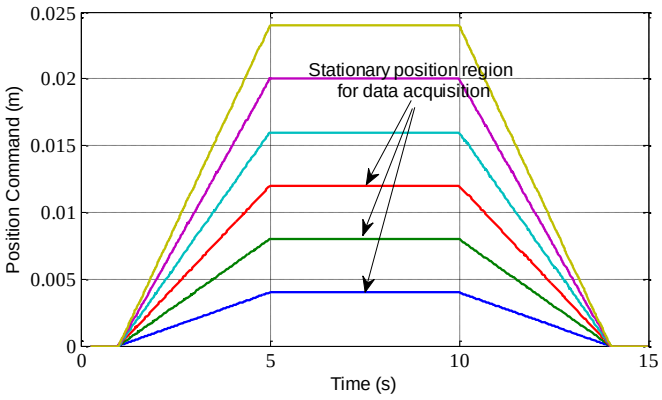



Figure 5.2 Position command profiles on the environment

Once the variation of $k$ with $x$ is obtain actuator is operated with series of different constant velocity commands similar to Figure 5.3. Each velocity command corresponds to 20mm compression depth. Since $\dot{x}$ is constant $\ddot{x}$ is approximated to be zero and reaction force from the environment obeys (5.6).

$$F_e = b_{(\dot{x})}\dot{x} + k_{(x)}x = F_b + F_k \tag{5.6}$$

Since variation of $k_{(x)}$ with $x$ is obtained in previous step and $F_e$ with respect to $\dot{x}$ are recorded in this step, the behaviour of $b$ with $\dot{x}$ can be estimated using (5.6).

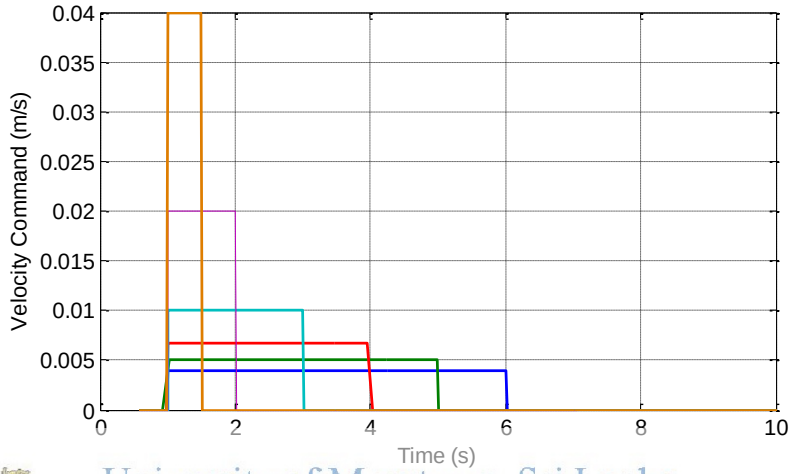





Figure 5.3 Constant velocity command profiles on the environment

Next, ramp velocity commands similar to Figure 5.4 are applied on the environment and reaction force from the environment obeys (5.4). Since variation of $k_{(x)}$ with $x$ and variation of $b_{(\dot{x})}$ with $\dot{x}$ for corresponding $x$ values are known, using (5.4) the behaviour of $m$ with $\ddot{x}$ can be estimated.

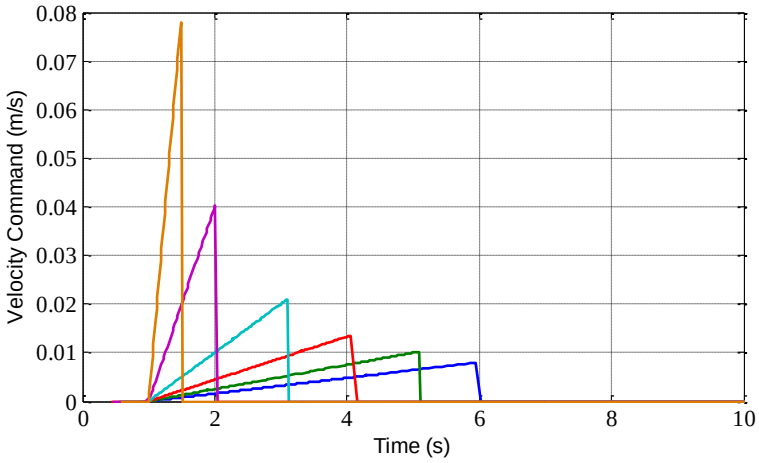



Figure 5.4 Ramp velocity command profiles on the environment.

Reaction force estimation for above steps is possible by Reaction Torque Observer (RTOB). These environmental impedances are calculates as a discrete time modeling.

The reaction force from the environment with compression depth $x$, can be represented by a relationship as (5.7).

$$[F_{reac,x}] = \left[ diag\left( [m_{(\ddot{x})} \quad b_{(\dot{x})} \quad k_{(x)}] \begin{bmatrix} \ddot{x} \\ \dot{x} \\ x \end{bmatrix} \right) \right]^T \tag{5.7}$$

Where,

$$[F_{reac,x}] = \begin{bmatrix} F_{reac,x_1} \\ F_{reac,x_2} \\ \vdots \\ F_{reac,x_n} \end{bmatrix}, m_{(\ddot{x})} = \begin{bmatrix} m_{(\ddot{x_1})} \\ m_{(\ddot{x_2})} \\ \vdots \\ m_{(\ddot{x_n})} \end{bmatrix}, b_{(\dot{x})} = \begin{bmatrix} b_{(\dot{x_1})} \\ b_{(\dot{x_2})} \\ \vdots \\ b_{(\dot{x_n})} \end{bmatrix}, k_{(x)} = \begin{bmatrix} k_{(x_1)} \\ k_{(x_2)} \\ \vdots \\ k_{(x_n)} \end{bmatrix},$$

$$\ddot{x} = [\ddot{x}_1 \quad \ddot{x}_2 \quad \dots \quad \ddot{x}_n], \quad \dot{x} = [\dot{x}_1 \quad \dot{x}_2 \quad \dots \quad \dot{x}_n], \quad x = [x_1 \quad x_2 \quad \dots \quad x_n]$$

Here, 1, 2, …,n denote the number of compression depths considered in the study.

## 5.3 Experimental Results



The slave actuator of the bilateral hardware setup is used to study the environmental object impedance model as a function of its motion parameters variation. The experimental parameters are listed in Table 5.1.

Table 5.1 Experimental parameters

| Parameter | Description | Value |
|---|---|---|
| $K_{pp}$ | Proportional gain of position controller | 650.0 |
| $K_{pd}$ | Derivative gain of position controller | 250.0 |
| $K_{pi}$ | Integral gain of position controller | 80.0 |
| $K_{vp}$ | Proportional gain of velocity controller | 150.0 |
| $K_{vd}$ | Derivative gain of velocity controller | 200.0 |
| $K_{vi}$ | Integral gain of velocity controller | 80.0 |
| $K_{tn}$ | Motor torque constant | 0.135 Nm/A |
| $J_n$ | Motor inertia | 0.00091 kgm$^2$ |
| $g_{dis}$ | Cut-off frequency of disturbance observer | 200 rad/s |
| $dt$ | Sampling time | 150 µs |

The experimental PID gain values of position controller and velocity controller are tuned until actuator response follows the position and the velocity commands given by the controller.

The experiment is carried out using a balloon and a rubber sponge as the environmental object. First, a constant force command is applied on the balloon and the rubber sponge and recorded its force position variation is as shown in Figure 5.5.

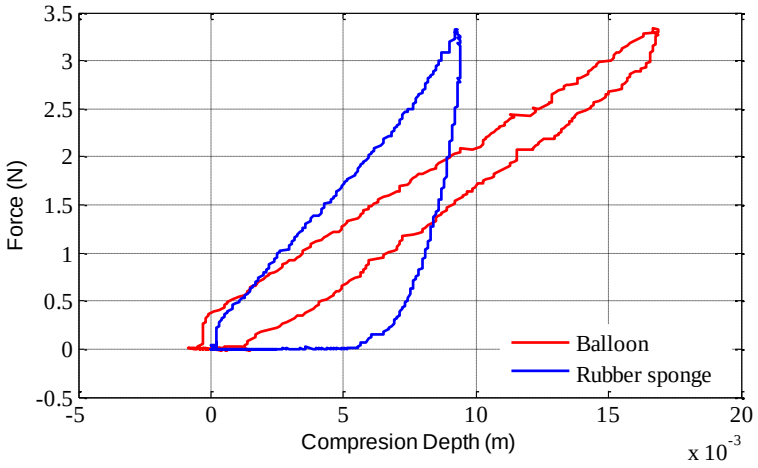



Figure 5.5 Force-position variation comparison of balloon and rubber sponge



This result shows that for the same operator force environmental object reaction is different for different objects which mean the object impedances are different. Since plotted area for the rubber sponge is larger than for the balloon it indicates that the rubber sponge has more hysteresis loss. The area enclosed by the graph represents the energy absorbed by the object which was tested. The gradient of the graph shows its impedance as Figure 5.6.

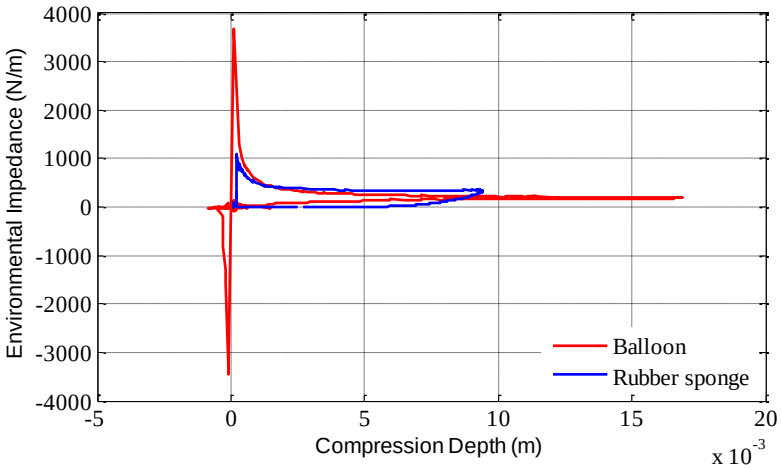



Figure 5.6 Impedance variation of balloon and rubber sponge

A constant force command is applied along the surface of the balloon and environmental impedance variation is shown in Figure 5.7. It shows that the impedance of the middle area is high and it reduces towards the edges of the balloon.

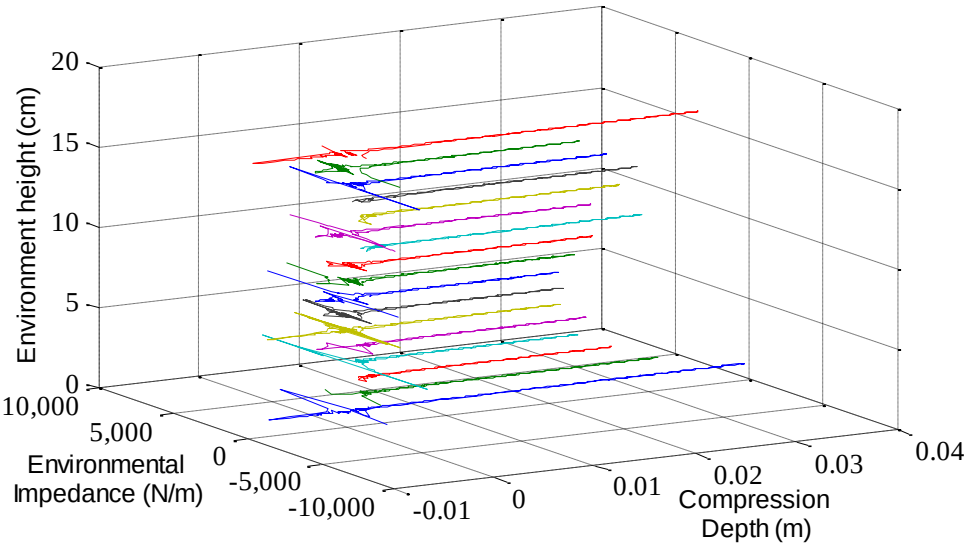



Figure 5.7 Balloon environmental impedance variation with its height

### 5.3.1 Environmental impedance variation with motion parameters



A middle point on the environment (balloon) is considered for further study. Estimation of environmental object impedance variation as its motion parameters require series of commands with position, step velocity and ramp velocity which were shown in Figure 5.2, Figure 5.3 and Figure 5.4, according to (5.5), (5.6) and (5.4). The actuator responses for position, step velocity and ramp velocity are shown in Figure 5.8, Figure 5.9 and Figure 5.10 respectively. It verifies that actuator follows the controller.

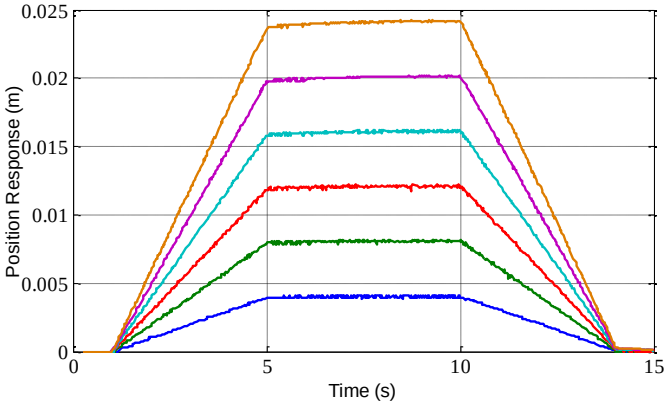



Figure 5.8 Position response of the actuator for environmental spring coefficient

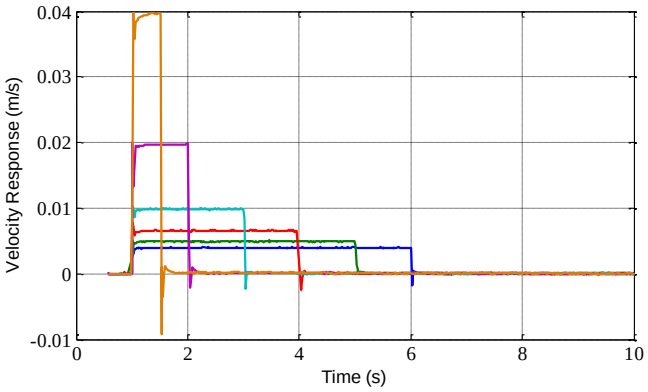



Figure 5.9 Step velocity response of the actuator for environmental damping coefficient estimation

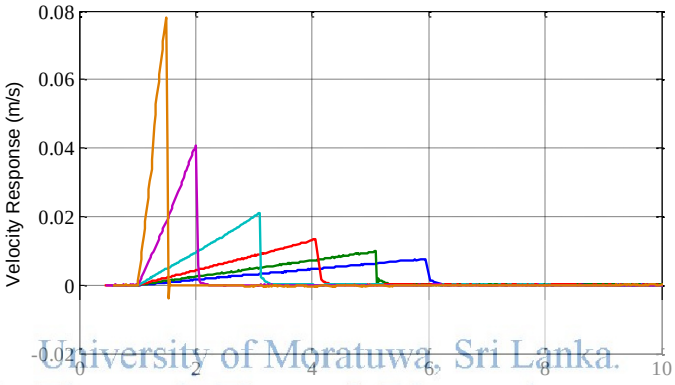





Figure 5.10 Ramp velocity response of the actuator for environmental mass coefficient estimation

The balloon is compressed up to 0.024m in the steps of 0.004m and RTOB response is recorded during the stationary position region and calculated variation of spring constant is shown in Figure 5.11.

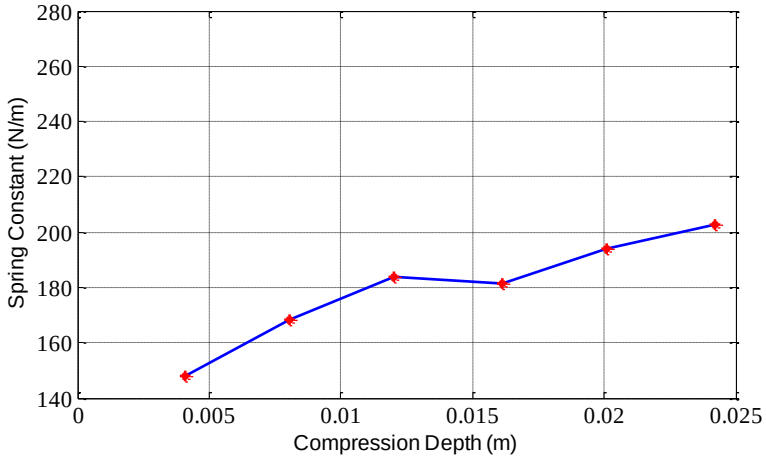



Figure 5.11 Spring coefficient variation of balloon with compression depth.

Set of step velocities are applied on the balloon which results in 0.020m compression depth in each velocity command. The variation of damping coefficient is calculated based on the recorded RTOB response with velocity and result of spring coefficient with compression depth obtained in the previous step. The calculated variation of damping coefficients for the balloon is shown in Figure 5.12.

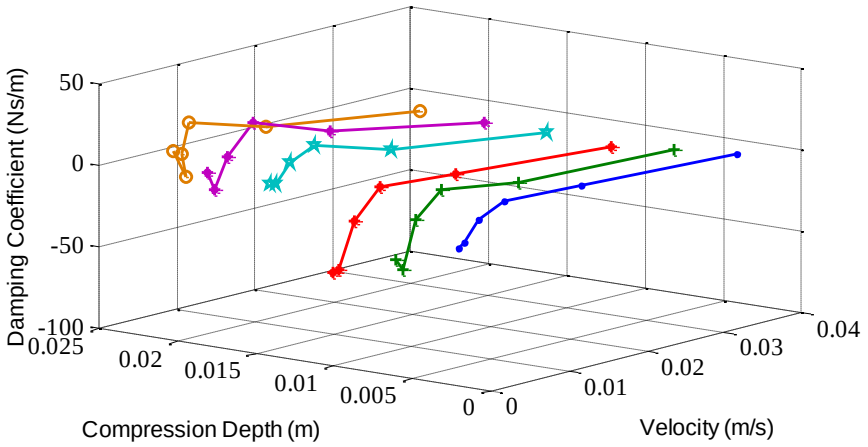



Figure 5.12 Damping coefficient variation of balloon with velocity and compression depth



Next, ramp velocity command which results in 0.020m compression depth in each ramp velocity command is applied on the balloon and variation of mass with acceleration is calculated using above derived estimated spring coefficients and damping coefficients. Calculated mass variation of balloon with acceleration is shown in Figure 5.13.

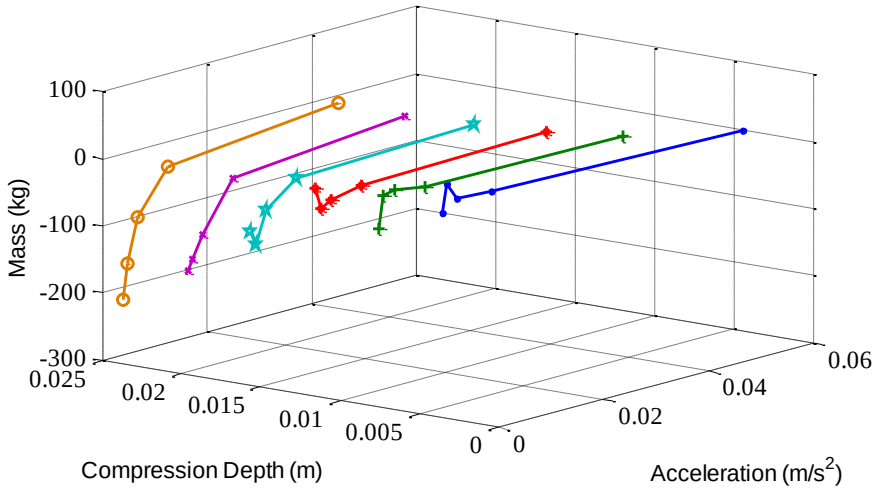



Figure 5.13 Calculated mass variation of balloon with acceleration and compression depth

We could observe from Figure 5.11 that the spring coefficient of the balloon is not constant. It shows a nonlinear variation with compression depth. Therefore assuming constant spring, mass, and damper coefficients during conventional object reaction modeling as in (5.1) is not valid for real environmental objects. According to Figure 5.12 estimated damping coefficient values of balloon are negative due to bounce back effect of the balloon. The above procedure is repeated for the rubber sponge and results of calculated spring constant, damping coefficient, and mass are shown in Figure 5.14, Figure 5.15 and Figure 5.16 respectively.

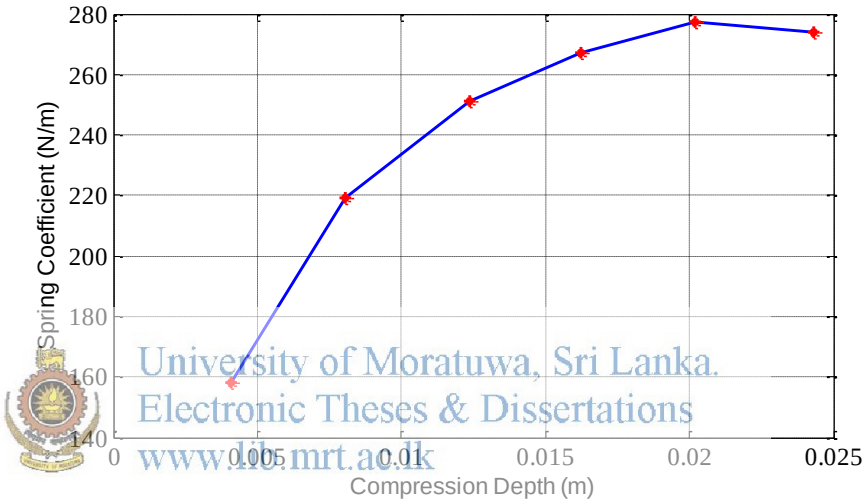




Figure 5.14 Spring coefficient variation of rubber sponge with compression depth

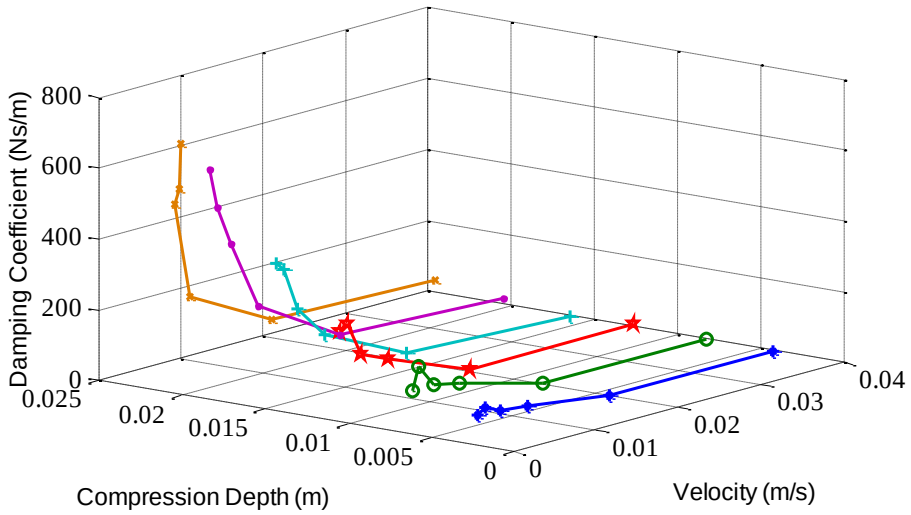



Figure 5.15 Damping coefficient variation of rubber sponge with velocity and compression depth

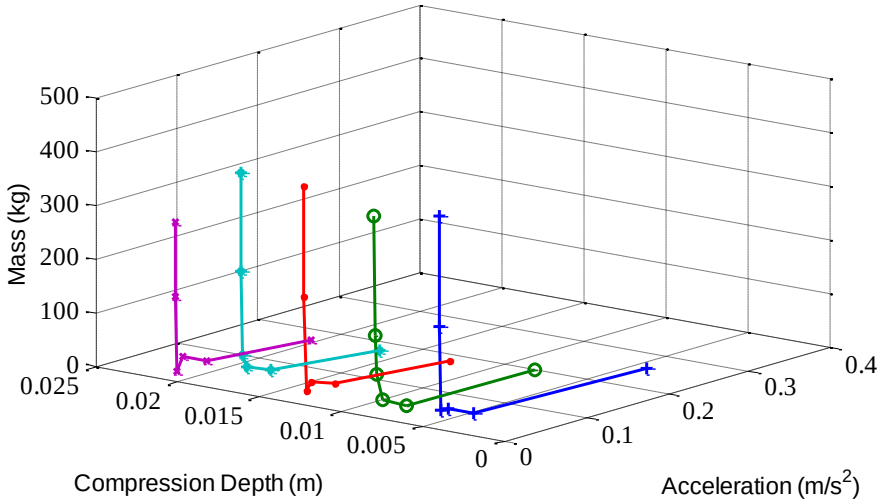



Figure 5.16 Calculated mass variation of rubber sponge with acceleration and compression depth.

When comparing two environments, results show that rubber sponge has high environmental impedance. The negative values of damping coefficient of balloon indicate the bounce back effect. The calculated mass of both balloon and rubber sponge varies in a considerable range which is beyond the actual mass of them. It reaches a constant value as the compression acceleration increased.

# CHAPTER 6

# 6 CONCLUSION

In this research single DOF force lock gripper operation and gripped object impedance variation with gripper motion parameters are studied.

An enhanced gripping mechanism to be used with the bilateral teleoperation was proposed. The proposed bilateral controller is exemplified for a single DOF rotary actuator at the slave side. However, if multiple DOF manipulators are used in bilateral teleoperation, locked object could be placed safely to the required position with minimal hazel where the operator need not to worry about the force limit of the object. The force limit of the object is set by the operator based on the experience and the object properties. The force lock protects the gripped object on the slave and the attainment of the force limit is notified to the master operator via a small vibration. In addition to the vibration the loss of reaction force coming from the slave environment could also be sensed by the master operator. The virtual spring which is attached to master side during active force lock in the slave side, attempts to equalize the master slave positions in the locked mode. The proposed system is modeled and simulated using MATLAB m code using real world parameters. Also it is tested with hardware setup.



But in real application the loss of reaction force coming from the slave environment which results in position step in the master side, panic the master operator. The modified virtual spring controller with added object force limit reference facilitates operator comfort as continuous force increasing beyond the defined environmental force limit if operator presses his lever towards force increasing direction. The results verify the applicability of new concept to bilateral teleoperation.

This research proposed a variable impedance model for environmental object representation. The Environment object impedance varies along the surface. The results of spring coefficient as a function of compression depth, damping coefficient as a function of compression depth and compression velocity, mass as a function of compression depth and compression acceleration show environmental impedance variation with different motion parameters. For a balloon damping coefficient gives a negative value which is not practicable as a result of the bounce back effect. As

damping coefficient is negative calculated mass is negative for the balloon for the given region. Calculated mass value does not represent a practical value due to the nonlinear behaviour of the responses. Conventionally used simple environmental impedance model is valid only for spring damper applications. If a real world object is expected to be modeled for an application like skill preservation the proposed impedance model as a function of its motion parameters should be used.

# REFERENCES


[1]. A.M.H.S.Abeykoon, and K.Ohnishi, "Tactile sensation improvement of a bilateral forceps robot with a switching virtual model," in *10th IEEE Int. Workshop on Advanced Motion Control, 2008*,pp.526-531, doi: 10.1109/AMC.2008.4516122

[2]. N. Murata, and S. Katsura, "Bilateral force feedback control using multi-degree-of-freedom mobile robot," in *2011 Proc. of SICE Annu. Conf.* , 2011,pp.1041-1046.

[3]. A.Hace, and M.Franc, "Pseudo-Sensorless High-Performance Bilateral Teleoperation by Sliding-Mode Control and FPGA," in *IEEE/ASME Trans. on Mechatronics*, vol.19, no.1, pp.384,393, Feb. 2014

[4]. L.Chan; F.Naghdy, and D.Stirling, "Application of Adaptive Controllers in Teleoperation Systems: A Survey," in *IEEE Trans. on Human-Machine Systems*, , vol.44, no.3, pp.337,352, June 2014. doi: 10.1109/THMS.2014.2303983

[5]. A.M.H.S.Abeykoon, and K.Ohnishi, "Realization of Virtual Slave Model of a Forceps Robot Using Bilateral Control," in *32nd Annu. Conf. on IEEE Industrial Electronics*,2006, pp.4468-4473. doi: 10.1109/IECON.2006.348100

[6]. D.T. Pham and E.Tacgin, "An Expert system for selection of robot grippers," *Expert Syst. Appl.*, vol.5, pp. 289-300, 1992.

[7]. M. K. Brown, "A Controlled Impedance Robot Gripper" in *AT&T Technical J.*,1985, vol. 64, no. 4, pp. 937-969.

[8]. J.M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K.J. Kuchenbecker, "Human-Inspired Robotic Grasp Control With Tactile Sensing," in *IEEE Trans. on Robotics*, vol.27, no.6, pp.1067-1079, doi: 10.1109/TRO.2011.2162271.





[9]. J. Becedas, I. Payo, and V. Feliu, "Two-Flexible-Fingers Gripper Force Feedback Control System for Its Application as End Effector on a 6-DOF Manipulator," in *IEEE Trans. on Robotics*, 2011, vol.27, no.3, pp.599-615, doi: 10.1109/TRO.2011.2132850.

[10]. M. Florescu, D. Strimbeanu, and D.M. Matei, "Force control of grasping operation for a tentacle robot," in *18th Int. Conf. on System Theory, Control and Computing*, 2014, pp.832-837, doi: 10.1109/ICSTCC.2014.6982522.

[11]. M. Goldfarb, and N. Celanovic, " A flexure-based gripper for small-scale manipulation" on *Robotica*, vol.null, pp. 181-187, Sept. 2000.

[12]. J.Y. Wang, and C.C. Lan, " A Constant Force Compliant Gripper for Handling Objects of Various Sizes" in *ASME Trans. of J. of Mechanical Design*, 2014, vol. 136.

[13]. M. Kamali, S.A.A. Moosavian, and F. Cheraghpour, "Improving grasp capabilities of KNTU hand using position & force sensors," in *22nd Iranian Conf. on Electrical Engineering*, 2014, pp.1278-1283, doi: 10.1109/IranianCEE.2014.6999731

[14]. R.D. Lorenz, K.M. Meyer, and D.M.V De Riet, "A novel, compliant, four degree-of-freedom, robotic fingertip sensor," in *IEEE Trans. on Industry Applications*, 1990, vol. 26, no. 4, pp. 613-619.

[15]. T.L. Gibo, D.R. Deo, Z.F. Quek; Okamura, A.M., "Effect of load force feedback on grip force control during teleoperation: A preliminary study," in *IEEE Haptics Symp.(HAPTICS)*, 2014 , pp.379-383, doi: 10.1109/HAPTICS.2014.6775485.

[16]. G.S. Gupta, S.C. Mukhopadhyay, C.H. Messom, and S.N. Demidenko, "Master–Slave Control of a Teleoperated Anthropomorphic Robotic Arm With Gripping Force Sensing," in *IEEE Trans. on Instrumentation and Measurement* , 2006, vol.55, no.6, pp.2136-2145, doi: 10.1109/TIM.2006.884393.

[17]. R.M. Pierce, E.A. Fedalei, and K.J. Kuchenbecker, "A wearable device for controlling a robot gripper with fingertip contact, pressure, vibrotactile, and grip force feedback," *IEEE Haptics Symp.*, 2014, pp.19-25, doi: 10.1109/HAPTICS.2014.6775428.

[18]. Y. Yokokura, S. Katsura, and K. Ohishi, "Motion copying system based on real-world haptics," in *10th IEEE Int. Workshop on Advanced Motion Control*, pp. 613-618, Mar. 2008. doi: 10.1109/AMC.2008.4516137

[19]. S.Katsura, W. Yamanouchi, and Y. Yokokura, "Real-World Haptics: Reproduction of Human Motion," in *IEEE Ind. Electronics. Mag.*, vol.6, no.1, pp.25-31, Mar. 2012. doi: 10.1109/MIE.2012.2182854

[20]. A.M.H.S.Abeykoon, and K.Ohnishi, "Virtual tool for bilaterally controlled forceps robot-for minimally invasive surgery," in *The Int. J. of Medical Robotics and Comput. Assisted Surgery*, vol.3, no.3, pp. 271-280, Aug 2007.

[21]. A.M.H.S. Abeykoon, and K. Ohnishi, "Bilateral Control interacting with a Virtual Model and Environment," in *IEEE Int. Conf. on Ind. Technology*, pp.1320-1325, Dec. 2006. doi: 10.1109/ICIT.2006.372520



[22]. T. Shimono, S. Katsura, and K. Ohnishi, "Abstraction and Reproduction of Force Sensation From Real Environment by Bilateral Control,"in *IEEE Trans. on Ind. Electronics*, vol.54, no.2, pp.907-918, April 2007. doi: 10.1109/TIE.2007.892744

[23]. N. Tsunashima, and S. Katsura, "Spatiotemporal Coupler: Storage and Reproduction of Human Finger Motions,"in *IEEE Trans. on Ind. Electronics*, vol.59, no.2, pp.1074-1085, Feb. 2012. doi: 10.1109/TIE.2011.2161247

[24]. S. Yajima, and S. Katsura, "Multi-DOF Motion Reproduction Using Motion-Copying System with Velocity Constraint,"in *IEEE Trans. on Ind. Electronics*, vol.61, no.7, pp.3765-3775, July 2014. doi: 10.1109/TIE.2013.2286086

[25]. T. Shimono, S. Katsura, and K. Ohnishi, "Reproduction of Real World Force Sensation by Bilateral Motion Control Based on Contact Impedance Model Taking Environmental Hysteresis into Account,"in *IEEE Int. Conf. on Mechatronics*, pp.613-618, July 2006. doi: 10.1109/ICMECH.2006.252596

[26]. T.Murakami, K.Ohnishi, " Observer Based Motion Control- Application to Robust Control and Parameter Identification," in *IEEE trans. on Ind. Electronics.*,1993

[27]. Y.S.E. Ali, S.B.M. Noor, S.M. Bashi, and M.K. Hassan,"Microcontroller performance for DC motor speed control system," in *Proc.of Nat. Power Engineering Conf.*, pp. 104- 109, 2003.

[28]. G. Huang, and S. Lee, "PC-based PID speed control in DC motor," in. *Proc. of the Int.Conf. on Audio, Language and Image Processing*, pp 400-407, July 2008

[29]. M. Mizuochi, T. Tsuji, and K. Ohnishi, "Improvement of disturbance suppression based on disturbance observer," in *9th IEEE Int. Workshop on Advanced Motion Control*, pp.229-234, 2006.

[30]. B.J.A. Helouvry, P. Dupont, and C.De Wit, "A survey of models, analysis tools and compensation method for the control of machines with friction,"in *Automatica*, Vol. 30, No.7 , pp. 1083-1138,1994

[31]. K. Ohnishi, M. Shibata, and T. Murakami : "Motion control for advanced mechatronics," in *IEEE/ASME Trans. on Mechatronics*, , vol.1, no.1pp.56-67, Mar 1996.

[32]. S. Katsura, K. Irie, and K. Ohishi, "Wideband Force Control by Position-Acceleration Integrated Disturbance Observer," in *IEEE Trans. on Ind. Electronics*, vol.55, no.4, pp.1699-1706, April 2008.

[33]. T.Murakami, K.Ohnishi, (1993) "Disturbance Observer Based Motion Control- Application to Robust Control and Parameter Identification," in *IEEE trans. on Ind. Electronics*.

[34]. S. Katsura, Y. Matsumoto, and K. Ohnishi : "Modeling of force sensing and validation of disturbance observer for force control," in *29th Ann. Conf. of the IEEE Ind. Electronics Society*, vol.1, pp. 291-296, 2003.

[35]. K.Ohishi, and K.Miyachi, Torque-speed regulation of DC motor based on load torque estimation," *IEEJ IPEC-Tokyo*, vol.2, pp.1209-1216,1983.

[36]. A. Sabanovic, and K. Ohnishi (2011) "*Motion control systems,"* in IEEE press John Willey and sons (Asia) pte Ltd, (First Edition).,2011

[37]. A.M H.S. Abeykoon, and K. Ohnishi: "Improvement of Tactile Sensation of a Bilateral Forceps Robot by a Switching Virtual Model," in *Trans.on Advanced Robotics*", Vol. 8, pp. 789-806 , 2008.

[38]. A.M.H.S. Abeykoon, and K. Ohnishi, "Virtual tool for bilaterally controlled forceps robot-for minimally invasive surgery," in *Trans. on int. J. of Medical Robotics and Computer Assisted Surgery*", ISSN 1478-5951, vol 3; No 3, pp. 271-280, 2007.

[39]. Y.S.E. Ali, S.B.M. Noor, S.M. Bashi, and M.K. Hassan, "Microcontroller performance for DC motor speed control system," in *Proc.of Nat. Power Engineering Conf*, 104- 109, Dec. 2003.

[40]. https://developer.mbed.org/users/hexley/notebook/qei_hw-interface-implementation-notes/

[41]. T. Shimono, S. Katsura, and K. Ohnishi, "Abstraction and Reproduction of Force Sensation From Real Environment by Bilateral Control,"in *IEEE Trans. on Ind. Electronics*, vol.54, no.2, pp.907-918, April 2007. doi: 10.1109/TIE.2007.892744

[42]. Y. Yokokohji, and T. Yoshikawa, “Bilateral control of Master-slave Manipulator for ideal Kinesthetic Coupling- Formulation and Experiment,” in *IEEE Trans. on Robotics and Automation* vol. 10, no. 5, pp 605-620,1994

[43]. M. K. C. D. Chinthaka, “Position Based Friction Estimation for Precise Motion Control”, University of Moratuwa, M.Sc thesis, July 2014

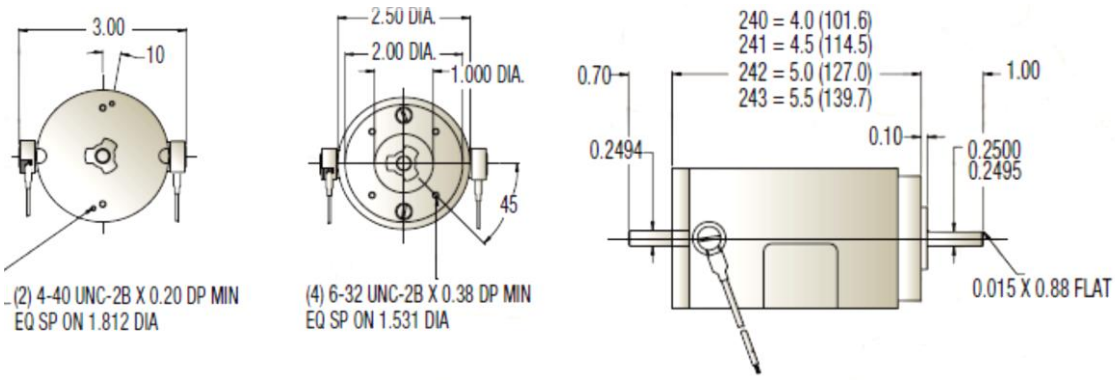
[44]. B. M. Pillai, “Parameters Estimation for Motion Control and Friction Compensation”, University of Moratuwa, M.Sc thesis, August 2013

# APPENDIX

## APPENDIX I : Specification of the motor

| | | |
|---|---|---|
| Manufacturer | : | Electrocraft |
| Version | : | E240 |
| Maximum Terminal Voltage | : | 60V DC |
| Supply Voltage | : | 32V DC |
| Continuous stall Torque | : | 20.5 Ncm |
| Peak Torque | : | 169.5 Ncm |
| Maximum Speed | : | 5000 rpm |
| Rotor Inertia | : | 0.268 $Kgcm^2$ |
| Maximum Friction Torque | : | 2.1 Ncm |
| Weight | : | 1Kg |
| Torque Constant | : | 13.5 Ncm/Amp |
| Terminal Resistance | : | 5.4 Ω |
| Armature Inductance | : | 8.2 mH |

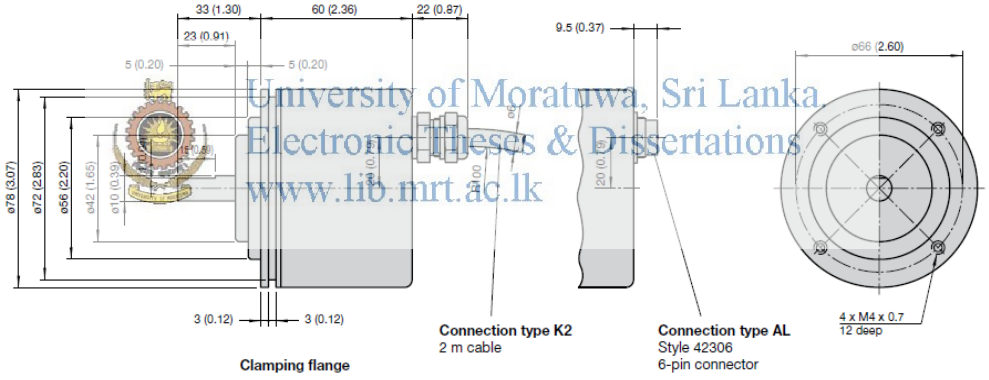





Dimensions of the motor

**APPENDIX II : Specification of the master encoder**

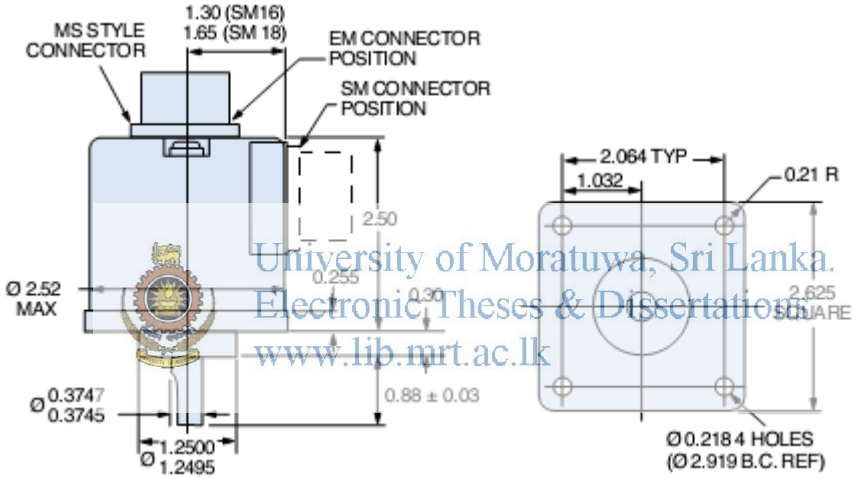
| | | |
|---|---|---|
| Pulse Count | : | 5,000 ppr (pulse per revolution) |
| Pulse count with QEI | : | 20,000 ppr |
| Output Frequency | : | ≤ 100 kHz |
| Supply Voltage | : | 10-30 VDC |
| Applied voltage | : | 12 VDC |
| Output | : | Push-pull |
| Current Consumption | : | ≤ 80 mA |
| Load Current | : | ≤ 40 mA, short circuit, |
| Voltage Drop | : | < 4 V |
| Response Time | : | 250 ns |
| Rotational Speed | : | < 6000 rpm |
| Moment of Inertia | : | ≤ 1.4 x $10^{-3}$ oz-in-$sec^2$ |




Dimensions of the master encoder

**APPENDIX III : Specification of the slave encoder**

| | | |
|---|---|---|
| Model | : | H25D (BEI) |
| Pulse Count | : | 2,500 ppr (pulse per revolution) |
| Pulse count with QEI | : | 10,000 ppr |
| Supply Voltage | : | 5 to 28 VDC available |
| Applied voltage | : | 12 VDC |
| Current Requirements | : | 100 mA typical +output load, 250 mA (max) |
| Voltage/Output | : | 28V/5: Line Driver, 5–28 VDC in, V out= 5 VDC |
| Frequency Response | : | 100 kHz, up to 1MHz with interpolation option |
| Moment of Inertia | : | 5.2 X $10^{-4}$oz-in-$sec^2$ |




Dimensions of the slave encoder

**APPENDIX IV : Specification of the coupler**

| | | |
|---|---|---|
| Shaft usage | : | 8mm x 8mm (0.315" x 0.315") |
| Length | : | 25mm |
| Diameter | : | 18mm |
| Material | : | Aluminum |
| Rated Torque | : | 1N.m |
| Max. Torque | : | 2N.m |
| Eccentricity Error | : | ±0.2mm |
| Shaft Angel | : | ≤2° |
| Max. Rotational(RPM) | : | 19000 |
| Rated | : | In-Phase Operate |

## APPENDIX V: MATLAB m code

```
function [x_response_s]=BC_D_MandS(t)
%PD controller Block parameters of master
Kp_m=350;
Kd_m=25;
PD_out_m=0.0;
filter_pd_m=0.0;
filter_pd_m_pre=0.0;
g_pd_m=50.0;

%PD controller Block parameters of slave
Kp_s=350;
Kd_s=25;
filter_pd_s=0.0;
filter_pd_s_pre=0.0;
PD_out_s=0.0;
g_pd_s=50.0;

%Motor parameters of master
Mn_m=0.40;
Kfn_m=24.0;

%Motor parameters of slave
Mn_s=0.4;
Kfn_s=24.0;

%DOB & RTOB paremeter of master
g_dis_m=150.0;
to_filter_m=0.0;
filter_out_m=0.0;
filter_out_pre_m=0.0;
DOB_out_m=0.0;
friction_m=0.0;
RTOB_out_m=0.0;

%DOB & RTOB paremeter of slave
g_dis_s=150.0;
to_filter_s=0.0;
filter_out_s=0.0;
filter_out_pre_s=0.0;
DOB_out_s=0.0;
friction_s=0.0;
RTOB_out_s=0.0;

%load model
B=5;
M=0.1;
k=250;

%controller Block parameters of slave force controller
Kp_FC=3.3;
Kd_FC=1.1;
g_FC=150.0;
g_FC_r=150.0;
filter_FC=0.0;
```

```
filter_FC_pre=0.0;
FC_error=0.0;
filter_FC_r=0.0;
dF_error=0.0;
dF_error_r=0.0;
filter_FC_r_pre=0.0;
F_defined=9.0;
FC_out=0.0;

%spring controller master
k_sp=100.0;

%Constant parameter define
c=10.0;
torque_error=0.0;
x_response_m=0.0;
x_response_pre_m=0.0;
x_error_m=0.0;
dx_response_m=0.0;
dx_response_m_pre=0.0;
F_m=0.0;
I_error_m=0.0;
load_force_m=0.0;
force_error_m=0.0;
motor_acc_m=0.0;
motor_velocity_m=0.0;
motor_velocity_pre_m=0.0;
x_response_s=0.0;
x_response_pre_s=0.0;
x_error_s=0.0;
dx_response_s=0.0;
dx_response_s_pre=0.0;
F_s=0.0;
I_error_s=0.0;
load_force_s=0.0;
force_error_s=0.0;
motor_acc_s=0.0;
motor_velocity_s=0.0;
motor_velocity_pre_s=0.0;
j=0.0;

dt=0.0001;
A=zeros(500001,5);
i=0;

for t=0.0:0.0001:50
    i=i+1;
%External force profile for master
if (t>=0.0)&(t<5.0)
    load_force_m=0.05+0.02*t;
elseif (t>=5.0)&(t<10)
    load_force_m=1.97*t-9.7;
elseif (t>=10.0)&(t<15)
    load_force_m=c;
elseif (t>=15.0)&(t<25)
    load_force_m=-t+25;
elseif (t>=25.0)&(t<30)
```

```
    load_force_m=0.0;
elseif (t>=30.0)&(t<31)
    load_force_m=-0.05*t+1.5;
elseif (t>=31)&(t<33)
    load_force_m=0.05*t-1.6;
else
    load_force_m=0.05;
end

x_error_m=x_response_s-x_response_m;
x_error_s=x_response_m-x_response_s;

filter_pd_s=filter_pd_s_pre+dx_response_s*dt;
filter_pd_m=filter_pd_m_pre+dx_response_m*dt;
filter_pd_s_pre=filter_pd_s;
filter_pd_m_pre=filter_pd_m;

dx_response_s=g_pd_m*(x_response_s-filter_pd_s);        %filter
gains are equel for both master and slave PD controller
dx_response_m=g_pd_s*(x_response_m-filter_pd_m);

PD_out_m=Kp_m*x_error_m+Kd_m*(dx_response_s-dx_response_m);
PD_out_s=Kp_s*x_error_s+Kd_s*(dx_response_m-dx_response_s);

F_s= PD_out_s*Mn_s-torque_error;
F_m=PD_out_m*Mn_m-torque_error;

if (abs(RTOB_out_s) <F_defined)|(load_force_m<0)
    load_force_m=0.0;
if (x_response_s>-0.05)&(load_force_m>0)
    load_force_s=-0.05+5.5*motor_velocity_s;
elseif (-0.05-x_response_s)>=0
load_force_s=M*motor_acc_s+B*motor_velocity_s+k*(x_response_s+0.05);
elseif
(load_force_m<0&(abs(x_response_m)<=0.95*abs(x_response_s))&x_respon
se_s<=-0.05)
load_force_s=M*motor_acc_s+B*motor_velocity_s+k*(x_response_s+0.05);
elseif (x_response_s>-0.05&(load_force_m<0))
    load_force_s=0.05+10.0*motor_velocity_s;
end

torque_error = RTOB_out_m + RTOB_out_s;
I_error_s=F_s*(1/Kfn_s)+DOB_out_s*(1/Kfn_s);
end

if (abs(RTOB_out_s) >= F_defined)&(load_force_m>=0)
 FC_error= F_defined+RTOB_out_s;

 filter_FC=filter_FC_pre+dF_error*dt;
 filter_FC_r=filter_FC_r_pre+dF_error_r*dt;
 filter_FC_pre=filter_FC;
 filter_FC_r_pre=filter_FC_r;

 dF_error=g_FC*(F_defined-filter_FC);
 dF_error_r=g_FC_r*(RTOB_out_s-filter_FC_r);
```

```
 FC_out=Kp_FC*FC_error+Kd_FC*(dF_error+dF_error_r);

 torque_error = RTOB_out_m;
if (j-x_response_s)>=0
load_force_s=M*motor_acc_s+B*motor_velocity_s+k*(x_response_s+0.05);
end
 I_error_s=FC_out*(1/Kfn_s)*(-1)+DOB_out_s*(1/Kfn_s);
 F_m=k_sp*(x_response_s-x_response_m)-5.0*motor_velocity_m-
torque_error;
end

I_error_m=F_m*(1/Kfn_m)+DOB_out_m*(1/Kfn_m);

%Motors
force_error_m=I_error_m*Kfn_m-load_force_m;
motor_acc_m=force_error_m*(1/Mn_m);
force_error_s=I_error_s*Kfn_s-load_force_s;
motor_acc_s=force_error_s*(1/Mn_s);
motor_velocity_m=motor_acc_m*dt+motor_velocity_pre_m;
motor_velocity_s=motor_acc_s*dt+motor_velocity_pre_s;
x_response_m=motor_velocity_m*dt+x_response_pre_m;
x_response_s=motor_velocity_s*dt+x_response_pre_s;

%Mster DOB & RTOB
to_filter_m=I_error_m*Kfn_m+motor_velocity_m*g_dis_m*Mn_m;
filter_out_m=filter_out_pre_m+g_dis_m*(to_filter_m-filter_out_m)*dt;
filter_out_pre_m=filter_out_m;
DOB_out_m=filter_out_m-motor_velocity_m*g_dis_m*Mn_m;
RTOB_out_m=DOB_out_m-friction_m;

%Slave DOB & RTOB
to_filter_s=I_error_s*Kfn_s+motor_velocity_s*g_dis_s*Mn_s;
filter_out_s=filter_out_pre_s+g_dis_s*(to_filter_s-filter_out_s)*dt;
filter_out_pre_s=filter_out_s;
DOB_out_s=filter_out_s-motor_velocity_s*g_dis_s*Mn_s;
RTOB_out_s=DOB_out_s-friction_s;
motor_velocity_pre_m=motor_velocity_m;
motor_velocity_pre_s=motor_velocity_s;
x_response_pre_m=x_response_m;
x_response_pre_s=x_response_s;

A(i,:)=[t, x_response_m, x_response_s, RTOB_out_m, RTOB_out_s];
hold on;
figure(1);
f1=plot(A(:,1),A(:,2),A(:,1),A(:,3));
xlabel('time(s)');
ylabel('position (m)');
hold on;
figure(2);
f2=plot(A(:,1),A(:,4),A(:,1),A(:,5));
hold on;
xlabel('time(s)');
ylabel('force responce (N)');
end
```

## APPENDIX VI: C code on mbed microcontroller

```
#include "mbed.h"
#include "rtos.h"
#include "SDFileSystem.h"
#include "qeihw.h"
#include "math.h"

#define PI 3.141592653
#define Kp 900.0
#define Kv 10.0
#define Ki 10.0
#define Gd 100.0
#define F_limit 0.10

QEIHW qei_s(QEI_DIRINV_NONE, QEI_SIGNALMODE_QUAD, QEI_CAPMODE_4X,
QEI_INVINX_NONE );

// Configuring two encoder modules
void ethernet_init();
Ethernet eth;

//Variable for get angle from ethernet
char buf[0x600];
float recv_m_angle = 0.0;
float recv_s_angle = 0.0;
float inc_now = 0.0, inc_pre = 0.0;

//Safety for mbed unused pins
DigitalIn safety_19(p19);
DigitalIn safety_20(p20);
DigitalIn safety_25(p25);
DigitalIn safety_26(p26);

//Motor pwm mbed pins
PwmOut pwm_m_clk(p21);    // clockwise rotation pwm pin MASTER
PwmOut pwm_m_anticlk(p22);// anti-clockwise rotation pwm pin MASTER
PwmOut pwm_s_clk(p23);    // clockwise rotation pwm pin for SLAVE
PwmOut pwm_s_anticlk(p24);//anti-clockwise rotation pwm pin SLAVE

// Motor H bridge pins
DigitalOut  Reset_AB_M(p27);
DigitalOut  Reset_CD_M(p28);
DigitalOut  Reset_AB_S(p29);
DigitalOut  Reset_CD_S(p30);

DigitalIn M_Dir(p9);
DigitalIn S_Dir(p10);

//Current sensor inputs
AnalogIn current_sensor_m_p(p15);//current sensor input MASTER +ve
AnalogIn current_sensor_m_n(p16);// current sensor input MASTER -ve
AnalogIn current_sensor_s_p(p17);// current sensor input SLAVE +ve
AnalogIn current_sensor_s_n(p18);// current sensor input SLAVE -ve

//LED output for testing code
DigitalOut led1(LED1);
```

```
DigitalOut led3(LED3);

//Current Sensor Directions
int Master_Direction=0;
int Slave_Direction = 0;

// Encoder Constants
float const encoder_pulses_s = 2400.0;

// Motor Constant and Inertia
float const J_const_m = 0.000910;
float const J_const_s = 0.000910;
float const Kt_const_m = 0.135;
float const Kt_const_s = 0.134;
float const Kt_constinv_m = 7.407407407;
float const Kt_constinv_s = 7.462586567;

//the main function variables
Timer timer;
Timer timer1;
FILE *fp;
Ticker ticker;
int counter=0;                          //data writing loop counter
int counter_old=0;
int counter_time;
int dt_us= 150, ramp_time=0.0;          // define main loop time in us
float dt;                               //loop time in seconds for
calculations



//velocity controller variables
float x_res_m = 0.0;
float x_res_s = 0.0;
float dx_res_m = 0.0;
float dx_res_s = 0.0;
float dx_e_sum_m = 0.0;
float dx_e_sum_s = 0.0;
float const G_filcon_v_m = 2.0;//Low pass filter gain velocity
float const G_filcon_v_s = 2.0;

//current controller variables
float const G_filcon_I1_m = 100.0;
float const G_filcon_I1_s = 100.0;
float const G_filcon_I_m = 100.0;
float const G_filcon_I_s = 100.0;
float  I1_act_m=0.0;
float  I1_act_s=0.0;
float I_act_m = 0.0;
float I_act_s = 0.0;
float I_ref_m = 0.0;
float I_ref_s = 0.0;
float I_res_m = 0.0;
float I_res_s = 0.0;
float I_err_m = 0.0;
float I_err_s = 0.0;
float I_tmp_m = 0.0;
float I_tmp_s = 0.0;
float tem_I_m = 0.0;
float tem_I_s = 0.0;
```

```
float d_I_m = 0.0;
float d_I_s = 0.0;
float pwm_I_M= 0.0;
float pwm_I_S= 0.0;
float const Ikp_m = 25.0, Iki_m =1.50, Ikd_m = 0.015;
float const Ikp_s = 25.0, Iki_s = 1.50,Ikd_s = 0.015;

//DOB and RTOB variables
float tmp_m = 0.0;
float tmp_s = 0.0;
float ob_sum_m = 0.0;
float ob_sum_s = 0.0;
float ob_sum_m1 = 0.0;
float ob_sum_s1 = 0.0;
float i_com_m = 0.0;
float i_com_s = 0.0;
float fric_m = 0.0;
float fric_s = 0.0;
float i_rto_m = 0.0;
float i_rto_s = 0.0;
float torque_dob_m=0.0;
float torque_dob_s=0.0;

//controller variables
float ddx_ref_m=0.0;
float ddx_ref_s=0.0;
float I_ref_m1=0.0;
float I_ref_s1=0.0;
float tem_x_m=0.0;
float tem_x_s=0.0;
float x_err_m=0.0;
float x_err_s=0.0;
float torque_error=0.0;
float DOB_out_m=0.0;
float DOB_out_s=0.0;

//PWM generator variables
float duty_m = 0.0;                    // PWM duty for master
float duty_s = 0.0;

//Force controller variables
float const Kp_FC=650.0,Kd_FC=1.10, Ki_FC=60.0;
float G_filcon_FC=10.0;
float FC_error=0.0;
float dF_e_cmd_sum=0.0;
float dF_e_cmd=0.0;
float dF_e_res_sum=0.0;
float dF_e_res=0.0;
float FC_out=0.0;
float F_m=0.0;
float F_m1=0.0;
float F_s=0.0;
float x_eq=0.0;

//Spring controller variables
float const Kp_SC=550.0,Kd_SC=10.0,Ki_SC=5.0,k_sp= 15.0;
float G_filcon_SC=20.0;
float SC_error=0.0;
```

```
float dS_e_sum=0.0;
float dS_e=0.0;
float SC_out=0.0;
float SC_tmp=0.0;
float VC_out=0.0;
float  dx_e_VC_m=0.0;
float dx_res_VC=0.0;
float dx_VC_m=0.0;

int a=0, b=0, c=0;
float FC_S=0.0;
float FC_M=0.0;

 void pwm_init(void) {
    pwm_m_clk.period_us(10);
    pwm_m_anticlk.period_us(10);
    pwm_s_anticlk.period_us(10);
    pwm_s_clk.period_us(10);

    pwm_m_clk.write(0.0f);// Set the ouput duty-cycle, specified as
a percentage (float)
    pwm_m_anticlk.write(0.0f);
    pwm_s_anticlk.write(0.0f);
    pwm_s_clk.write(0.0f);

    Reset_AB_M = 1;        //ENABLE RUNNING MODE  (H BRIDGE ENABLE)
    Reset_CD_M = 1;
    Reset_AB_S = 1;
    Reset_CD_S = 1;
}
void velocity_m() {
        int size2 = eth.receive();
        if(size2 > 0) {
        eth.read(buf, size2);
        memcpy(&recv_m_angle, buf, sizeof(float));
        x_res_m = recv_m_angle;
        }
     dx_e_sum_m += dx_res_m*dt;
     dx_res_m =G_filcon_v_m*( x_res_m-dx_e_sum_m);
}
void velocity_s() {
    int32_t position = 0;
    qei_s.SetDigiFilter(480UL);
    qei_s.SetMaxPosition(0xFFFFFFFF);
    position = qei_s.GetPosition();
    x_res_s = -1.0*position * 2.0 * PI / encoder_pulses_s;

    dx_e_sum_s += dx_res_s*dt;
    dx_res_s = G_filcon_v_s*(x_res_s-dx_e_sum_s);
}
void current_pid(){
   Master_Direction = M_Dir.read();
   if(Master_Direction == 0){                  //master clockwise
        I_res_m = current_sensor_m_p.read();
        I1_act_m = -1.0*((I_res_m*3.3/0.74787687701613) );
      }
   else if(Master_Direction == 1) {     //master anticlockwise
        I_res_m =  current_sensor_m_n.read();
```

```
    I1_act_m = 1.0*((I_res_m*3.3)/0.713239227822580);  }

    I_act_m =I_act_m_pre + G_filcon_I1_m*(I1_act_m-I_act_m)*dt;
    I_act_m_pre=I_act_m;
    I_err_m = I_ref_m - I_act_m;
    I_tmp_m += (Iki_m * dt * I_err_m);
    tem_I_m += d_I_m*dt;
    d_I_m = G_filcon_I_m*(I_err_m - tem_I_m);
    pwm_I_M=((I_err_m * Ikp_m) + I_tmp_m + (d_I_m * Ikd_m));

    Slave_Direction = S_Dir.read();
    if(Slave_Direction == 0){     //slave clockwise
        I_res_s =  current_sensor_s_p.read();
        I1_act_s = -1.0*((I_res_s*3.3)/0.717075441532258 );
    }
   else if (Slave_Direction == 1){
        I_res_s =  current_sensor_s_n.read(); //slave anticlockwise
        I1_act_s = 1.0*((I_res_s*3.3)/0.724138445564516);}

    I_act_s =I_act_s_pre + G_filcon_I1_s*(I1_act_s-I_act_s)*dt;
    I_act_s_pre=I_act_s;
    I_err_s = I_ref_s - I_act_s;
    I_tmp_s +=  (Iki_s* dt * I_err_s);
    tem_I_s += d_I_s*dt;
    d_I_s = G_filcon_I_s*(I_err_s - tem_I_s);
    pwm_I_S=((I_err_s * Ikp_s) + I_tmp_s + (d_I_s * Ikd_s));
    }

void Disob() {
    tmp_m      = Gd*J_const_m*dx_res_m;
    ob_sum_m1  = Kt_const_m*I_act_m+tmp_m;
    ob_sum_m  += Gd*(ob_sum_m1-ob_sum_m)*dt;
    DOB_out_m = (ob_sum_m - tmp_m);
    i_com_m   = (ob_sum_m - tmp_m)*Kt_constinv_m;    //read current
    fric_m = 0.0;
    torque_dob_m= DOB_out_m-fric_m;
    i_rto_m = torque_dob_m*Kt_constinv_m;

    tmp_s      = Gd*J_const_s*dx_res_s;
    ob_sum_s1  = Kt_const_s*I_act_s+tmp_s;
    ob_sum_s  += Gd*(ob_sum_s1-ob_sum_s)*dt;
    DOB_out_s = (ob_sum_s - tmp_s);
    i_com_s   = (ob_sum_s - tmp_s)*Kt_constinv_s;    //read current
    fric_s = 0.0;//0.011;
    torque_dob_s = DOB_out_s-fric_s;
    i_rto_s = torque_dob_s*Kt_constinv_s;
    }

int Controller(void) {
if (((torque_dob_s*-1.0) <F_limit)|((x_eq-
x_res_m)<0.0&(torque_dob_m<0.0))){
    a=0;
    x_err_m=x_res_s-x_res_m;
    tem_x_m += (Ki* dt* x_err_m);
    ddx_ref_m = Kp*x_err_m +  Kv*(dx_res_s-dx_res_m)+tem_x_m;
    x_err_s=x_res_m-x_res_s;
    tem_x_s += (Ki* dt* x_err_s);
    ddx_ref_s = Kp* x_err_s +  Kv*(dx_res_m-dx_res_s)+tem_x_s;
```

```
    F_m=ddx_ref_m*J_const_m-torque_error;
    F_s=ddx_ref_s*J_const_s-torque_error;
    torque_error= torque_dob_m+torque_dob_s;
    I_ref_s1 = Kt_constinv_s*F_s;
    I_ref_s  =I_ref_s1 + DOB_out_s*Kt_constinv_s;
    I_ref_m =  Kt_constinv_m*F_m+DOB_out_m*Kt_constinv_m;
    timer1.stop();
    timer1.reset();
    FC_S=0.0;
    FC_M=0.0;
    }
if (((torque_dob_s*-1.0) >=(0.90*F_limit))&((torque_dob_s*-
1.0)<=F_limit)){//
        FC_S=I_ref_s;
        FC_M=F_m;}

if((torque_dob_s*-1.0) >= F_limit){
    a=1;
    x_eq=x_res_s;
    I_ref_s1 = Kt_constinv_s*FC_S*1.0;
    I_ref_s  =FC_S; //I_ref_s1+ DOB_out_s*Kt_constinv_s;
    timer1.start();

if(((x_eq-x_res_m)>=0.0)|(torque_dob_m>=0.0)){
    if(timer1.read()<0.500){
        if(c%100==0){
            b=!b;
            }
            c++;
        x_err_m=x_eq+0.05*b-x_res_m;
        dx_e_VC_m += dx_res_VC*dt;
        dx_res_VC =G_filcon_m*(x_err_m-dx_e_VC_m);
        tem_x_m += (Ki* dt* x_err_m);
        ddx_ref_m = Kp*x_err_m+  Kv*(dx_res_VC)+tem_x_m;

        VC_out=ddx_ref_m*J_const_m;}
    else{VC_out=0.0;}

    SC_error=k_sp*(x_eq-x_res_m)+VC_out+F_limit-torque_dob_m;
    SC_tmp += (Ki_SC * dt * SC_error);
    dS_e_sum += dS_e*dt;
    dS_e = G_filcon_SC*(SC_error-dS_e_sum);
    SC_out=(Kp_SC*SC_error+Kd_SC*(dS_e)+SC_tmp);  }

    I_ref_m =
Kt_constinv_m*J_const_m*(SC_out+VC_out)+DOB_out_m*Kt_constinv_m;
    }
    return 0;
}

void PWM_Generator() {
      duty_m =  pwm_I_M;

    if (duty_m> 0.0) {
        if (duty_m > 0.9) {
            duty_m = 0.9;
        }
```

```
        pwm_m_clk = 0.0;
        pwm_m_anticlk = duty_m;
    }

    if (duty_m < 0.0) {
        if (duty_m < -0.9) {
            duty_m = -0.9;
        }
        pwm_m_anticlk = 0.0;
        pwm_m_clk = -1.0 * duty_m;
    }

       duty_s = pwm_I_S;

    if (duty_s > 0.0) {
        if (duty_s > 0.9) {
            duty_s = 0.9;
        }
        pwm_s_clk = 0.0;
        pwm_s_anticlk = duty_s;
    }
    if (duty_s < 0.0) {
        if (duty_s < -0.9) {
            duty_s = -0.9;
        }
        pwm_s_anticlk = 0.0;
        pwm_s_clk = -1.0 * duty_s;
    }
}

void cleanup_module(void){
    pwm_m_clk = 0.0;                    // pwm cleanup module
    pwm_m_anticlk = 0.0;
    pwm_s_anticlk = 0.0;
    pwm_s_clk = 0.0;

    Reset_AB_M = 0;                 //Reset H bridge
    Reset_CD_M = 0;
    Reset_AB_S = 0;
    Reset_CD_S = 0;

    led1=0;
    led3=0;
}

//RTOS
void Control_body() {           // Control Part - main code
        velocity_m();
        velocity_s ();
        Disob();
        Controller();
        current_pid();
        PWM_Generator();
        counter++;
   }

void thread_2(void const *argument){
    led1=1;
```

```
    SDFileSystem sd(p5, p6, p7, p8, "sd");
    FILE *fp = fopen("/sd/BCG.csv", "w");

    if(fp == NULL) {
         for(int i=0;i<5;i++){
            led3=!led3;
            wait(1.0);
            }
    }

   while(counter<300000){
        if(counter>=(counter_old+100)){
        fprintf(fp, "%d %f %f %f %f
\n",timer.read_us(),torque_dob_m,torque_dob_s,x_res_m,x_res_s);
            counter_old=counter;
            led3=!led3;
            }
        }

    fclose(fp);
    timer.stop();
    cleanup_module();
    ticker.detach ();
    wait(1.0);
}
void ethernet_init(){
    eth.set_link(Ethernet::HalfDuplex100);
    wait_ms(1000);                              // Needed after startup.

    if(eth.link()) {
        for(int i=0;i<3;i++){
            led3=!led3;
            wait(1.0);
            }
    }
}

int main() {
    ethernet_init();
    pwm_init();
    timer.start();
    dt=dt_us/1000000.0;

    ticker.attach_us(&Control_body, dt_us);
    Thread
thread(*thread_2,NULL,osPriorityAboveNormal,DEFAULT_STACK_SIZE*10.0,
NULL);
}
```