

1-1-2008

# Power efficiency estimation based health monitoring and fault detection of modular and reconfigurable robot

Jing Yaun  
*Ryerson University*

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>



Part of the [Electrical and Computer Engineering Commons](#)

---

## Recommended Citation

Yaun, Jing, "Power efficiency estimation based health monitoring and fault detection of modular and reconfigurable robot" (2008).  
*Theses and dissertations*. Paper 950.

This Thesis is brought to you for free and open access by Digital Commons @ Ryerson. It has been accepted for inclusion in Theses and dissertations by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact [bcameron@ryerson.ca](mailto:bcameron@ryerson.ca).

TJ  
211-35  
Y28  
008

# **POWER EFFICIENCY ESTIMATION BASED HEALTH MONITORING AND FAULT DETECTION OF MODULAR AND RECONFIGURABLE ROBOT**

by

**JING YUAN, B. Eng.**

Toronto, Canada

September 12, 2008

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the Program of

Electrical and Computer Engineering

Toronto, Canada

© JING YUAN 2008

PROPERTY OF  
RYERSON UNIVERSITY LIBRARY

I hereby declare that I am the sole author of this thesis or dissertation.

I authorize Ryerson University to lend this thesis or dissertation to other institutions or individuals for the purpose of scholarly research.

[REDACTED]

(Signature)

I further authorize Ryerson University to reproduce this thesis or dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

[REDACTED]

# ABSTRACT

Power Efficiency Estimation Based Health Monitoring and Fault Detection of Modular and Reconfigurable Robot

© Jing Yuan, 2008

Master of Applied Science

in the Program of Electrical and Computer Engineering

Ryerson University

Power efficiency degradation of machines often provides intrinsic indication of problems associated with their operation conditions. Inspired by this observation, in this thesis work, a simple yet effective power efficiency estimation based health monitoring and fault detection technique is proposed for modular and reconfigurable robot with joint torque sensor. The design of the Ryerson modular and reconfigurable robot system is first introduced, which aims to achieve modularity and compactness of the robot modules. Critical components, such as the joint motor, motor driver, harmonic drive, sensors, and joint brake, have been selected according to the requirement. Power efficiency coefficients of each joint module are obtained using sensor measurements and used directly for health monitoring and fault detection. The proposed method has been experimentally tested on the developed modular and reconfigurable robot with joint torque sensing and a distributed control system. Experimental results have demonstrated the effectiveness of the proposed method.



## ACKNOWLEDGMENTS

I wish to express my deep gratitude to my supervisors, Dr. Guangjun Liu and Dr. Bin Wu for their support and the knowledge that they shared during my graduate studies at Ryerson University.

I am grateful to Xiaojia He, Sajan Abdul, Dr. Yugang Liu, Lan Shang and Dr. Ren Lu of the Systems and Control Lab for their beneficial discussions on my research. I am also grateful to Dr. Yongqiang Lang, Jingya Dai, Jiacheng Wang, Bing Gong and all fellow students at LEDAR for their useful discussions.

Special thanks to my uncle, Z. Liu, for his continuous support and help throughout both my research work and daily life.

I wish to share my achievements with my parents. I am very grateful for their understanding and support.

# TABLE OF CONTENTS

<b>Chapter 1 Introduction .....</b>	<b>1</b>
<b>1.1. Motivations .....</b>	<b>1</b>
<b>1.2. Background.....</b>	<b>2</b>
1.2.1 Modular and Reconfigurable Robot.....	2
1.2.2 Health Monitoring and Fault Detection .....	5
<b>1.3. Objectives and Contributions .....</b>	<b>8</b>
<b>1.4. Thesis Outline .....</b>	<b>9</b>
<b>Chapter 2 Literature Review .....</b>	<b>10</b>
<b>2.1. Health Monitoring and Fault Detection .....</b>	<b>10</b>
<b>2.2. Conclusions .....</b>	<b>15</b>
<b>Chapter 3 MRR System .....</b>	<b>16</b>
<b>3.1. Introduction .....</b>	<b>16</b>
<b>3.2. System Design of MRR .....</b>	<b>16</b>
3.2.1 Architecture for MRR System .....	16
3.2.2 Hardware Architecture of MRR Module .....	19
<b>3.3. MRR Friction Model.....</b>	<b>29</b>
<b>3.4. MRR Control System Architecture .....</b>	<b>30</b>
3.4.1 Distributed Control System.....	30
3.4.2 Fault Tolerant Control .....	33
<b>3.5. Conclusions .....</b>	<b>33</b>
<b>Chapter 4 Health Monitoring and Fault Detection Strategy .....</b>	<b>34</b>
<b>4.1. MRR System Fault Classification and Failure Modes Identification.....</b>	<b>34</b>
<b>4.2. Health Monitoring and Fault Detection Strategy.....</b>	<b>39</b>
4.2.1 Power Based Health Monitoring for MRRs.....	39
4.2.2 Power Based Fault Isolation for MRRs .....	46
<b>4.3. Conclusions .....</b>	<b>48</b>
<b>Chapter 5 Experiments .....</b>	<b>49</b>
<b>5.1. Ryerson MRR System Description .....</b>	<b>49</b>

<b>5.2. Faults Simulation.....</b>	<b>52</b>
<b>5.3. Experimental Results and Analysis .....</b>	<b>54</b>
5.3.1 Software Simulated Degradation Detection.....	54
5.3.2 Hardware Simulated Degradation Detection .....	58
5.3.3 Conclusions.....	62
<b>Chapter 6 Conclusions and Future Work .....</b>	<b>63</b>
<b>6.1. Conclusions .....</b>	<b>63</b>
<b>6.2. Major Contributions.....</b>	<b>65</b>
<b>6.3. Future Work.....</b>	<b>65</b>
<b>Appendices.....</b>	<b>67</b>
<b>References.....</b>	<b>78</b>

# LIST OF TABLES

Table 3-1: MRR2 joint module specifications .....	20
Table 3-2: Position sensor parameters .....	26
Table 5-1: Parameters of an MRR2 joint module .....	51
Table 5-2: Parameters in the friction model.....	55
Table 5-3: Power efficiency under different conditions.....	56

## LIST OF FIGURES

Figure 1-1: Generic control system architecture of MRR manipulator .....	4
Figure 1-2: Reconfigurable Robot Arm .....	5
Figure 2-1: Framework of model based fault detection process .....	12
Figure 2-2: General framework for fault diagnosis .....	14
Figure 3-1: Control system architecture of MRR manipulator .....	17
Figure 3-2: Schematic diagram of a joint module .....	18
Figure 3-3: Modular joint system .....	18
Figure 3-4: The controller module used for modular joint .....	20
Figure 3-5: Elmo driver system architecture .....	21
Figure 3-6: Driver on modular joint control system .....	21
Figure 3-7: Harmonic drive components .....	22
Figure 3-8: Flexpline with three strain gauges .....	24
Figure 3-9: Hall sensor .....	26
Figure 3-10: The reference initial position detection process .....	28
Figure 3-11: Centralized control system architecture .....	31
Figure 3-12: Distributed control system architecture .....	32
Figure 4-1: Different types of faults for inverter .....	37
Figure 4-2: Equivalent dynamic circuit for one phase of brushless DC motor .....	37
Figure 4-3: Brushless DC motor disassembled view .....	38
Figure 4-4: Robot joint in resting state .....	43
Figure 4-5: Velocity trace for MRR joint .....	45
Figure 4-6: Components for a typical modular joint .....	46
Figure 5-1: Modular and reconfigurable robot - configuration 1 .....	49
Figure 5-2: Modular and reconfigurable robot - configuration 2 .....	50
Figure 5-3: Hardware architecture of an MRR module .....	50
Figure 5-4: Joint torque and commanded current waveforms .....	56
Figure 5-5: Power efficiency estimates vary with simulated degradation .....	56
Figure 5-6: Experimental results with two joints moving together .....	58
Figure 5-7: Experimental results for power input and output .....	60

Figure 5-8: Experimental results with the additional spring load engaged and released.....	61
Figure A- 1: Torque sensor amplifier circuit connection .....	67
Figure A- 2: Encoder circuit connection .....	68
Figure A- 3: Hall sensor circuit connection .....	68
Figure A- 4: Brake circuit connection.....	69
Figure A- 5: Elmo driver circuit connection .....	69
Figure A- 6: PCB board schematic .....	70
Figure B- 1: MRR2 electrical control system test procedure .....	71

# LIST OF PRINCIPAL SYMBOLS

## Roman

$a_{ij}$	Amplitude of the frequency component of torque sensor signal
$B_i$	Viscous friction coefficient
$e_a, e_b, e_c$	Instantaneous back EMFs
$f$	Driver conversion factor
$F_{Ci}$	Coulomb friction
$f_l(q, \dot{q})$	Joint frictional torque
$F_q(q, \dot{q})$	Position dependent friction and low velocity friction model uncertainty
$F_{Si}$	Static friction
$F_{\tau i}$	Positive parameter related to the Stribeck effect
$G$	Link gravity
$h$	Torque sensor signal fluctuation
$i_a, i_b, i_c$	Instantaneous motor phase current
$I_d$	Current from power supply
$I_{mi}$	Moment of inertia of the $i^{th}$ rotor
$k_j$	Torque sensor amplifier gain
$K_T$	Motor torque constant
$m$	Torque sensor strain gauges configuration factor
$n$	Artificially introduced $K_T$ degradation factor
$P_f$	Power consumed by modeled joint friction
$P_I$	Iron loss due to hysteresis and eddy currents
$P_{in}$	System input power
$P_J$	Power loss due to the moment of inertia
$P_{out}$	Output mechanical power
$P_r$	Mechanical power input from motor rotor



$P_R$	Copper loss due to winding resistance
$P_w$	Other minor power losses on the system
$q$	Joint position
$\dot{q}$	Joint velocity
$q_{hs}$	Hall sensor signal
$q_{pi}$	Absolute position of encoder marking pulses on link side
$r$	Distance from rotor to the joint link centre
$s_f$	Signal fluctuation from strain gauge
$T_e$	Total torque excited by the back EMF
$U_d$	Power supply voltage
$u_i$	Robust compensator output
$v_c$	Command voltage from the DSP
$z_i$	Unit vector along the axis of rotation of $i^{th}$ joint
$z_{mi}$	Unit vector along the axis of rotation of the $i^{th}$ rotor

#### **Greek**

$\beta$	Rotational angle of the actuator shaft
$\gamma$	Reduction ratio of speed reducer
$\eta$	Joint power efficiency
$\eta_e$	Electric power efficiency
$\eta_m$	Mechanical transmission power efficiency
$\theta$	Joint link position and the horizontal position angle
$\lambda$	Motor total flux linkage
$\tau_i$	Overall control input
$\tau_J$	Torque sensor feedback
$\tau_r$	Rotor output torque
$\psi_{ij}$	Phase error
$w$	Joint mechanical speed



# Chapter 1 Introduction

## 1.1. Motivations

As more and more robots are introduced in space, industry, and private homes, the fault in robot system has become an important issue that needs to be properly addressed. When a robot stops performing its intended functions, either due to an internal fault or external factors, it can result in costly and even catastrophic consequences. The problem is often exacerbated if the fault is not dealt with in a timely manner.

It is, therefore, desirable to have a fault tolerant system that is capable of continued operation, possibly at degraded performance, in the event of faults in some of its parts [1]. Fault tolerance is a sought-after property for critical systems due to economic and/or safety concerns. A typical fault tolerant system should be equipped with at least three fundamental capabilities, i.e. fault detection, fault diagnosis, and fault accommodation. Fault detection is a binary decision process confirming whether or not a fault has occurred in a system. Fault diagnosis is to determine or identify the type and location of faults, which may require constant health monitoring of the system performance. Fault accommodation comprises any steps necessary to ensure continued safe operation of the system. While fault detection and diagnosis can usually be achieved by adding special purpose hardware, such as torque and current sensors, etc., at costs of increased system cost and complexity, fault accommodation may be realized by system reconfiguration, e.g., through system redundancy, soon after the fault occurs

In the mean time, the increased demand for system safety and availability, as well as low

maintenance cost, also drives the need for condition based maintenance, in which system health monitoring capability plays a crucial role in facilitating the prediction of potential system faults.

This thesis describes the author's contributions to the development of a Modular and Reconfigurable Robot (MRR) system, with an emphasis on providing it with particularly the fault detection and health monitoring capabilities, with only minimum additional hardware and computation load.

## 1.2. Background

This thesis work covers a range of work from MRR design and control, to its safe operation and health monitoring. The following sections provide background information related to the subject of this thesis study.

### 1.2.1 Modular and Reconfigurable Robot

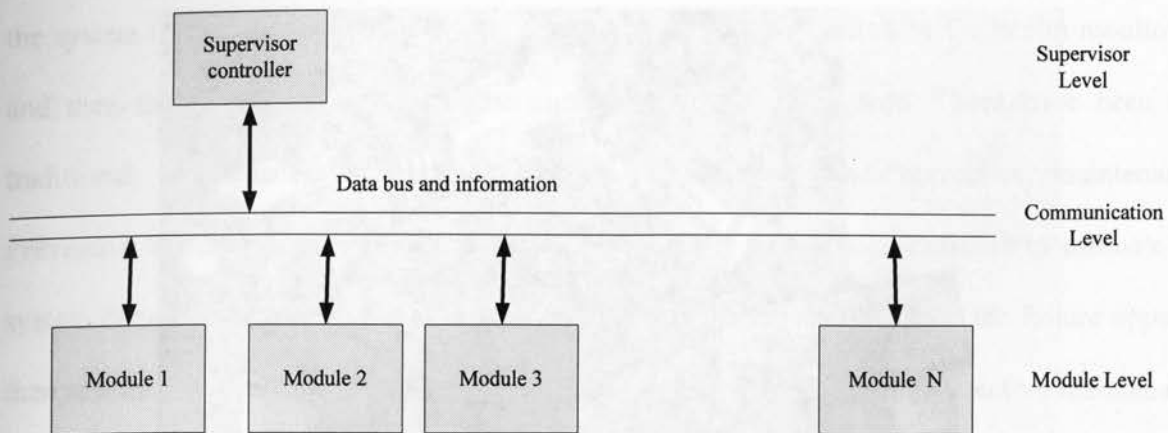
The economies and markets today are becoming increasingly unpredictable and changing radically. Many existing manufacturing systems are inadequate to meet flexible manufacturing requirement. Traditional robots are configured to perform certain tasks. The kinematics and dynamics of traditional robots are well studied. The shortcoming of traditional robots is that they can only carry out specific tasks as they are configured. If they are used to perform tasks outside their configurations, poor efficiency will be resulted in, if the tasks can be performed at all. It is therefore desirable to have a reconfigurable robot for both economic and functional purposes.

The concept of MRR manipulator was first proposed in the middle of 1980 [1, 2]. The purpose of the MRR manipulator was to adapt to different tasks, which makes the robot capable of switching from one task to another and therefore reducing the costs. One goal of the MRR design is to make the modular joint with lighter weight, less mechanical components, and higher

power efficiency.

Unlike traditional robots that are designed to perform particular tasks and have fixed configurations, MRRs are formed with multiple similar or identical modules which can be assembled in various configurations to achieve the promises of great versatility, robustness and suitability for different applications. MRRs consist of links and joints, which are traditional mechatronic components. These components can be considered as basic building blocks for the MRR. These building blocks can easily be disconnected and reconnected in different arrangements to form a new configuration.

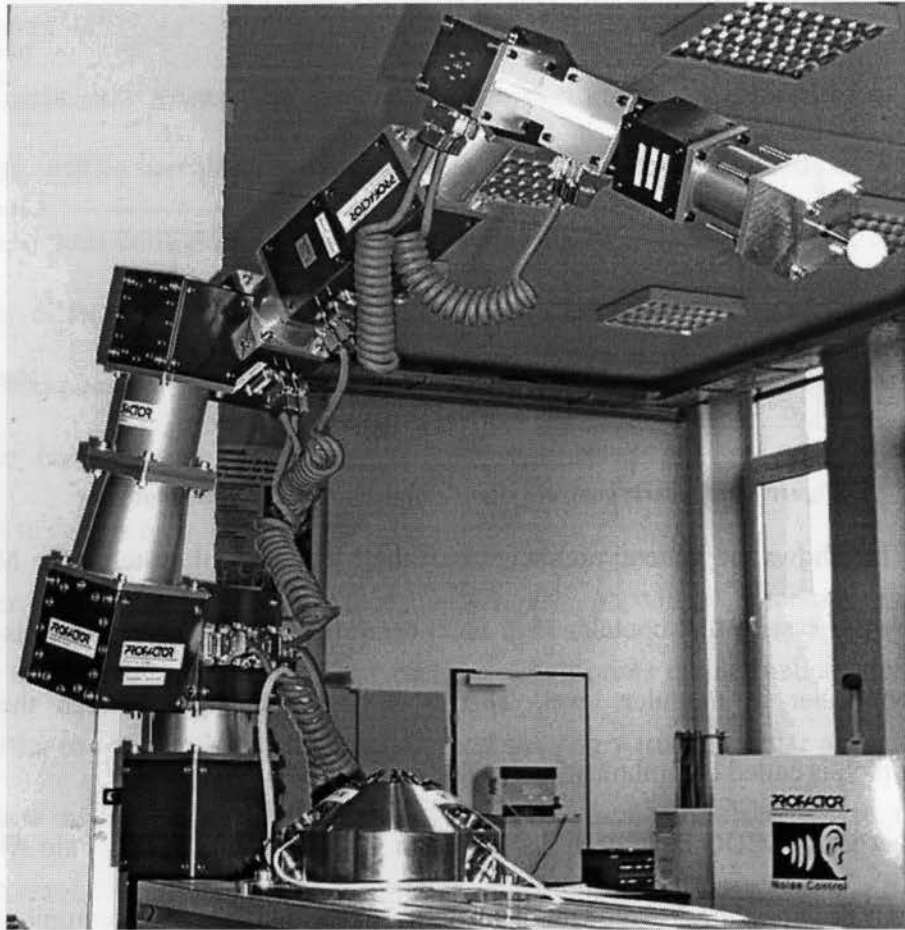
However, the reconfigurability of an MRR also introduces uncertainties in system performance in terms of kinematics and dynamics, which poses major challenges for the MRR control. In order to compensate the uncertainty and develop a truly modular and reconfigurable robot, a torque sensor will have to be integrated into each modular joint to facilitate the distributed control of the MRR [3, 4]. Using the joint torque sensor signal, friction model uncertainties can be compensated, and each MRR modular joint can be independently stabilized, when modules are added to or removed from the system. Recent research works on the MRR focus on compact modular mechanical design, mechatronics integration, robot kinematics and dynamics analysis, control scheme and optimal configuration method development, as well as the MRR fault tolerance.



**Figure 1-1: Generic control system architecture of MRR manipulator**

Figure 1-1 shows the general architecture of the MRR control system. The MRR control system is a hybrid system that contains high level centralized controller in supervisor level and distributed controller in modular level. There's a data exchange between the two level controllers, which is called communication level.

Some successful MRRs have been developed in the last decade. Xerox Palo Alto Research Center (PARC) designed the Polybot robot in California which is a robot compiled with many repeated simple modules [5], National Institute of Advanced Industrial Science and Technology (AIST) developed M-TRAN robot [6] and Engineering Services Inc.(ESI) developed an MRR in Toronto [7]. Other examples of modular robots developed over the years are: CONRO in the Information Sciences Institute in the University of Southern California [8], Telecube in PARC [9], TOMMS of Toshiba Corporation [10], Tetrobot in Rensselaer Polytechnic Institute [11] and IRIS in the Robotics and Automation Laboratory in the University of Toronto [12].



**Figure 1-2: Reconfigurable Robot Arm [13]**

Figure 1-2 shows a modular 6-DOF reconfigurable robot arm that is composed by basic reconfigurable robot manipulator blocks, which is called Intelligent Mechatronical Components [13]. This MRR consists of six modular joints (the blue cubes) called PowerCube, and is produced by Schunk Company. The joints are connected via connector elements (the golden parts)

### 1.2.2 Health Monitoring and Fault Detection

System health monitoring and fault detection are important for electrical and mechanical systems, with concerns of system safety and equipment reliability. In a complex system, the combined condition of all components shows the health state of the system. Faults of any parts of



the system that affect the proper system operation need to be detected by the health monitoring and then the right decision on maintenance operation can be made. There have been two traditional maintenance methods: preventive maintenance and corrective maintenance. Preventive maintenance is a schedule-based maintenance which uses statistics to estimate the system behaviors; corrective maintenance method lets the system run until the failure appears, then restore the machine to good health condition. The shortcoming for preventive maintenance is that the maintenance costs a lot for routine estimate, though corrective maintenance reduces the maintenance costs, but the unexpected component failures may happen during the mission and leads the mission fail, and the system down time is not expectable [14]. As the development of robot system growing fast, the traditional methods can't meet the maintenance requests, many new approaches are proposed for health management.

In order to improve health management method, the introduction of the system failures detection is necessary. The system failure warning indicates the system health condition, which can improve the electrical or mechanical system maintenance and avoid catastrophic failure [15]. The study of the system condition and the fault model are required for the warning system. Several health management technologies can meet such requirement as a sufficient maintenance action. Health management integrates controller, subsystem and components health monitoring strategies including diagnosis and prognosis technologies. Diagnostics is used to observe system performance while Prognostic and Health Management (PHM) is used to predict future state of health based on current and historical health conditions [16, 17]. The need for the health monitoring is driven by the requirement of the ability to detect the faults in electrical-mechanical system. Technical approaches for health monitoring include signal processing, experience based algorithms, model based algorithms and hybrid approaches. Health monitoring combines

sensing, recording and interpretation of environmental and operational parameters to indicate the health of a system. There are three types of parameters that can be used to indicate the health condition of a system:

1. The performance degradation, such as deviation of operation parameters from their expected values;
2. The physical or electrical degradation, such as material cracking, corrosion or electrical resistance change;
3. Life-cycle environment changes, such as signal frequency, operation duration, ambient temperature and humidity [18].

With the help of health monitoring, usual maintenance can be reduced, as well as the support cost for the system. At the same time, the reliability of the motor can be improved as the state of the actuator is monitored by the health management.

Health monitoring of a system examines its working conditions and increases safety of the system operation. For some electrical systems, it is difficult to access the components directly, and health monitoring becomes even more important [19]. Health monitoring and fault detection combine sensing, recording and interpretation of environmental and operational parameters to indicate the health of a system. The physical or electrical degradation, such as material cracking, corrosion or electrical resistance change can cause deviation of operation parameters from their expected values, and some life-cycle environment changes, such as frequency, duration, ambient temperature and humidity are also the hint of the system degradation [20].

To perform the Health Monitoring and PHM, the first step is to identify the data needed for the observation and the access to these data. The corresponding algorithms should then be developed and integrated into the prognostics and health management system [21]. These

algorithms need to show not only the evidence on the present health state of the system, but also health trends on expected future usage. Therefore, two algorithms are needed, one is for signal processing which uses the signals from sensors to observe present health of the system, the other is the trending algorithm in prognostics phase. It is based on the result of the first algorithm that the health trend can be then estimated, and the decision can be made in terms of whether the components of the system can work properly until the end of the mission [22].

### 1.3. Objectives and Contributions

The need for MRR health monitoring is driven by the requirement of the ability to detect faults in MRR systems, maintaining safety and supporting repair actions. The objective of this thesis is thus set to develop a method that facilitates the health monitoring and the fault detection for the MRR through power efficiency monitoring, as power efficiency degradation of machines is often an indication of health problems. For each module within an MRR, problems such as excessive friction, driver gating delay or broken coils will cause power efficiency degradation. Therefore, the observation of power efficiency can provide a means for health monitoring and fault detection. The torque measurement at each joint is not only used for control, but also for determining the output power. The proposed approach is straightforward and does not require the sophisticated dynamic model of the entire robot. As demonstrated with the experimental results, it can cover a wide spectrum of faults associated with the MRR modules.

The major contributions of this thesis can be categorized into two areas, the hardware development for a light and compact MRR joint module and the proposed MRR health monitoring and fault detection method. More details are listed in the following:

1. A DSP based MRR joint controller has been developed with improvement in weight and size.



2. A novel and effective power efficiency estimation based method has been proposed for health monitoring and fault detection of MRR.
3. Various fault detection simulations for MRR system degradations, including both mechanical and electrical components, have been performed.
4. Experiments have been conducted to verify the proposed power efficiency estimation based health monitoring and fault detection method.

#### 1.4. Thesis Outline

This thesis is organized to present the related subjects in six chapters.

The background of health monitoring and fault detection and recent developments in health monitoring and fault detection of electrical and mechanical devices are reviewed in Chapter 2.

In Chapter 3, detailed MRR system architecture is discussed. The system hardware design and component selection are described. The robot dynamic equations and friction model are given. The MRR control system is presented, with discussions on the distributed system control strategy and fault tolerant control concept.

Chapter 4 presents the proposed power estimation based health monitoring and fault detection method. The potential faults of the mechanical and electrical devices in the MRR system are classified. Then the health monitoring strategy is introduced. Based on this strategy, the fault detection and isolation approaches for the MRR system are discussed separately.

The experiments are presented in Chapter 5. Two types of simulated degradation are given and the power estimation based health monitoring and fault detection method is implemented and tested. The results have demonstrated the effectiveness of the proposed approach.

In the last chapter, summaries and conclusions are drawn. Possible future works are discussed.

## Chapter 2 Literature Review

### 2.1. Health Monitoring and Fault Detection

Due to the sustained needs for robotic application in remote and hazardous environments, and with emerging applications in medicine and bioengineering for the treatment of disease, robot reliability and fault tolerance have received significant research attention. Several factors must be considered for robot operation in the presence of a fault. These factors include: detection of the fault; characterization, quantification, and identification of the fault; and then response to the fault by halting the system and/or accommodating the fault (e.g., through a robust or adaptive controller or through system redundancy). Fault detection and fault diagnosis are enabling technologies to realize a fault tolerant system, and hence, has received significant interest [3].

During the last decades, there have been many published works in the subject area trying to address the fault detection and diagnosis issues in robot applications. To perform the robot actuator fault detection, approaches based on joint torque sensing have been reported [23, 24]. In [25], joint torque sensor signal is analyzed to determine the health of a robot. Most of these literatures were focused on specific components in the actuation system, such as the driver, motor, bearing and gearbox, etc. fault detection of the electrical systems, parameter estimation techniques are often used based on accessible measurements of the electrical signals such as motor voltage, current, and output angular velocity [26-28], etc. In [28], voltages at several points of the PWM motor drive system are compared, and in [27] the neural network technique is employed for diagnosis of an actuation system. Vibration analysis algorithms are often used to

detect faults in bearings, ball screws and gears [29-31]. They are based on accelerometer data, which is not normally available for electrical actuators.

There are many different types of fault that can cause system degradations, such as cracks, corrosion, and wear. In order to provide a prediction of a fault in a system, various methods have to be used to either directly or indirectly assess the stage or degree of the degradation. The direct approaches use appropriate inspection technique to detect the damage type, and the indirect approaches use the known characteristics of the undamaged system as the model, and compare the current system characteristics with that of the healthy model. Alternatively, if damages at certain stage can be accurately modeled, the degree of the degradation can be determined when the current system characteristics matches the fault model.

Some reported health monitoring schemes focus only on either the electrical or mechanical variables, and others take the overall system behavior into account. The most common methodology for the health monitoring is to check the limits of each selected variable. For example, a system has few measurable output variables  $Y(t)$ , the system health monitoring can be performed by limit-check, or called threshold-check. This method checks whether the quantities of the output  $Y(t)$  is within a tolerance zone  $Y_{\min} < Y(t) < Y_{\max}$ . If  $Y(t)$  falls outside the tolerance zone, the alarm will be set to trigger a fault diagnosis. These classical approaches do not give a deeper insight of the system and usually do not allow a fault diagnosis.

As a result, various methods for fault detection and health monitoring using mathematical models have been developed [32-34]. The task consists of the detection of faults in the processes, actuators and sensors by using the dependencies between different measurable signals. These dependencies are expressed by mathematical process models.

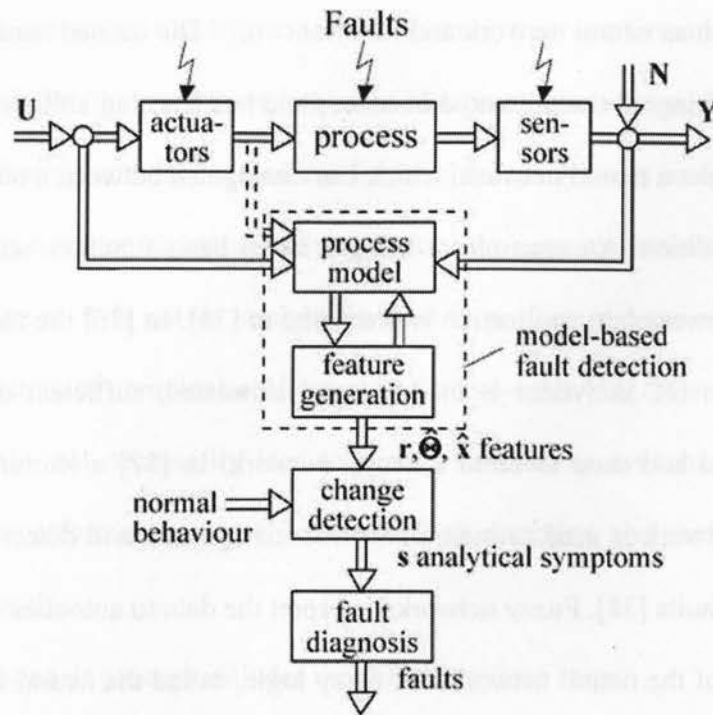


Figure 2-1: Framework of model based fault detection process [16]

Model-based fault detection approach is based on system models. Assuming the fault occurred in the system will cause certain changes in model parameters, model-based approach uses the relations between several measured variables to extract useful information that may be caused by the fault. The relations of these variables are presented in the form of the process model equations. The general process of the model based fault detection is shown in Figure 2-1, where the mathematical process model represents the relationships between the measured input signal and output signals. The parameters associated with the normal behavior of this model can be extracted which are then used to compare with those through either direct measurements or estimated values using observers. The comparison results can be analyzed and used to support the fault detection and diagnosis.

Another fault detection approach is through statistics, which is essentially a data-based method. The data based method relies on the historic data of the system. There are various data

based methods such as neural network and fuzzy network. The trained neural network is capable of correctly classifying all the presented instances and has learning ability. The sufficient data is collected to construct a neural network, which can distinguish between a no fault condition and a recorded fault condition. An example of using a radial basis function neural network for fault diagnostics in a power plant application is presented in [35]. In [36] the main power circuit of a 12 kilowatt DC to DC converter is modeled and simulated, sufficient data of the waveform pattern is collected and used to build a neural network. In [37] a learning architecture of the sigmoid neural network is used to monitor the robotic systems and detect the system abnormal performance and faults [38]. Fuzzy networks interpret the data to automate the diagnosis process. The combination of the neural network and fuzzy logic, called the neural-fuzzy network, results in a hybrid intelligent system that synergizes these two techniques by combining the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks. Since precise mathematical models for complex systems are difficult to obtain, data-based methods using artificial intelligence become popular choices for fault diagnosis.

This thesis work mainly focuses on the system health monitoring through sensors and the actuator. The health of the sensors is monitored through sensor data validation, and the health of the actuators is obtained by examining the states of the actuator control and driver systems, as well as that of the mechanical transmission.

Sensor reliability is the key for the system health monitoring. There are four typical sensor faults for analog sensors: dead, excessive noise, drift, and offset. The sensor noise is due to the system environment, which is classified as Johnson thermal noise, process noise, and sensor noise. The dead sensor and sensor noise faults can be detected through the statistical analysis of the individual sensors data. The drift and the offset faults, however, require the use of model



based fault detection methods, which are derived from system performance equations. For stochastic systems, Kalman filters are frequently used to estimate these types of fault.

The health monitoring and fault detection for mechanical systems and structures can be realized by monitoring vibration signatures of the shaft or the acoustic variation due to the wear through sensors. However, it is not so straight forward to perform the same task on electrical devices, due to their nature of complex architecture and micro-scale [16]. In an electrical system, component failures may not necessarily lead to system failures, or even have any immediate impact on the system performance, which make the fault detection and assessment of system degradation more difficult.

A typical fault diagnosis process includes the following three steps:

- 1) Fault detection: decides whether a fault exists in the system.
- 2) Fault isolation: determines the type of the fault and the location, i.e. the component, device or software module, causing the fault.
- 3) Fault identification: estimate the severity of the fault.

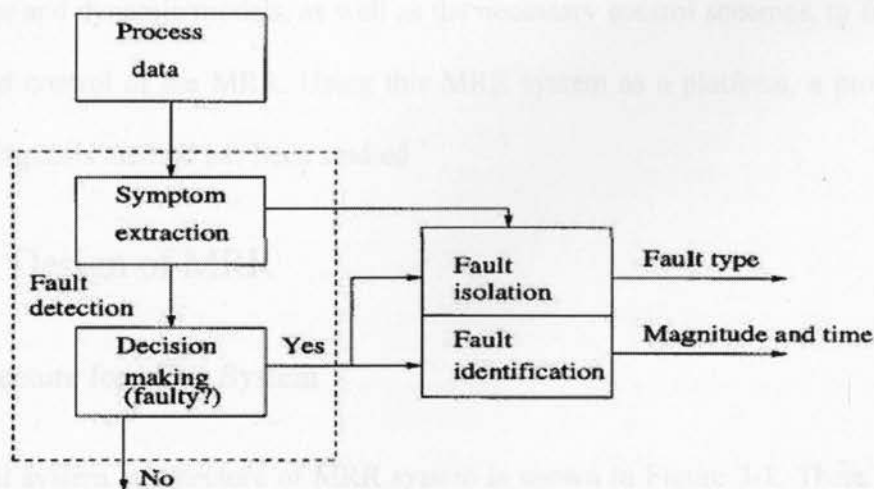


Figure 2-2: General framework for fault diagnosis

Figure 2-2 shows the framework of the system fault diagnosis. The process data is collected and symptoms of the fault are extracted through fault detections. The symptoms obtained are then used to determine the type and the location of the faults. Finally, the diagnosis information, in terms of the magnitude and the time of occurrence and duration, is generated through fault identification procedure.

## 2.2. Conclusions

The literature of health monitoring and fault detection has been reviewed in this chapter. The factors to be considered for degraded robot operation are pointed out, followed by the introduction of the common methodology for the health monitoring. The fault detection approaches are classified as model based method and data based method, and the applications for the two methods are discussed. Finally, the fault diagnosis process is described.

## Chapter 3 MRR System

### 3.1. Introduction

MRR has been studied at Ryerson University for several years. A three-module MRR manipulator is designed at Ryerson University and is under testing with a distributed torque sensing based control system. Two different joints with links, in terms of the size, MRR1 module and MRR2 module, are designed, fabricated and integrated, which can be reconfigured into different configurations. A torque sensor is incorporated within each joint. CANOpen is used as the communication medium among the modular controllers and the supervisory controller. A DSP is used as the modular joint controller for on-line control and programming, and a PC is used as the supervisory controller for off-line programming.

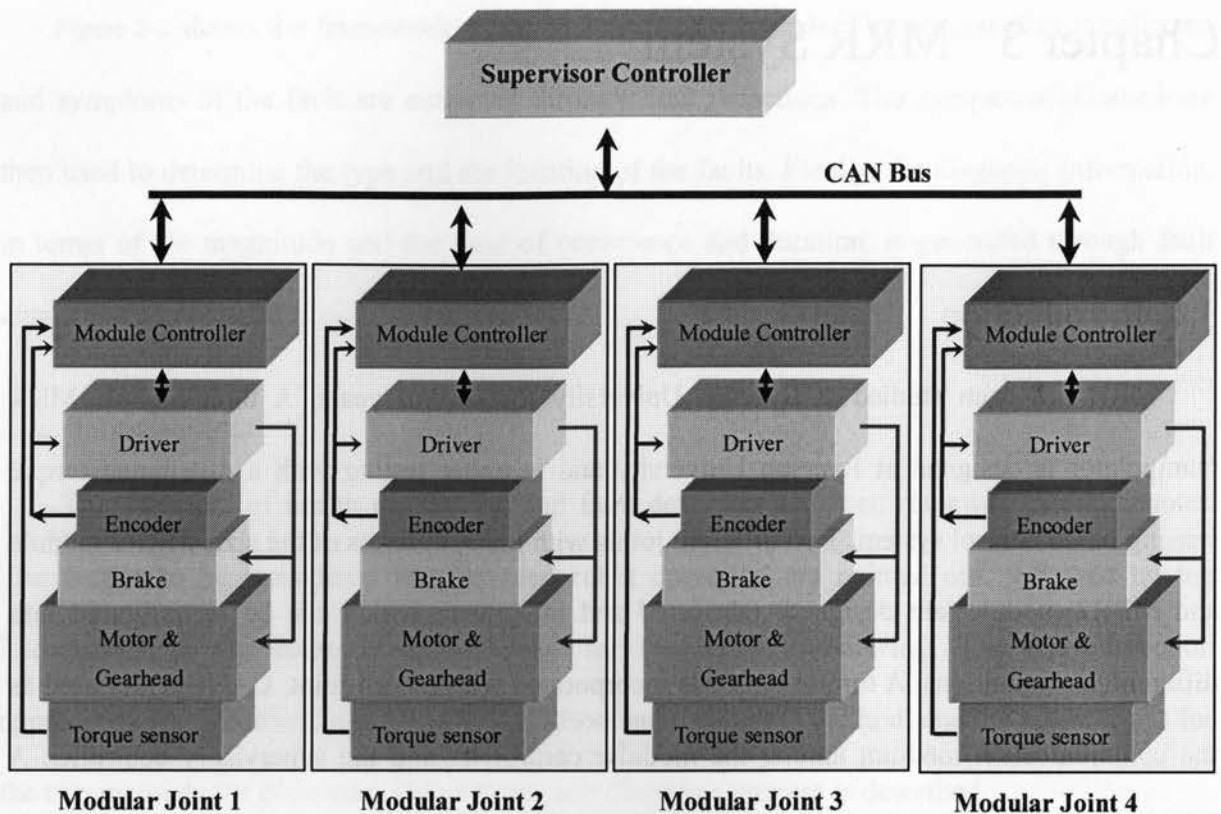
This chapter describes the hardware design of MRR and the development of the joint module structure and dynamic models, as well as the necessary control schemes, to facilitate the construction and control of the MRR. Using this MRR system as a platform, a proposed fault detection and diagnosis method has been studied.

### 3.2. System Design of MRR

#### 3.2.1 Architecture for MRR System

The control system architecture of MRR system is shown in Figure 3-1. There are several modular joints in the system under the supervision of the main controller.





**Figure 3-1: Control system architecture of MRR manipulator**

As shown in the figure above, the control system consists of a supervisor controller, a communication bus and numerous modular controllers. The supervisor controller is composed by a PC with off line task planning. The communication protocol for MRR system is the CANOpen, which is a high-level open protocol released by the CAN-in-Automation (CiA) Organization. CANOpen is applied on supervisor level, communication level and modular joints level. For each module controller, the DSP chip is used as the controller. As the torque sensor integrated in each module, torque control as well as the position control can be applied for MRR system.

A rotary MRR is composed by a number of modules. Each module has a rotary joint, a speed reducer, and a torque sensor as show in Figure 3-2.

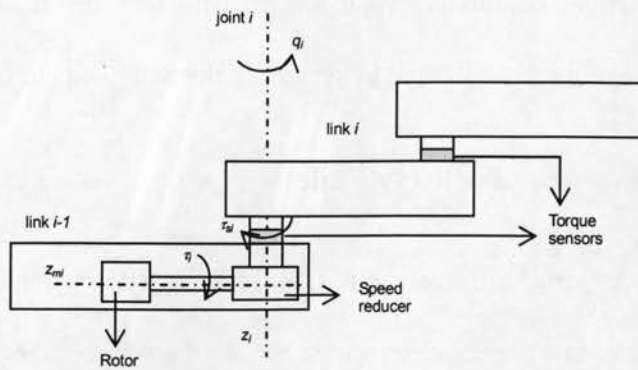


Figure 3-2: Schematic diagram of a joint module

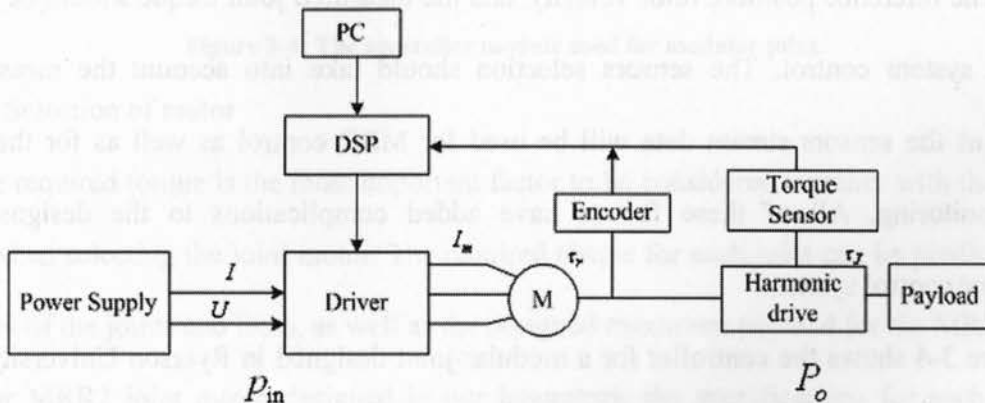


Figure 3-3: Modular joint system

Figure 3-3 illustrates the actuation system for an MRR module. The MRR joint module is actuated by a brushless DC motor. The motor rotor is connected to a harmonic drive and transmits the joint torque to the payload side. The position of the joint is measured by an encoder and the joint torque by a torque sensor. Based on the feedback of the sensor signals, the controller determines the next torque command and sends it to the driver. The DSP controller support torque control and the position control for each module.

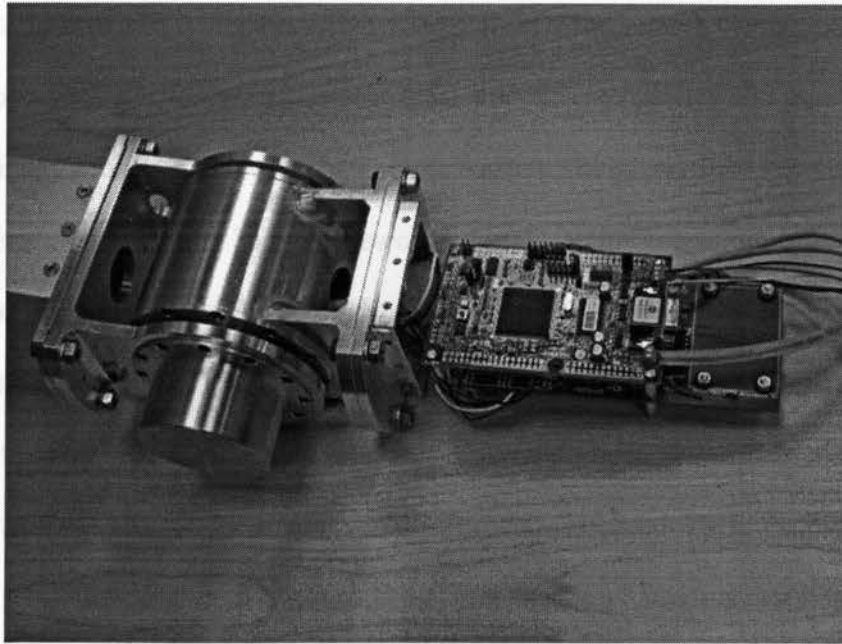
In this actuation system, potential faults on the torque sensor, driver, motor and mechanical

power transmission components need to be monitored. Overload and overheating accelerate the electrical device degradation; component wear and transmission mechanism problems increase the friction. Persistent degradation will lead to system faults and needs to be detected.

### 3.2.2 Hardware Architecture of MRR Module

Unlike the traditional robot, the configuration of the MRR is not fixed and the payload, speed, and other specifications are not determined before the whole robot is built. The sequence for MRR configuration is to build the modular joint first, then assembly several modular joints together with certain specification, after the assembling, the payload torque and the speed can be decided. The reference position, rotor velocity, and the measured joint torque should be used for the MRR system control. The sensors selection should take into account the measurement accuracy, as the sensors stream data will be used for MRR control as well as for the system health monitoring. All of these factors have added complications to the designs of the manipulator control system.

Figure 3-4 shows the controller for a modular joint designed in Ryerson University, where sensors are embedded in the joint. The controller, including the DSP, the driver and the interface circuits, are connected to the joint and are housed in the link.



**Figure 3-4: The controller module used for modular joint**

### 1. Selection of motor

The required torque is the most important factor to be considered together with the size and weight when selecting the joint motor. The required torque for each joint can be predicted based on weight of the joints and links, as well as the designed maximum payload for the MRR. For the particular MRR2 joint motor designed in our laboratory, the specifications for each joint are shown in Table 3-1.

**Table 3-1: MRR2 joint module specifications**

Item	Joint 1	Joint 2
Peak torque (Nm)	32.8	25.48
Rotation range (degree)	-120~120	-120~120

Using these specifications, the rated torque of the motor can be determined.

### 2. Selection of motor driver

The Elmo motor controller module is specially designed for controlling DC brush or brushless motors. It is selected as the joint motor driver for the MRR2, for its compact size, data processing capacity, and available interface options. Figure 3-5 shows the system architecture of the selected Elmo motor controller. Only the Power Stage of this module is used as the inverter to directly drive the joint motor. The control algorithm along with the joint module controller is implemented on a separated PCB board containing a DSP. Elmo motor controller is also mounted on this PCB board to form the modular joint controller module.

### System Architecture

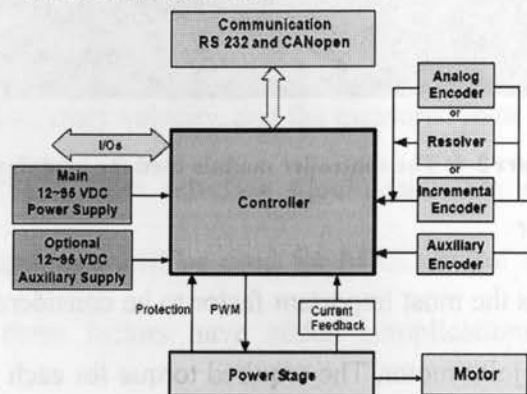


Figure 3-5: Elmo driver system architecture

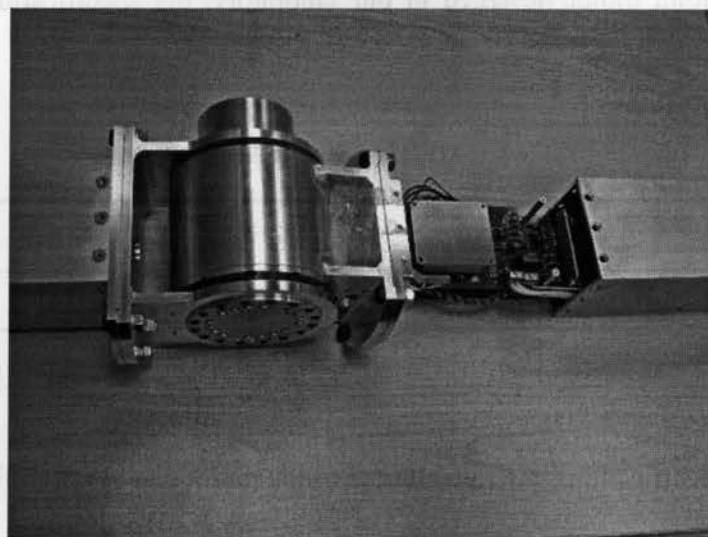


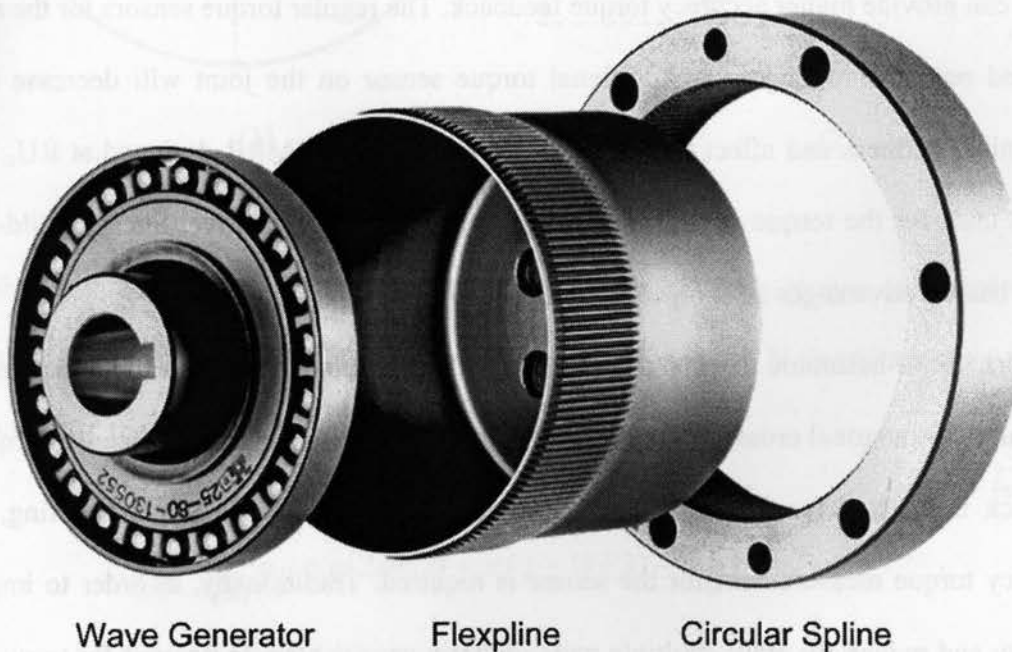
Figure 3-6: Driver on modular joint control system



The DSP in the modular joint controller module sends the torque command to the Elmo motor controller through the communication interface, which then drives the motor to perform the intended tasks. Figure 3-6 shows an Elmo motor controller mounted on the modular joint controller module.

### 3. Selection of the harmonic drive and torque sensor with amplifier

The harmonic drive is an input/output gearing mechanism which is typically used for gearing reduction. It was developed in 1955 and applied for aerospace applications primarily [39]. Compared with the gearhead, the compact design of the harmonic drive enables a high gear reduction with a small size. Figure 3-7 shows the components of a harmonic drive.



**Figure 3-7: Harmonic drive components [40]**

The harmonic drive is composed by three parts: wave generator, flexpline and circular spline. The wave generator is a bearing with an elliptical inner race and a flexible outer race. It is nested inside the flexpline, which is a flexible cup with small gear teeth on the external of the

rim. The circular spline is a rigid ring with internal gear teeth, and the diameter of the circular spline is slightly larger than the flexspline. The cup of the flexpline is thin and flexible enough to adapt the elliptical profile of the wave generator, and the external gear teeth of the flexpline can mesh the internal teeth of the circular spline. When the harmonic drive is used as the speed reduction in MRR design, the inner wave generator is mounted on the motor shaft and the circular spline is fixed. Therefore, the three components can rotate at different velocity on the same axis.

The measurement of the joint torque/force is not only for the torque control, but also for the health monitoring. The joint torque can be measured through the motor current, while the torque sensor can provide higher accuracy torque feedback. The regular torque sensors for the robots are mounted onto the robot joints. Additional torque sensor on the joint will decrease the robot mechanical stiffness and affect the robot performance. For the MRR designed at RU, the strain gauges used for the torque sensor are integrated to the harmonic drive. Such a build-in torque sensor has the advantages of compact size and good stiffness.

The use of harmonic drive and torque sensor in combination allows the joint torque to be detected by orthogonal crossing strain gauges cemented on the flexpline. Since the torque sensor feedback is the key for the MRR system distributed control and health monitoring, the high accuracy torque measurement for the sensor is required. Traditionally, in order to improve the linearity and reduce the error, multiple pairs of strain gauges have to be used for torque sensors. Each pair of stain gauges is usually placed on the opposite phase of the ellipse, to achieve the cancellation of the fluctuations in their outputs. The fluctuation signal mainly consists of a fundamental frequency component and its first harmonic [40], which has two periods and four periods per input shaft revelation separately. The traditional approaches can only cancel the

fundamental component of the fluctuation. In order to reduce the fluctuation of the signal and the number of strain gauge required, the torque sensor amplifier is introduced for the torque sensor, as shown in Fig. 3-8.

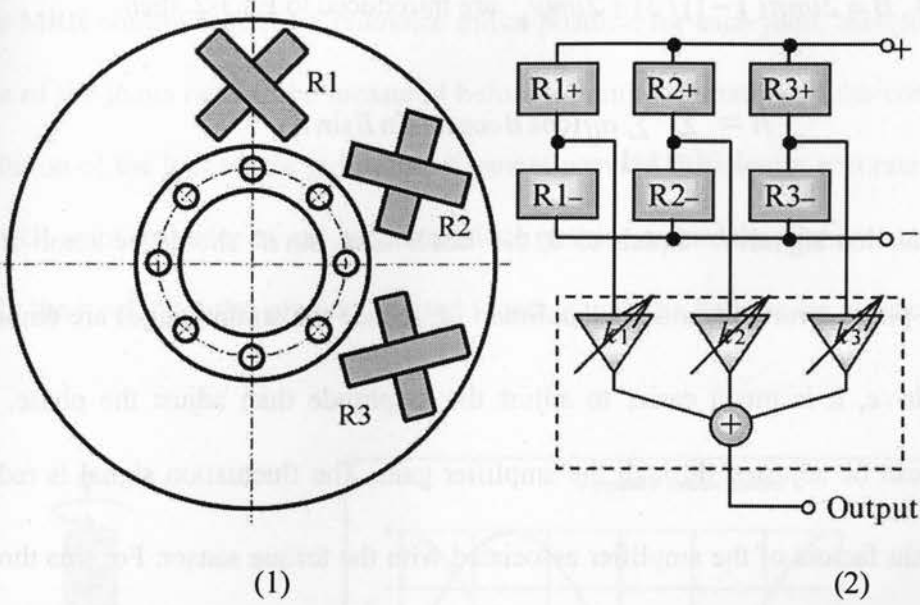


Figure 3-8: Flexpline with three strain gauges

As shown in Figure 3-8 (1), there are only three pairs of strain gauges cemented on the diaphragm of the flexpline. Assuming the fluctuation signal includes  $N$  frequency components, according to [40], the fluctuation signal for each strain gauge  $s_{ff}$  is:

$$s_{ff} = \sum_{i=1}^N a_{ij} \cos[2im(\beta - \pi(j-1)/3) - \psi_{ij}] \quad (3-1)$$

where  $a_{ij}$  is the amplitude of the frequency components,  $\beta$  is the rotational angle of the input shaft,  $\psi_{ij}$  denotes the phase errors and the  $m$  is the gear configuration factor that  $m = (\gamma - 1) / \gamma$ , where  $\gamma$  is the joint reduction ratio. Therefore the sensor output fluctuation signal  $h$  is the total fluctuation signal:



$$h = \sum_{j=1}^3 \sum_{i=1}^N a_{ij} \cos[2im(\beta - \pi(j-1)/3) - \psi_{ij}] \quad (3-2)$$

In order to reduce the total fluctuation signal, which is to make  $h$  equals to 0, the equation  $A = 2im\beta$  and  $B = 2im\pi(j-1)/3 + 2im\psi_{ij}$  are introduced to Eq.3-2, then:

$$h = \sum_{j=1}^3 \sum_{i=1}^N a_{ij} (\cos B \cos A + \sin B \sin A) \quad (3-3)$$

If the fluctuation signal  $h$  equals to 0, the  $\cos B$  and  $\sin B$  should be zero.  $B$  can be adjusted by the phase errors  $\psi_{ij}$  and the amplitude  $a_{ij}$ . Since the strain gauges are embedded in the harmonic drive, it is much easier to adjust the amplitude than adjust the phase, and the amplitude  $a_{ij}$  can be adjusted through the amplifier gain. The fluctuation signal is reduced by adjusting the gain factors of the amplifier associated with the torque sensor. For this three strain gauges torque sensor, there's separate amplifier for each pair of strain gauge as shown in Figure 3-8(2). By adjusting the three amplifier factors  $k_j$ , the fluctuation signal can be minimized.

#### 4. Selection of brake

The brake for the MRR joint is a power-off brake, which will be released when the power for the brake is on. Therefore, the power failure will cause the brake engaged and protect the manipulator from falling down. The brake is selected such that the brake force will be sufficient for the peak joint torque situation in the MRR system.

#### 5. Selection of encoder and hall sensor

The encoder and the hall sensor are both position sensors. The encoder selected for MRR is an incremental encoder, which is used to measure the motor rotated angle. The hall sensor for MRR is a GMW 360ASMF absolute position sensor, and it is used to measure the motor instantaneous position as shown in Figure 3-9. The value of the joint absolute instantaneous

position is stored in the hall sensor when the system power turned off, when the MRR is powered on again and preparing for the new movement, the hall sensor provides the previous joint position information. The parameters of the encoder and hall sensors are shown in Table 3-2.

The MRR control requires a reference initial position for each joint, therefore the absolute positions of the joints need to be measured before the implementation of the control law. Since the resolution of the hall sensor is limited, it cannot provide sufficiently accurate initial position of the MRR joint. In order to get accurate initial position value, the higher resolution encoder, especially the marking pulse pin can be used together with the hall sensor to calculate the initial position.

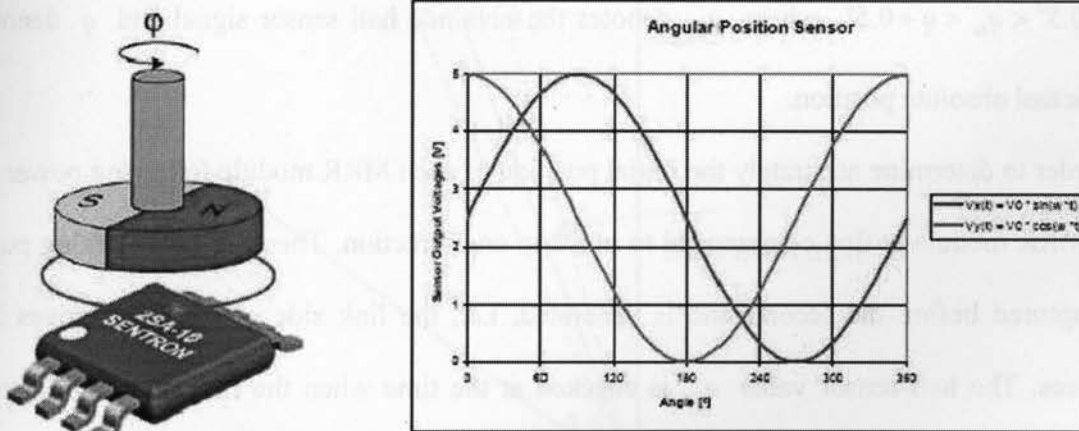


Figure 3-9: Hall sensor

Table 3-2: Position sensor parameters

	Encoder	Hall sensor
Applied range	N/A	+/-90 deg
Accuracy	1000 (PPR)	±0.5 deg

The encoder has three output signals: A+ and B+ have 90 degrees phase shift, which is used to judge direction, and the marking signal Z+ produces one pulse per rotor shaft revolution. In

order to obtain an accurate reference initial position of the joint, both the marking signal and the hall sensor signal are used:

- 1) The encoder marking signal will generate one pulse per rotor shaft revolution, which equals to 3.6 degrees for the link side of the joint as the speed reducer factor is 101:1. Then as the joint moving range on link side is -120 to +120 degrees, each marking pulse signal corresponds to a fixed marking position within this range and there is 3.6 degrees between every two adjacent each marking position. The absolute position of each marking position can be calibrated as  $q_{pi}$ .
- 2) The hall sensor can provide absolute position with an accuracy of  $\pm 0.5^\circ$  i.e.,  $q + 0.5^\circ < q_{hs} < q + 0.5^\circ$ , where  $q_{hs}$  denotes the obtained hall sensor signal and  $q$  denotes the actual absolute position.
- 3) In order to determine accurately the initial position of each MRR module following power on, the MRR module is first commanded to move in one direction. Then the first marking pulse is captured before the second one is generated, i.e., the link side of the joint moves 3.6 degrees. The hall sensor value  $q_{hs}$  is checked at the time when the first marking pulse is detected. Since the absolute difference between this hall sensor reading  $q_{hs}$  and the current link position  $q_{pi}$  is only equal to or less than  $0.5^\circ$ , which is relatively small comparing the 3.6 degrees marking position increment, the reference initial position  $q_{pi}$  can be detected by corresponding approximate  $q_{hs}$ . The process to determine the reference position is shown in Figure 3-10. Then the position measurement accuracy is equal to the resolution of the encoder. On the link side of the joint, the encoder increment is divided by the gear ratio of the harmonic drive  $\gamma$ .

- 4) Consider this first marking pulse position  $q_{pi}$  as the reference initial position. Record the incremental encoder value  $\beta$  after the  $q_{pi}$  is detected, which is known accurately by design or calibration, and the instantaneous position joint position is determined by:

$$q = q_{pi} + \frac{\beta}{\gamma} \quad (3-4)$$

Then  $q$  represents the accurate joint position measurement.

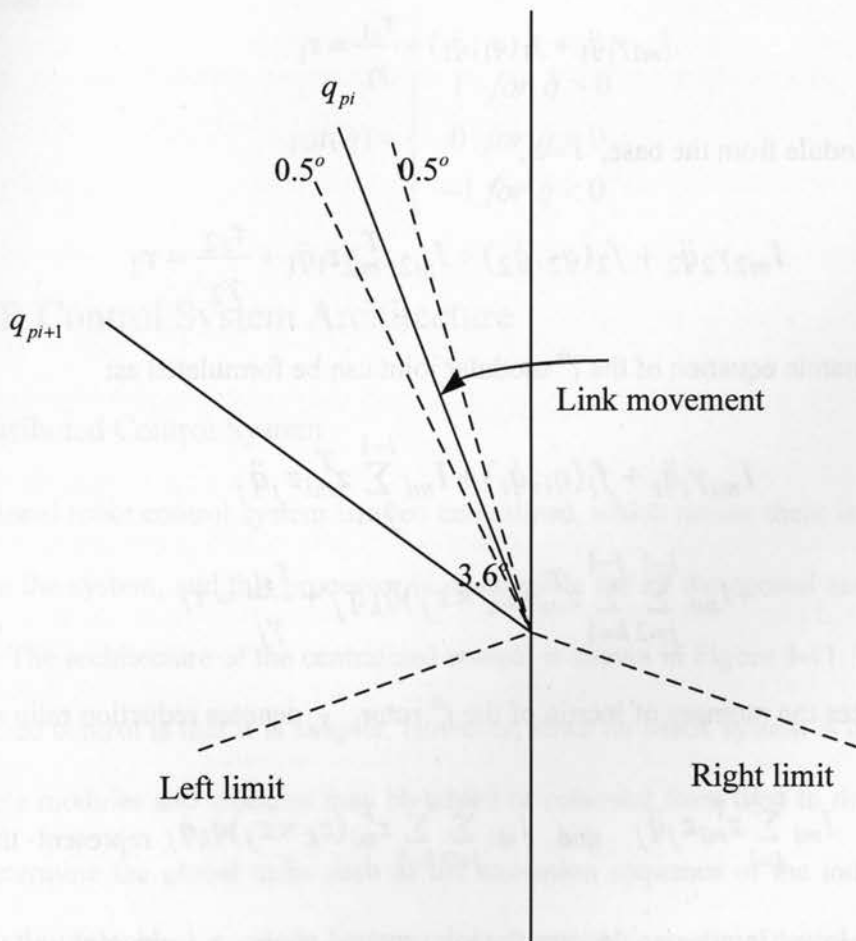


Figure 3-10: The reference initial position detection process

### 3.3. MRR Friction Model

The dynamic model of MRR joint modules can be formulated as described in [3]. There are  $n$  modular joints in the MRR system, and each joint integrates a speed reducer and a torque sensor. The modular joints are connected in serial. The base module is denoted the first joint and the top module is the  $n^{th}$  joint. Ignoring the joint flexibility issue and the transmission delay caused by speed reducer, the dynamic equation of the joint can be obtained, according to [41].

For the base module,  $i = 1$ :

$$I_{m1}\gamma_1\ddot{q}_1 + f_1(q_1, \dot{q}_1) + \frac{\tau_{s1}}{\gamma_1} = \tau_1 \quad (3-5)$$

For the second module from the base,  $i = 2$ ,

$$I_{m2}\gamma_2\ddot{q}_2 + f_2(q_2, \dot{q}_2) + I_{m2}z_{m2}^T z_1 \ddot{q}_1 + \frac{\tau_{s2}}{\gamma_2} = \tau_2 \quad (3-6)$$

For  $i \geq 3$ , the dynamic equation of the  $i^{th}$  modular joint can be formulated as:

$$\begin{aligned} I_{mi}\gamma_i\ddot{q}_i + f_i(q_i, \dot{q}_i) + I_{mi} \sum_{j=1}^{i-1} z_{mi}^T z_j \ddot{q}_j \\ + I_{mi} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} z_{mi}^T (z_k \times z_j) \dot{q}_k \dot{q}_j + \frac{\tau_{si}}{\gamma_i} = \tau_i \end{aligned} \quad (3-7)$$

where  $I_{mi}$  denotes the moment of inertia of the  $i^{th}$  rotor,  $\gamma_i$  denotes reduction ratio of the speed

reducer ( $\gamma_i \geq 1$ );  $I_{mi} \sum_{j=1}^{i-1} z_{mi}^T z_j \ddot{q}_j$  and  $I_{mi} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} z_{mi}^T (z_k \times z_j) \dot{q}_k \dot{q}_j$  represent the coupling

torque due to the lower joints,  $q_i$  denotes the joint rotated angle,  $z_{mi}$  denotes unit vector along

the axis of rotation of the  $i^{th}$  rotor, and  $z_i$  denotes the unit vector along the axis of rotation of joint

$i$ .  $f_i(q, \dot{q})$  is the joint frictional torque, which is assumed to be a function of the joint position

and velocity [1]:

$$f_l(q, \dot{q}) = B_l \dot{q}_l + F_q(q, \dot{q}) + (F_{Cl} + F_{Sl} \exp(-F_{\tau l} \dot{q}^2)) \text{sat}(\dot{q}) \quad (3-8)$$

where  $F_{Cl}$  denotes the Coulomb friction related parameter,  $B_l$  denotes the viscous friction coefficient,  $F_{Sl}$  denotes the static friction related parameter,  $F_{\tau l}$  is a positive parameter related to the Stribeck effect, and  $F_q(q, \dot{q})$  is a function of joint position and velocity, which represents the position dependency of friction and other friction modeling errors. The saturation function can be defined as:

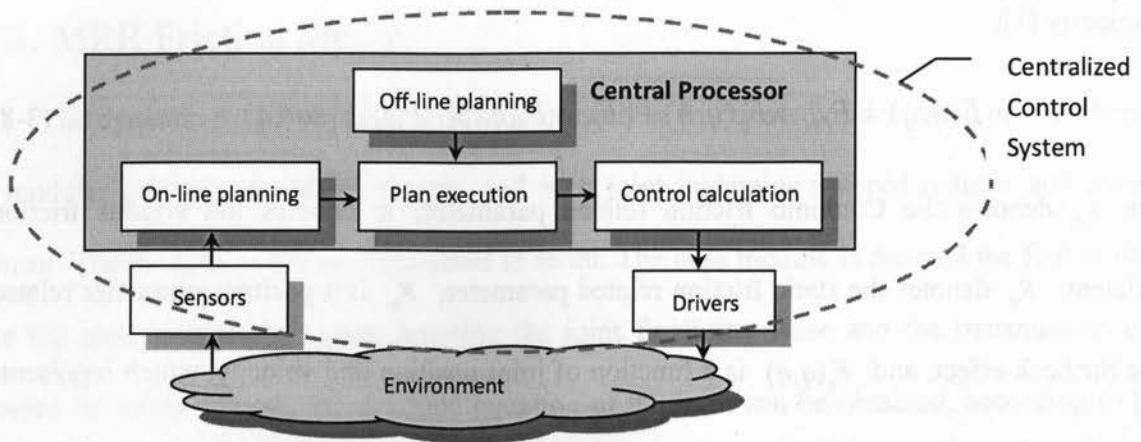
$$\text{sat}(\dot{q}) = \begin{cases} 1 & \text{for } \dot{q} > 0 \\ 0 & \text{for } \dot{q} = 0 \\ -1 & \text{for } \dot{q} < 0 \end{cases} \quad (3-9)$$

### 3.4. MRR Control System Architecture

#### 3.4.1 Distributed Control System

Traditional robot control system is often centralized, which means there is only one control processor in the system, and this processor is responsible for all the control and computation of the system. The architecture of the centralized control is shown in Figure 3-11. The advantage of the centralized control is that it is simpler. However, since an MRR system is usually composed with multiple modules and modules may be added or removed from time to time, the processor needs to determine the global tasks such as the execution sequence of the individual joints, as well as the module level controls such as motor speed control or the sensor stream data processing, etc., which makes the centralized control difficult. Therefore the distributed control for the MRR system is necessary.





**Figure 3-11: Centralized control system architecture**

A distributed control system is shown in Figure 3-12, where the control for each module is performed by the individual module level processor while the global level control tasks are processed by a supervisory level controller. The module level distributed controller is used to process the data and information from sensors, calculate control algorithms and perform control functions. The global controller is mainly for the global tasks such as robot trajectory planning and coordinating the control of the multiple joints, etc. Open hardware and software architectures along with an open communication protocol should be incorporated in the system to handle the module addition and removal. In order to achieve the distributed control and motion coordination, the communication between controllers should support the information flow among different modules and between the module controllers and the global controller.

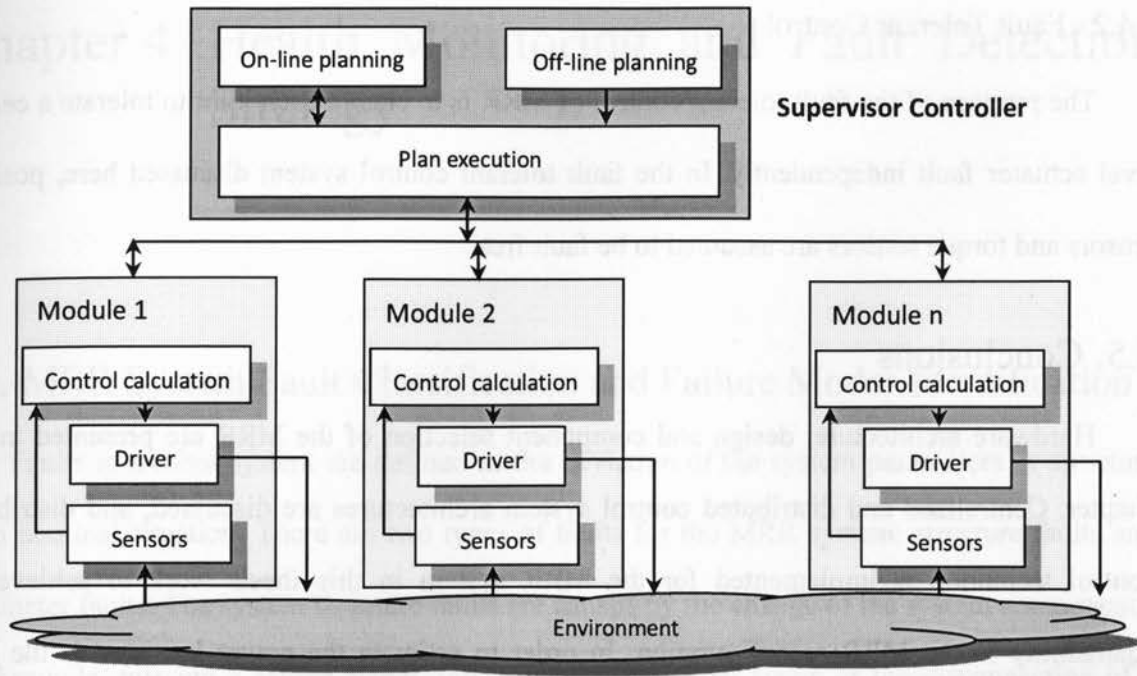


Figure 3-12: Distributed control system architecture

In order to adapt to the modular robot reconfiguration instantly, a modular distributed control technique is applied for this MRR system. The overall control law for each joints of the MRR can be defined as [3]:

$$\tau_i = \frac{\tau_{si}}{\gamma_i} + u_i \quad i = 1, 2, \dots, n \quad (3-9)$$

The control input  $u_i$  should be designed joint by joint. To compensate the model uncertainty, the saturation-based robust controller can be used. As a result, a decomposition based control design approach is developed by Liu and Goldenberg [42, 43]. This control approach distinguishes the uncertain parameters and variables of different physical types, so that effective corresponding compensators can be designed.

### 3.4.2 Fault Tolerant Control

The purpose of the fault tolerant control of MRR is to enable each joint to tolerate a certain level actuator fault independently. In the fault tolerant control system discussed here, position sensors and torque sensors are assumed to be fault free.

## 3.5. Conclusions

Hardware architecture, design and component selection of the MRR are presented in this chapter. Centralized and distributed control system architectures are discussed, and distributed control technique is implemented for the MRR system in this thesis work to achieve the adaptability to the MRR reconfiguration. In order to estimate the power loss due to the joint friction, a selected friction model for each MRR joint is also described in this section.

# Chapter 4 Health Monitoring and Fault Detection Strategy

## 4.1. MRR System Fault Classification and Failure Modes Identification

Faults in a robot system are defined as the deviation of the system parameters or structure from nominal situation. There are two types of faults for the MRR system: structure faults and parameter faults. The system structure faults are caused by the change of the system components, for example, missing a sensor signal, the damage to the joint frame or the disconnection of a component in the actuator block. Such faults cause changes in the integrated system structure. The system parameters faults are caused by mechanical component wear or electrical component damage. In a close-loop control system, all these faults will cause the system degradation and functional loss.

Faults that result in system abnormal behaviors can be distinguished by their characteristics. The steadily increasing faults could be caused by wear. Intermittent faults are often due to instantaneous electrical contact problems. Abrupt faults that lead the system break-down are usually caused by accidental incidences such as the electric cable shorting to ground or being cut off, etc.

The faults may occur in all three levels of an MRR system. The faults in the supervisory control level can usually be attributed to software problems, which can cause the system to collapse. The communication level faults will result in problems in the data stream transmission between the supervisory controller and the modular controllers. The types of fault that are of

particular interests to this thesis work are those found in the joint module, which affect the robot performance and efficiency.

The MRR system faults in modular level can be classified as follows [44]:

- 1) Controller faults: the controller faults may be caused by I/O signal reading or signal processing. It will affect the system performance.
- 2) Sensor faults: the sensor faults result in substantial errors on sensor readings.
- 3) Actuator faults: the controller is functioning properly, but the system output is not what is intended.

In each modular joint, the controller is a DSP based module with necessary I/O interface circuitry. The controller faults most often happen in the I/O interface circuitry. The DSP I/O ports are rated for 0~3V of signal range, while sensors are powered by the 5V power supply, which produce a signal range of 0-5V. Interface circuits are necessary to transform the wider range of sensor feedback into the range compactable to that of the DSP. Failures in the interface circuitry may cause damage to the DSP I/O port and the DSP processor, which will result in a total failure of the modular controller and leave the robot joint uncontrolled.

The sensors in each modular joint, including the position sensors (encoder and hall sensor) and the torque sensor, may have faults on the sensors themselves or on the sensors' signal transmission, which in combination provide incorrect sensing signals to the DSP. Torque sensor is a key component in MRR modules. Proper operation of the torque sensor is important for the distributed control of the MRR [1][4][5]. The faults found in the torque sensor are most often associated with the strain gauges and the signal transmission. Since the torque sensor used for Ryerson's MRR system is embedded in the harmonic drive. The likelihood of the strain gauge problem is low. However, the integration of the harmonic drive with torque sensor, though

improve the compactness of the system, has left very limited space for laying out the signal leads. These jammed signal leads from the harmonic drive to the coupled torque sensor amplifier may be prone to wiring problems during assembling. When the edge of the harmonic drive cuts or breaks the insulation of the leads causing shorts between sensor signal leads and the 5V power supply or ground, the torque sensor output signal will be a constant, which will result in failure in distributed torque control of the MRR system.

Comparing to controller faults and the sensors faults, actuator faults are more complicated. Failures in driver, motor, and the mechanical transmission all have the potential to affect the joint power transmission. The actuator faults include both electrical faults and mechanical faults, and these faults may combine together to make the fault distinguish and isolation even more difficult. The characteristics of the complex faults will be discussed, and the effect of these faults to the system performance and power transmission will be considered. A fault detection approach will be proposed based on analysis and understanding of these fault characteristics.

Figure 4-1 indicates the potential driver inverter faults with dashed. The faults will occur if these dashed are physically connected. The faults on the driver may occur at the switching inverter, including line to line short-circuit, single line shorted to ground at the machine terminal, and short-circuit of the switch. These driver faults may cause over current in the inverter [8].



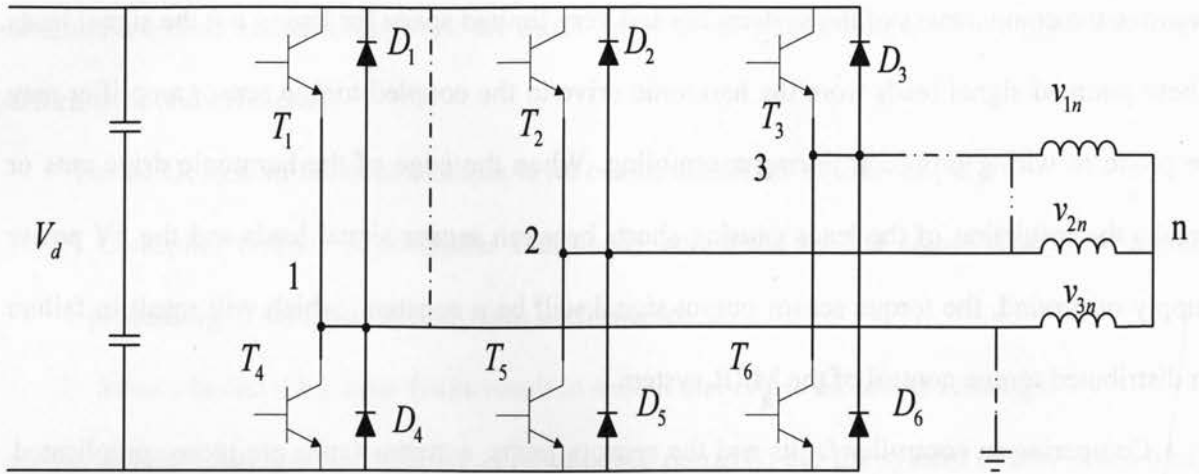


Figure 4-1: Different types of faults for inverter

A brushless DC motor is used in each MRR joint, which generates trapezoid-shaped back EMF on the winding coils. Assuming that the self and mutual inductances are constant, the equivalent dynamic circuit per phase of brushless DC motor is shown in Figure 4-2.

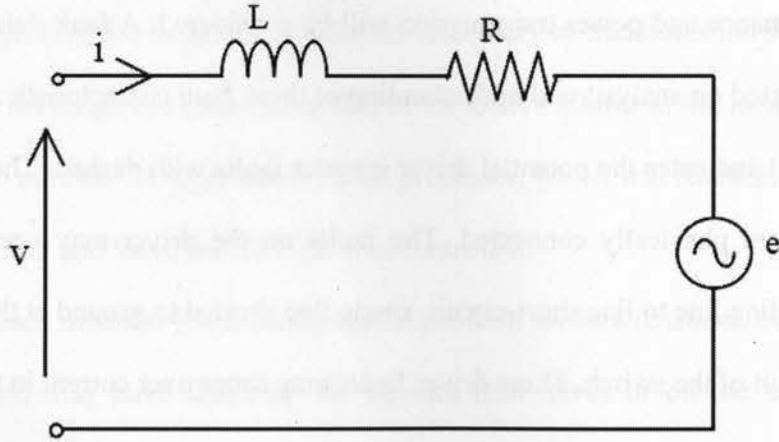


Figure 4-2: Equivalent dynamic circuit for one phase of brushless DC motor

The circuit equation can be written as:

$$v = iR + L \frac{di}{dt} + e \quad (4-1)$$

Through the driver, such motor is operated by exciting two phases of the stator winding among the three phases; only two switches are operating at the same time. The voltage equation

of the three phases is:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_a & 0 & 0 \\ 0 & R_b & 0 \\ 0 & 0 & R_c \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L_a & 0 & 0 \\ 0 & L_b & 0 \\ 0 & 0 & L_c \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \quad (4-2)$$

Therefore the torque excited by the back EMF is:

$$T_e = \frac{e_a \cdot i_a + e_b \cdot i_b + e_c \cdot i_c}{\omega} \quad (4-3)$$

where  $e_a, e_b$  and  $e_c$  are the instantaneous back EMFs,  $i_a, i_b$  and  $i_c$  are the instantaneous current and  $\omega$  denotes the motor mechanical velocity. Any short circuit failure, either in the DC link capacitor or power switch, or between inverter output phases, as shown in Figure 4-1 with dashed lines, will cause over current in the inverter. The broken winding in any one of the motor phases or single one phase shorting to the ground will lead to the decrease in output power of that phase, and therefore the excited torque  $T_e$ .

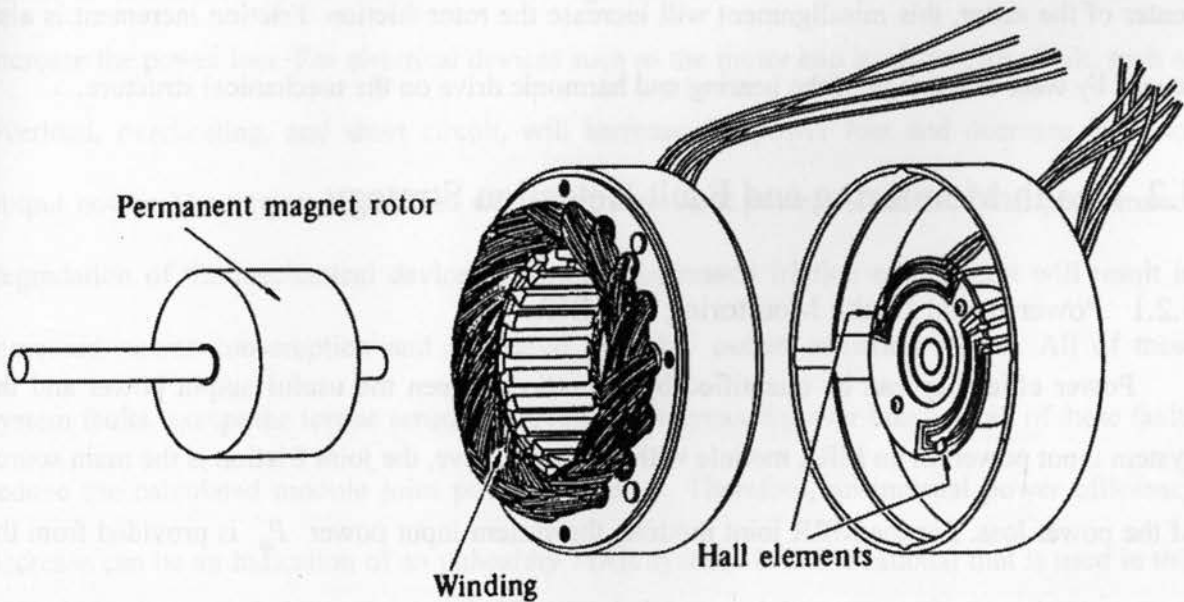


Figure 4-3: Brushless DC motor disassembled view [45]

Figure 4-3 shows the structure of a brushless DC motor. This construction is similar to the

AC motor, or known as permanent magnet synchronous motor. The rotor is composed of one or more permanent magnets and the winding are on the stator. The hall sensors are used to detect the rotor position and produce position signal for the gating control of the driver circuitry.

The faults in the brushless DC motor can be categorized as stator faults and rotor faults. Stator faults are associated with stator coil failures. Vibrations and mechanical stress during operation or assembling may scratch the coil insulation, which causes short or open circuits in phase windings. If the stator phase winding breaks, the magnetic field decreases, which causes the motor output torque to decrease or even halts the motor. Overheating will change the winding resistance value and cause more power losses on the wires, and overload can cause overheating or even short circuits. Even when the motor is working in good condition, it has over time degradation that will have some influences on the motor performance [14]-[17]. Rotor faults can be attributed to eccentricities, asymmetries, rotor unbalance, rotor magnets damages and misalignments [18]. From a dynamic point of view, when the center of the rotor is not at the center of the stator, this misalignment will increase the rotor friction. Friction increment is also caused by wear or damage of the bearing and harmonic drive on the mechanical structure.

## 4.2. Health Monitoring and Fault Detection Strategy

### 4.2.1 Power Based Health Monitoring for MRRs

Power efficiency can be quantified by the ratio between the useful output power and the system input power. In an MRR module with harmonic drive, the joint friction is the main source of the power loss. For the MRR joint module, the system input power  $P_{in}$  is provided from the power supply, power consumed by modeled joint friction is  $P_f$  and the output mechanical power is denoted by  $P_{out}$ . Ignoring other minor power losses, the joint power efficiency can then

be obtained from:

$$\eta = \frac{P_{out}}{P_{in} - P_f} \quad (4-4)$$

and

$$P_{in} = P_{out} + P_R + P_f + P_I + P_w \quad (4-5)$$

where  $P_R$  denotes the copper loss due to winding resistance,  $P_I$  denotes the iron loss due to hysteresis and eddy currents,  $P_w$  denotes other minor power losses on the system. Even in a healthy actuator system, there exists a power loss  $P_w$  due to voltage drop in the inverter power switching devices, winding resistance changes, and friction on harmonic drive. Excessive power loss may indicate system faults. Therefore the task of health monitoring of MRR system can be indirectly realized through monitoring the changes in its power efficiency, which is easily achievable for the MRR with output torque sensors incorporated.

In a distributed control system using torque sensor feedback at each MRR module, the input power is determined by the controller. The degradation of the components in the system will increase the power loss. For electrical devices such as the motor and its driver, any fault, such as overload, overheating, and short circuit, will increase the power loss and decrease the rotor output power. The torque sensor fault will affect the output power calculation. The performance degradation of the mechanical device as result of increased friction of the joint will result in increased power consumption and decreased available output actuation torque. All of these system faults, except the torque sensor fault, result in increased power loss, and all of these faults reduce the calculated module joint power efficiency. Therefore, an unusual power efficiency decrease can be an indication of an unhealthy MRR system. It is this rational that is used in this thesis work to provide the necessary health monitoring for the MRR system.

Furthermore, fault isolation (or diagnosis) may also be performed through further power

efficiency estimations on lower level components or subsystems when the overall power efficiency estimation detects an abnormal operation condition of the system.

For fault detection on torque sensors and other mechanical components in the system, the input and output power of the mechanical components should be determined. For fault detection on motors and drivers, power efficiency can be calculated (or estimated) from the power out of the power supply and the power from the motor output.

In this thesis study, a health monitoring technique based on estimated power efficiency for the MRR is proposed. In order not to complicate the issue, we assume the DSP based controller that is used to perform the health monitoring is fault free. When there is hardware performance degradation in an MRR joint, due to either the bearing friction increase, or the motor winding coil resistance increase, the power efficiency will drop comparing with that in the healthy condition. If the degradation has accumulated to a certain degree, or some abrupt faults mentioned earlier occur, the power efficiency will drop below a certain level. The level of the power efficiency may be divided into different ranges to correspond different system conditions, such as healthy, worn, and defective, etc.

For the MRR joint system, the input power and output power are:

$$\begin{aligned} P_{in} &= U_d I_d \\ P_{out} &= \tau_j \dot{q} \end{aligned} \quad (4-6)$$

where  $U_d$  and  $I_d$  denote the voltage and direct current from the power supply, respectively;  $\tau_j$  denotes joint torque from the torque sensor;  $q$  and  $\dot{q}$  are the joint angle and velocity, respectively. In an ideal motor, the winding resistance and inductance are zero, and the motor will spin at a rate that the back EMF exactly equals to the applied voltage. As a result, the overall efficiency coefficient of the joint can be defined as:

$$\eta = \frac{\tau_j \dot{q}}{U_d I_d - P_f} \quad (4-7)$$

As can be seen from Eq. 4-7, the system power efficiency coefficient is proportional to the angular velocity. When the motor angular velocity equals to zero, it makes the efficiency coefficient defined by Eq. 4-7 also equal to zero. Such sudden decrease of the efficiency coefficient indicates the fault through this fault detection approach, even though the system may be working properly. This can also happen during normal MRR motion control, when the joint velocity changes the direction.

Practically, in order to minimize the noise impact on instant power calculation, the input and output power are derived by first calculating the average energy over a period of time and then dividing it by the same time duration. The power efficiency can then be obtained by the following equation:

$$\eta = \frac{\frac{\int_{t_1}^{t_2} P_{out} dt}{\Delta t}}{\frac{\int_{t_1}^{t_2} (P_{in} - P_f) dt}{\Delta t}} = \frac{\int_{t_1}^{t_2} P_{out} dt}{\int_{t_1}^{t_2} (P_{in} - P_f) dt} = \int_{t_1}^{t_2} \frac{P_{out}}{(P_{in} - P_f)} dt \quad (4-8)$$

In the mean time, the rotor torque can be derived from the permanent magnetic flux on the stator coil current, which is proportional to the magnetic flux linkage and average coil current as shown below:

$$\tau_r = K_T i \quad (4-9)$$

where  $\tau_r$  denotes rotor output torque,  $K_T$  denotes the motor torque constant. For the trapezoidal back EMF waveform, the peak value is:

$$e = \lambda \omega \quad (4-10)$$



$\lambda$  denotes the total flux linkage of the motor, and  $w$  is the joint mechanical speed. When the robot joint is in a resting state, there is no back EMF stimulated on the winding as the motor velocity is zero in Figure 4-4.

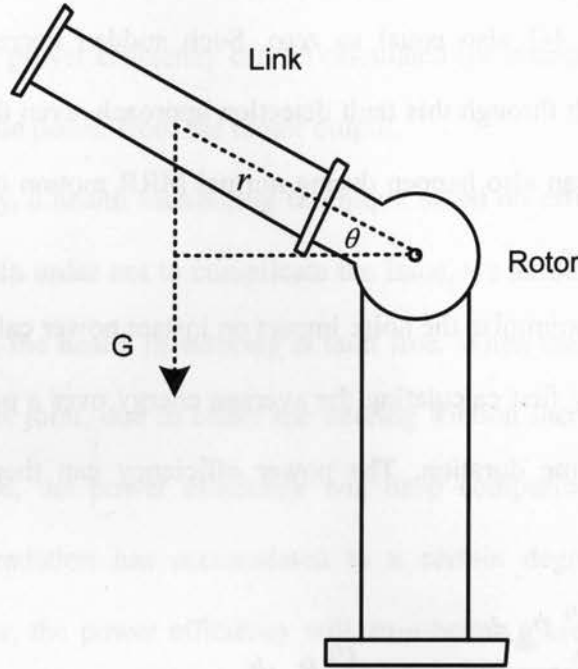


Figure 4-4: Robot joint in resting state

The torque generated by the current in the coils counteracts the gravity of link section:

$$\tau_r = K_T i = Gr \cos \theta \quad (4-11)$$

where  $G$  denotes the gravity of the link,  $r$  denotes the distance from rotor to the joint link centre and  $\theta$  denotes the angle between the joint link position and the horizontal position. If the joint link is blocked by other object or has a collision, the coil current in the motor generates the torque to counteract the link torque:

$$\tau_r = K_T i = \tau_j \quad (4-12)$$

When the angular velocity  $\dot{q}$  is zero, at the two normal conditions mentioned above, there is no mechanical output and the energy efficiency  $\eta$  will drop to zero. Correspondingly, these two conditions should be excluded when using the power based health estimation method. Other than these two conditions, the zero joint velocity may be caused by the brake fault, i.e. the excessively high brake friction. If the maximum torque generated by the motor cannot overcome the high brake friction, the joint will be “locked up”. This kind of brake fault may be caused either by improper or poor workmanship when the brake is assembled together, or due to some brake component failures that leaves the brake in a constant locking position.

From Eq. 4-12, the mechanical power input from motor rotor can be calculated as:

$$P_r = K_T i \gamma \dot{q} \quad (4-13)$$

Considering the MRR joint dynamic coupling torques and other system frictions, each modular joint can be formulated as:

$$I_{mi} \gamma \ddot{q} + f_l + I_{mi} \sum_{j=1}^{i-1} z_{mi}^T z_j \ddot{q}_j + I_{mi} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} z_{mi}^T (z_k \times z_j) \dot{q}_k \dot{q}_j + \frac{\tau_J}{\gamma} = \tau_r \quad (4-14)$$

The output power, or called estimated input power for the  $i^{th}$  joint in a period of time between  $t_1$  and  $t_2$  becomes:

$$P_{ine} = \int_{t_1}^{t_2} [\tau_r \gamma - I_i \gamma^2 \ddot{q}_i - \gamma f_l - \gamma I_{mi} \sum_{j=1}^{i-1} z_{mi}^T z_j \ddot{q}_j - \gamma I_{mi} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} z_{mi}^T (z_k \times z_j) \dot{q}_k \dot{q}_j] \dot{q}_i dt \quad (4-15)$$

Therefore, the power efficiency of the modular joint can be obtained from:

$$\eta = \frac{\int_{t_1}^{t_2} P_{out} dt}{\int_{t_1}^{t_2} P_{ine} dt} = \frac{\int_{t_1}^{t_2} P_{out} dt}{\int_{t_1}^{t_2} \tau_r \gamma \dot{q}_i dt - P_J - \int_{t_1}^{t_2} [\gamma f_l + \gamma I_{mi} \sum_{j=1}^{i-1} z_{mi}^T z_j \ddot{q}_j + \gamma I_{mi} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} z_{mi}^T (z_k \times z_j) \dot{q}_k \dot{q}_j] \dot{q}_i dt} \quad (4-16)$$

where  $P_J$  is the power loss due to the moment of inertia in this joint, which can be obtained from:

$$P_J = \int_{t_1}^{t_2} I_{mi} \gamma^2 \ddot{q}_i \dot{q}_i dt \quad (4-17)$$

In order to minimize and simplify the power efficiency calculation,  $P_J$  can be eliminated from the Eq. 4-16 by proper selection of the  $t_1$  and  $t_2$ , such that the joint velocity  $\dot{q}$  at these two particular times equals zero, as shown in Eq. 4-18:

$$P_J = \int_{t_1}^{t_2} I_{mi} \gamma^2 \ddot{q}_i \dot{q}_i dt = I_{mi} \gamma^2 \int_0^0 \dot{q}_i d\dot{q}_i = 0 \quad (4-18)$$

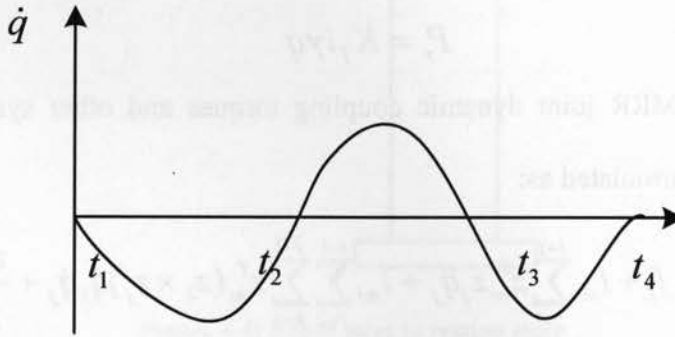


Figure 4-5: Velocity trace for MRR joint

For example, given the velocity trace shown in Figure 4-5, the estimated output power of the manipulator movement corresponding to the three time periods are:

$$\begin{aligned} P_{oute1} &= \int_{t_1}^{t_2} \left( \tau_r \gamma - \gamma f_l - \gamma I_{mi} \sum_{j=1}^{i-1} z_{mi}^T z_j \ddot{q}_j - \gamma I_{mi} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} z_{mi}^T (z_k \times z_j) \dot{q}_k \dot{q}_j \right) \dot{q} dt \\ P_{oute2} &= \int_{t_2}^{t_3} \left( \tau_r \gamma - \gamma f_l - \gamma I_{mi} \sum_{j=1}^{i-1} z_{mi}^T z_j \ddot{q}_j - \gamma I_{mi} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} z_{mi}^T (z_k \times z_j) \dot{q}_k \dot{q}_j \right) \dot{q} dt \\ P_{oute3} &= \int_{t_3}^{t_4} \left( \tau_r \gamma - \gamma f_l - \gamma I_{mi} \sum_{j=1}^{i-1} z_{mi}^T z_j \ddot{q}_j - \gamma I_{mi} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} z_{mi}^T (z_k \times z_j) \dot{q}_k \dot{q}_j \right) \dot{q} dt \end{aligned} \quad (4-19)$$

The power efficiency for the joint mechanical transmission device can be obtained:

$$\begin{aligned}
 \eta_1 &= \int_{t_1}^{t_2} \left( \frac{\tau_j}{\tau_r \gamma - \gamma f_l - \gamma I_{mi} \sum_{j=1}^{i-1} z_{mi}^T z_j \ddot{q}_j - \gamma I_{mi} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} z_{mi}^T (z_k \times z_j) \dot{q}_k \dot{q}_j} \right) dt \\
 \eta_2 &= \int_{t_2}^{t_3} \left( \frac{\tau_j}{\tau_r \gamma - \gamma f_l - \gamma I_{mi} \sum_{j=1}^{i-1} z_{mi}^T z_j \ddot{q}_j - \gamma I_{mi} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} z_{mi}^T (z_k \times z_j) \dot{q}_k \dot{q}_j} \right) dt \\
 \eta_3 &= \int_{t_3}^{t_4} \left( \frac{\tau_j}{\tau_r \gamma - \gamma f_l - \gamma I_{mi} \sum_{j=1}^{i-1} z_{mi}^T z_j \ddot{q}_j - \gamma I_{mi} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} z_{mi}^T (z_k \times z_j) \dot{q}_k \dot{q}_j} \right) dt
 \end{aligned} \tag{4-20}$$

#### 4.2.2 Power Based Fault Isolation for MRRs

As mentioned in Section 4.3.1, fault isolation (or diagnosis) may also be performed through further power efficiency estimations on lower level components or subsystems when the overall power efficiency estimation detects an abnormal operation condition of the system. For particular MRR system applications, the faults caused by electrical devices and mechanical components can then be distinguished.

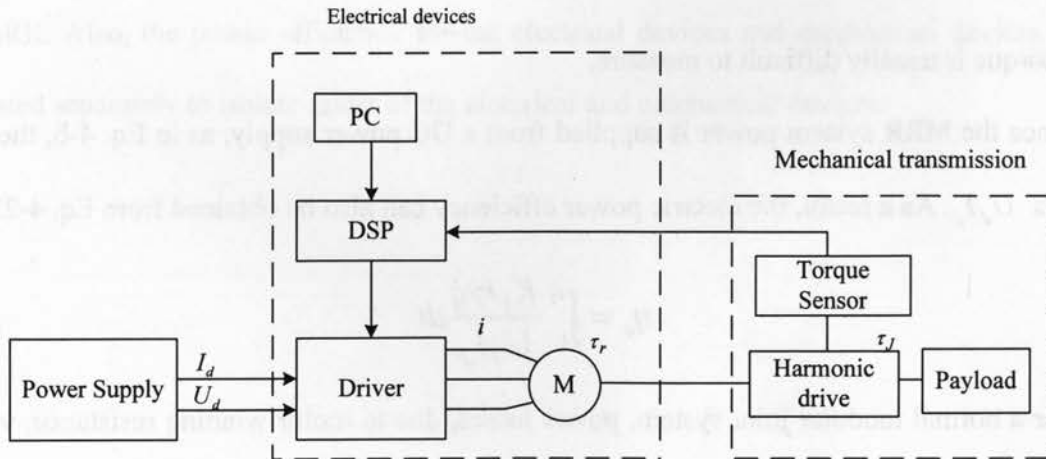


Figure 4-6: Components for a typical modular joint

As shown in Figure 4-6, the rotor torque  $\tau_r$  can be obtained through Eq. 4-12, and the output torque can be measured by the torque sensor. The power efficiency,  $\eta_m$ , for the mechanical transmission can be obtained as:

$$\eta_m = \int_{t_1}^{t_2} \frac{\tau_J \dot{q}}{(\tau_r - f_l) \gamma \dot{q}} dt = \int_{t_1}^{t_2} \frac{\tau_J}{(\tau_r - f_l) \gamma} dt \quad (4-21)$$

This mechanical transmission power efficiency  $\eta_m$  can be used to isolate the torque sensor and mechanical components faults. For example, a sudden drop in  $\eta_m$  could be an indication of either the torque sensor's output signal is somehow lost or the joint motor has failed to provide the actuation torque.

In the mean time, the electric power efficiency,  $\eta_e$ , can also be obtained as:

$$\eta_e = \int_{t_1}^{t_2} \frac{\tau_r \gamma \dot{q}}{K_T i \gamma \dot{q}} dt = \int_{t_1}^{t_2} \frac{\tau_r}{K_T i} dt \quad (4-22)$$

Eq. 4-22 represents the ratio between the motor output power and the commanded electric input by the DSP based modular joint controller. If the monitored mechanical power efficiency  $\eta_m$  appears to be normal and there is an unexpected drop in  $\eta_e$ , failures either on the motor or on the driver can be concluded. Practically, Eq. 4-12 is used to obtain  $\tau_r$ , as the actual motor rotor output torque is usually difficult to measure.

Since the MRR system power is supplied from a DC power supply, as in Eq. 4-6, the input power is  $U_d I_d$ . As a result, the electric power efficiency can also be obtained from Eq. 4-23:

$$\eta_e = \int_{t_1}^{t_2} \frac{K_T i \gamma \dot{q}}{U_d I_d} dt \quad (4-23)$$

For a normal modular joint system, power losses, due to motor winding resistance, voltage drop in power switching devices, and other normal mechanical factors, will have a normal

variation range, which also corresponds to a normal variation range in the power efficiency. Therefore there must exist certain thresholds for both the electric and mechanical power efficiencies, beyond which faults or degradations can be declared and assessed.

A decreasing trend in the power efficiency may represent a gradual degradation, while a sudden change with significant magnitude in the power efficiency may indicate an abrupt fault. On the other hand, if  $\eta_m$  decreases to a certain threshold level, the mechanical system can be declared faulty. Otherwise the fault can be narrowed down on the electrical subsystem system when  $\eta_e$  decreases to a predefined threshold, while  $\eta_m$  appears to be normal.

### 4.3. Conclusions

A proposed power efficiency estimation based health monitoring and fault detection strategy is presented in this chapter. To study the health monitoring and fault detection of the MRR, modular joint faults are classified and the effect of these faults on the joint power efficiency is studied at the beginning. Then a power efficiency estimation based health monitoring and fault detection method is introduced, which uses the sensor signals for power efficiency estimation. The health conditions of each MRR module are monitored independently of the other joints of the MRR. Also, the power efficiency for the electrical devices and mechanical devices can be estimated separately to isolate faults of the electrical and mechanical devices.



## Chapter 5 Experiments

### 5.1. Ryerson MRR System Description

Experiments on the power based health monitoring and fault detection have been performed on the 3-DOF MRR developed in our laboratory, as shown in Figure 5-1.

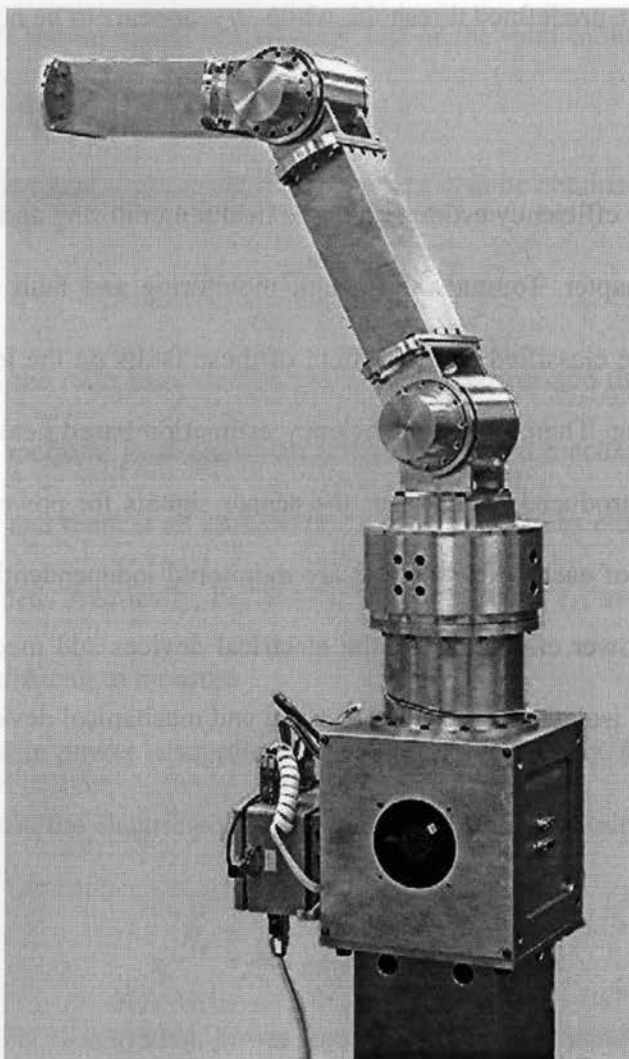
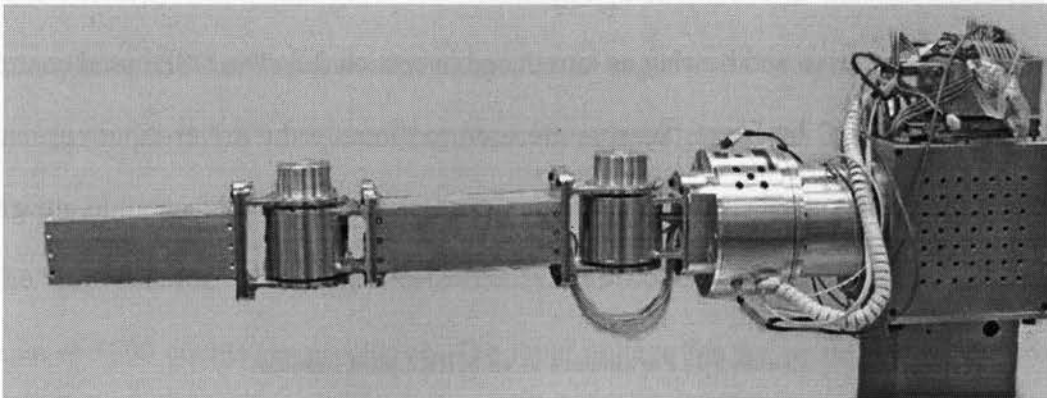
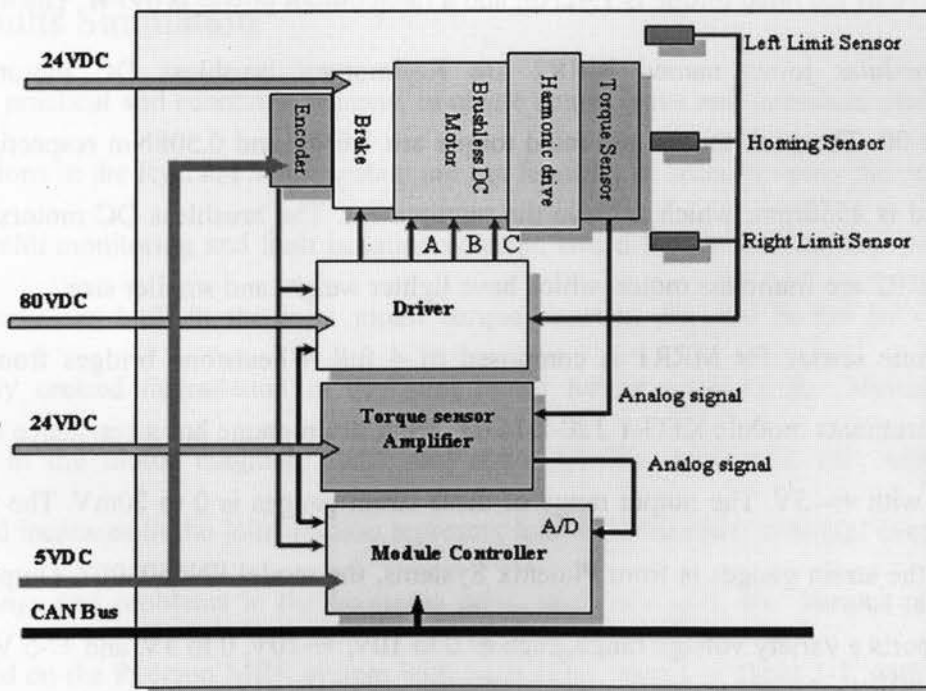


Figure 5-1: Modular and reconfigurable robot - configuration 1



**Figure 5-2: Modular and reconfigurable robot - configuration 2**

This MRR system consists of three modular joints in two different sizes. The base joint can rotate by 120 degrees and two upper joints can rotate by 240 degrees. Each joint in the MRR system can be easily added or removed, and the manipulator can be reconfigured as shown in Figure 5-1 and Figure 5-2.



**Figure 5-3: Hardware architecture of an MRR module**

The hardware architecture of MRR joint is shown in Figure 5-3, which is composed of

electrical devices including controller, driver, motor and sensors, and mechanical components such as the harmonic drive and bearing as introduced in section 3.2. The DSP based controller is programmed using the C language. Sensors are used to measure the driver input current, rotor position and the joint torque. The nominal parameters of the upper joint module are given in Table 5-1.

**Table 5-1: Parameters of an MRR2 joint module**

Gear reduction ratio	101:1
Supply voltage(V)	70
Driver conversion factor(A/V)	4
Motor torque coefficient $K_t$ (Nm/A)	0.165

The base motor used for Ryerson MRR1 system, is Moog brushless DC motor model BN-42-33EU-03. The peak torque of the motor is 8.82Nm and the rated speed is 4710rpm. The motor is selected as the rated torque is 1.42Nm and a rated output power is 697W. The motors for upper two modular joints named MRR2 are Kollmorgen brushless DC motors model RBE-01512-B-00. The peak torque and rated torque are 1.6Nm and 0.508Nm respectively, and the rated speed is 4550rpm, which achieve the requirement. The brushless DC motors for both MRR1 and MRR2 are frameless motor, which have lighter weight and smaller size.

The torque sensor for MRR1 is composed of 4 full Wheatstone bridges from Kyowa Electronic Instruments module KFG-1-120-D16-11. Each strain gauge has a resistance of 120  $\Omega$ , and is biased with +/- 5V. The output range of these strain gauges is 0 to 20mV. The amplifier coupled with the strain gauges is from Phoenix Systems, the model PN5603007. Output of this amplifier supports a variety voltage range, such as 0 to 10V, +/-10V, 0 to 5V, and +/-5 V, etc, and current loop (4 to 20 mA) operation. The harmonic drive for MRR1 is from Harmonic Drive System with model cfs-32-100-2A-GR-IV-SP-A1228, and with a ration of 101:1. MRR2 uses

Harmonic Drive System model SHG-17-100-2SO-SP (TSU), a device that combines the torque sensor, amplifier and the harmonic drive together to improve its accuracy and compactness as discussed in section 3.2. The ratio of harmonic drive is 101:1 and the amplifier output signal is ranged with  $\pm 5$  V.

The encoder for MRR1 is HS15-05/05-2000-0-03-T5-01 from Torque Sensor with resolution of 5000 counts per revolution. The input voltage for the encoder is ranged from 5 to 26V. MRR2 encoder is KS2302-50\*20-16/12 from Admotec Inc with 1000 counts per revolution. Due to the harmonic drive, the encoder increment is divided by the gear ratio and the resolution is considered  $1.01 \times 10^5$  on the link side. The input voltage for this encoder is 5V. MRR2 uses hall sensor to improve the accuracy of the position detection. A GMW model 360 ASMF4 is adopted, and its output is ranged from 0.5 to 4.5V.

## 5.2. Faults Simulation

For practical and economic reasons, to create actual faults and introduce actual component degradations in the Ryerson MRR system are not feasible. In order to verify the proposed power based health monitoring and fault isolation concept, two different methods are used to simulate the degradations both in the joint motor torque constant  $K_T$ , and in the joint friction. The artificially created degradation in the joint motor torque constant  $K_T$  represents potential changes in the motor magnetic field, and stator winding problems, etc., while artificially generated increases in the joint friction represent loss of lubrication, potential excessive wear of the bearings and problems in the harmonic drive, and rotor axis, etc. Various tests have been conducted on the Ryerson MRR system with parameters shown in Table 5-1, with the simulated faults and degradations.

Using the distributed fault tolerant control for the MRR system, the command voltage from

the DSP controller at motor driver output is:

$$v_c = \frac{i}{f} \quad (5-1)$$

where  $f$  denotes the driver conversion factor and  $i$  is the driver output current, the commanded voltage is also proportional to the rotor output torque, according to Eq. 4-12. In order to simulate an electrical degradation in the joint motor, the commanded voltage from the DSP controller,  $v_c$ , is multiplied by an artificially introduced factor  $n$  ( $0 < n < 1$ ) in the program code. Therefore the commanded rotor output power given by Eq. 4-13 becomes:

$$P_r = nK_T v_c f \gamma \dot{q} \quad (5-2)$$

The introduction of the artificial factor  $n$  in Eq. 5-2 can be equivalently taken as the motor torque constant has, somehow, been changed to  $n$  times of the original value  $K_T$ , due to whatever reasons that can cause degradation in motor magnetic field and motor winding faults. Since the motor torque constant equation is:

$$K_T = rBLN \quad (5-3)$$

the additional power loss  $\Delta P_r$ , as a result of this  $n$  time changes in the motor torque constant, can be quantified as:

$$\Delta P_r = K_T(1-n)v_c f \gamma \dot{q} \quad (5-4)$$

Keeping in mind that the fault tolerance mechanism built-in with the distributed control has the ability to “tolerate” this artificially modified controller command to the motor drive by incrementally bringing up the commanded driver output voltage to  $1/n$  of its initially commanded value based on artificially modified motor torque constant ( $nK_T$ ), at which the required (or correct) amount of electric input power to the joint motor is reached. However, these changes, both artificially and caused by the fault tolerant control algorithm, are not transparent to the



health monitoring system. The proposed power based fault isolation algorithm will still try to estimate the joint motor power efficiency using its perceived commanded torque  $\tau_r$ , defined in Eq. 4-12, as shown in Eq. 5-5 below:

$$\eta = \int_{t_1}^{t_2} \frac{\tau_J \dot{q}}{(K_T v_c f - f_l) \gamma \dot{q}} dt = \int_{t_1}^{t_2} \frac{\tau_J}{(K_T v_c f - f_l) \gamma} dt \quad (5-5)$$

This way, for each given factor  $n$ , the corresponding power efficiency can be determined. As a result, the relations between the simulated degradation in  $K_T$  (represented by the value of  $n$ ) and the corresponding changes in power efficiency can be obtained.

In order to simulate the mechanical degradation, friction like addition resistive torque on the mechanical transmission devices should be applied. Since the motor shaft and the harmonic drive are assembled in the joint, which are not easily accessible, it is difficult to add extra any resistive torque directly on the motor shaft and the harmonic drive. Consequently, a spring load attached to the motor shaft through the joint brake is used to generate the necessary friction. When the brake is released, the spring has no impact to the motor shaft. Only when the brake is engaged, the spring load allows the rotor to move in a small range with added resistive torque before the motor eventually halts. This spring load condition is used to simulate mechanical degradation, similar to the increase in the rotor shaft friction. Assuming the spring load allows the rotor to turn  $\alpha$  degrees, the joint movement trajectory for the purpose of fault isolation tests should be limited within  $\alpha$  degrees.

### 5.3. Experimental Results and Analysis

#### 5.3.1 Software Simulated Degradation Detection

The modular robot is configured as shown in Figure 5-1 and the distributed fault tolerant

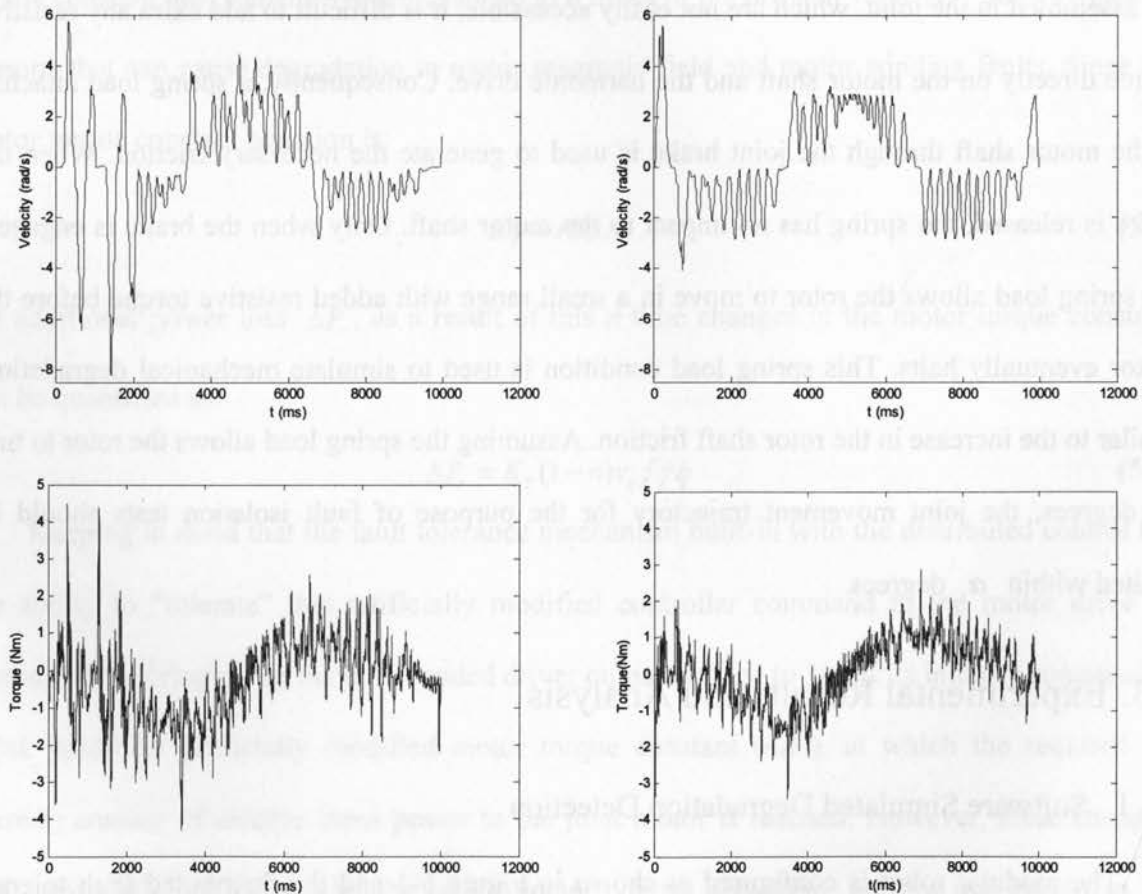


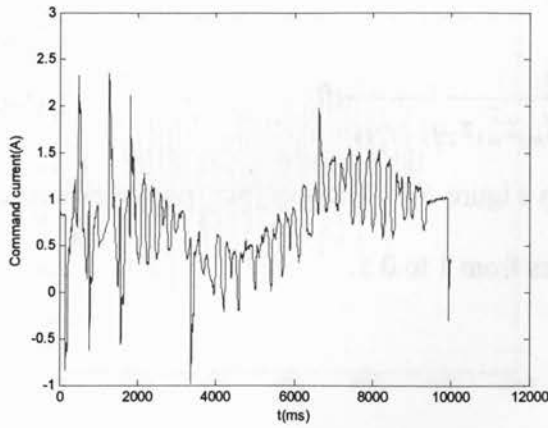
control law is applied. For estimation of the joint friction, some empiric values, listed in Table 5-2 are used.

**Table 5-2: Parameters in the friction model**

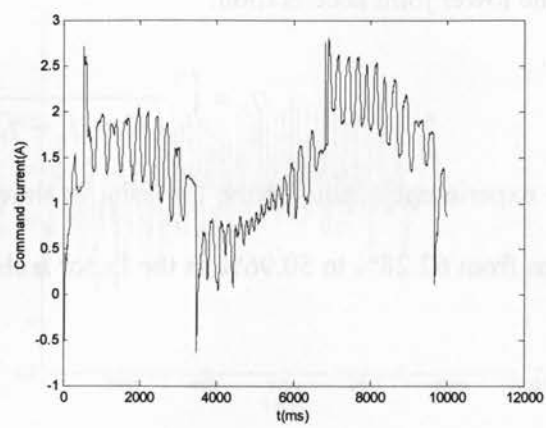
parameter	$b_i$	$f_{ci}$	$f_{si}$	$f_{\tau i}$	$I_{mi}$
Joint 2	0.00124	0.025	0.027	15	3.46e-5
Joint 3	0.00124	0.025	0.027	15	3.46e-5

The experiment is based on the top joint. The two lower joints keep rested, and the desired trajectory for the top joint is:  $q_d = 25(\sin(8t/2\pi) - 0.5\sin(16t/2\pi))$  for  $0 \leq t \leq 10s$ . Figure 5-4 shows the joint torque  $\tau_j$  and the commanded motor current  $I$ , with the introduced factor  $n$  being 1 and 0.6 in two separate experiments.





**n=1**



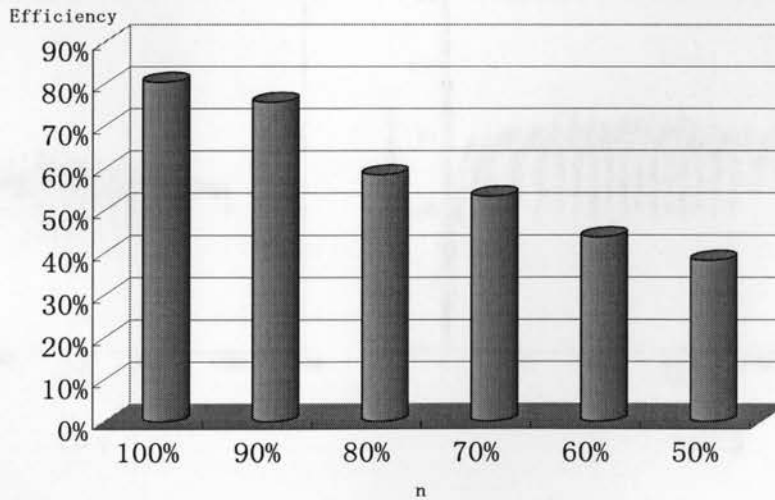
**n=0.5**

**Figure 5-4: Joint torque and commanded current waveforms**

Based on Eq. 4-21, Table 5-3 and Figure 5-5 show how the system provides desirable output while the power efficiency varies as the introduced factor  $n$  changes from 1 to 0.5.

**Table 5-3: Power efficiency under different conditions**

$n$	1	0.9	0.8	0.7	0.6	0.5
Efficiency (%)	80.5	75.7	54.49	53.4	43.74	38.19



**Figure 5-5: Power efficiency estimates vary with simulated degradation**

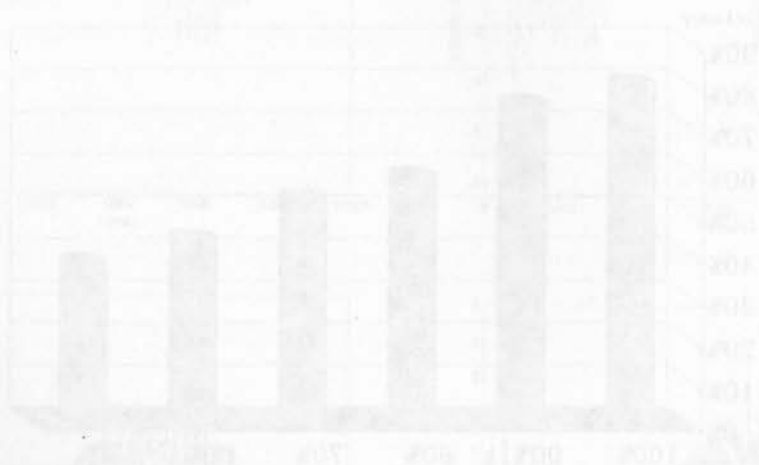
Then let the top and the second joint move together with desired trajectory  $q_d = 25(1 - \cos(2t - 1)\pi/40)$  for  $0 \leq t \leq 10s$ . The dynamic equation of the third joint is affected

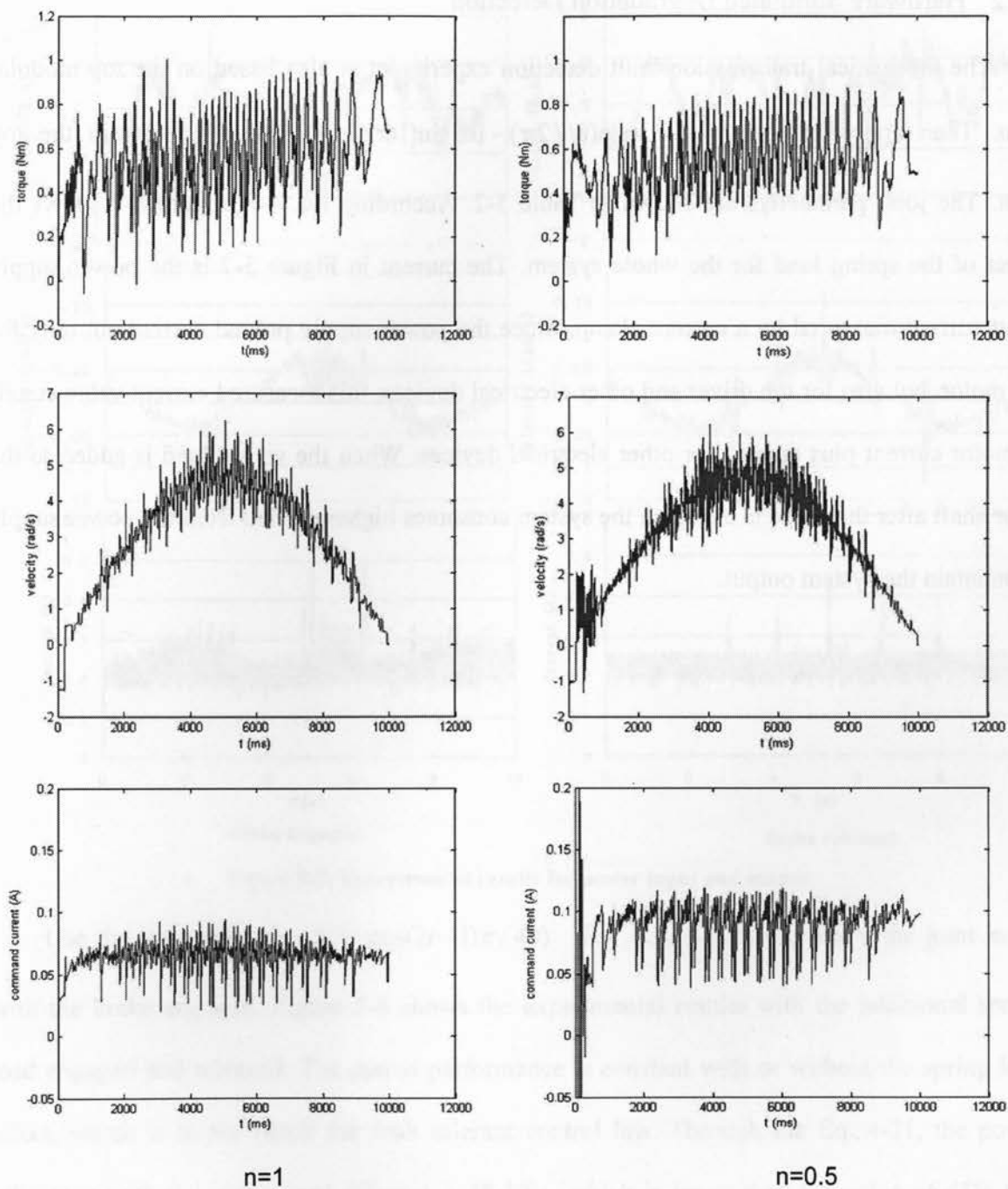
by the lower joint acceleration:

$$\eta_3 = \int_{t_1}^{t_2} \frac{\tau_3 \dot{q}_3}{(K_T v_c f_r - f_{l3} - I_{m3} z_{m3}^T z_2 \ddot{q}_2) \gamma \dot{q}_3} dt \quad (5-6)$$

The experiment result for the top joint is shown in Figure 5-6. The top joint power efficiency drops from 62.28% to 50.96% as the factor  $n$  changes from 1 to 0.5.

$n$	Efficiency (%)
1.0	62.28
0.9	58.75
0.8	54.69
0.7	51.4
0.6	47.74
0.5	50.96





**Figure 5-6: Experimental results with two joints moving together**

As two joints moving together, the coupling effect from lower joint has influence on the top joint, and makes the measured torque value of the top joint remain positive, as shown in Figure 5-6.

### 5.3.2 Hardware Simulated Degradation Detection

The mechanical transmission fault detection experiment is also based on the top modular joint. Then apply trajectory  $q_d = 3(\sin(8t/2\pi) - 0.5\sin(16t/2\pi))$  and  $0 \leq t \leq 10s$  to the top joint. The joint parameters are shown in Table 5-2. According Eq. 4-7, Figure 5-7 shows the effect of the spring load for the whole system. The current in Figure 5-7 is the power supply input current measured by a current clamp. Since the power supply provided current not only for the motor, but also for the driver and other electrical devices, this measured current value equals to motor current plus current for other electrical devices. When the spring load is added to the rotor shaft after the brake is engaged, the system consumes higher current from the power supply to maintain the system output.

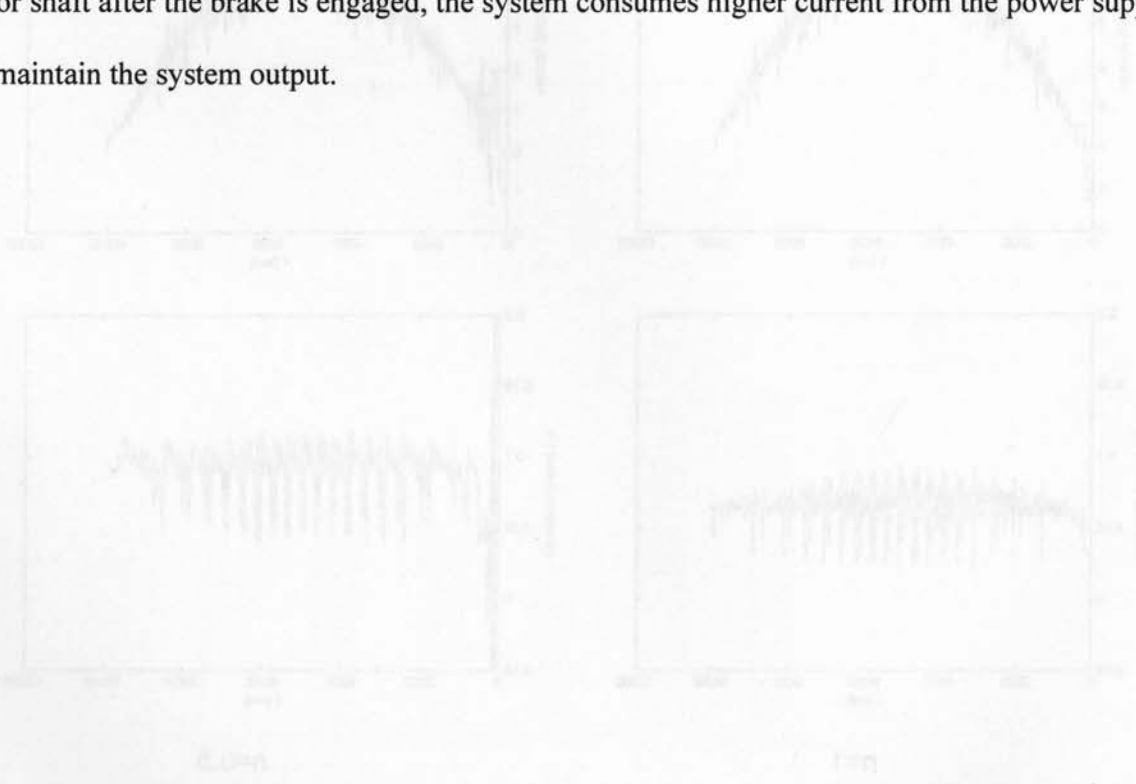
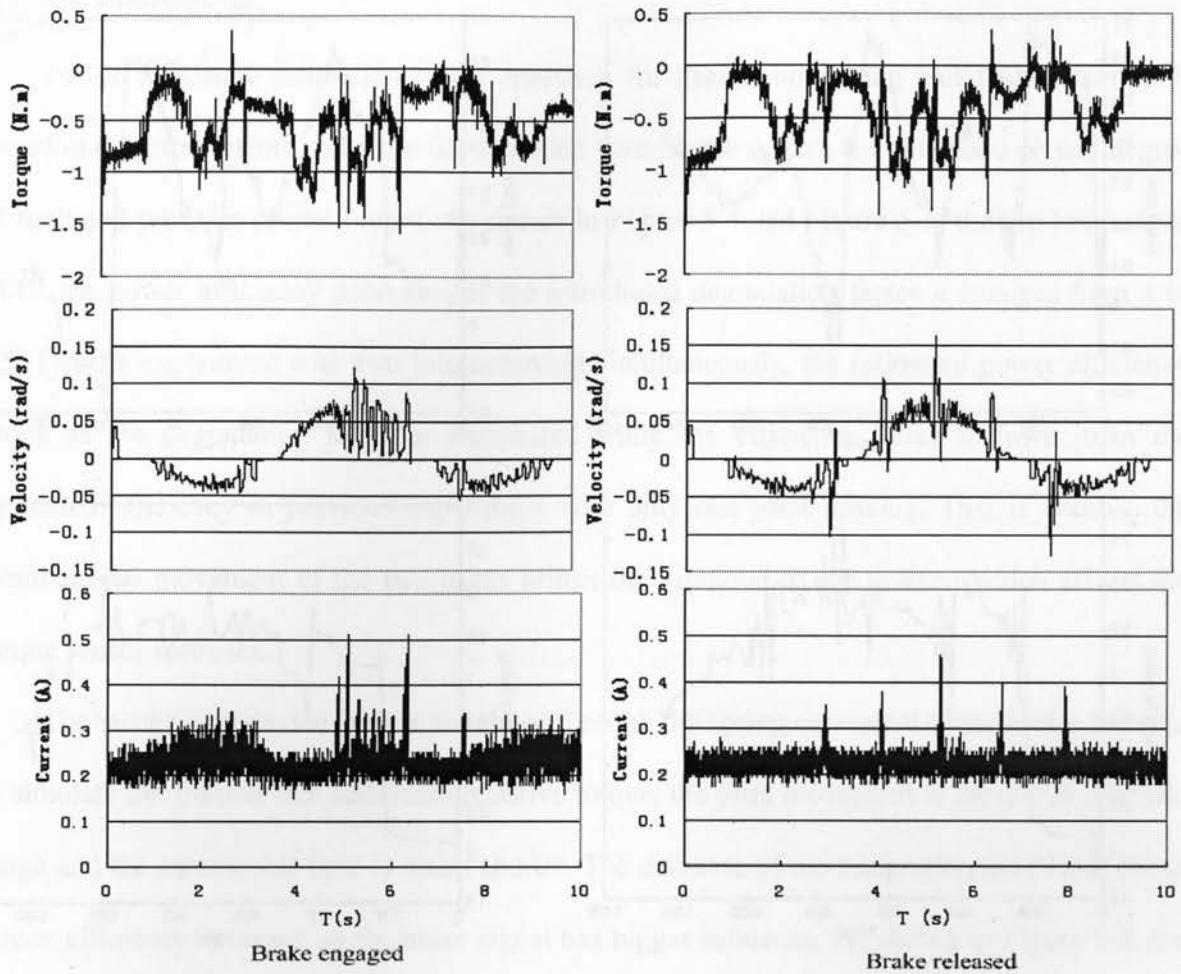


Figure 5-4: Experimental results with two joints moving together

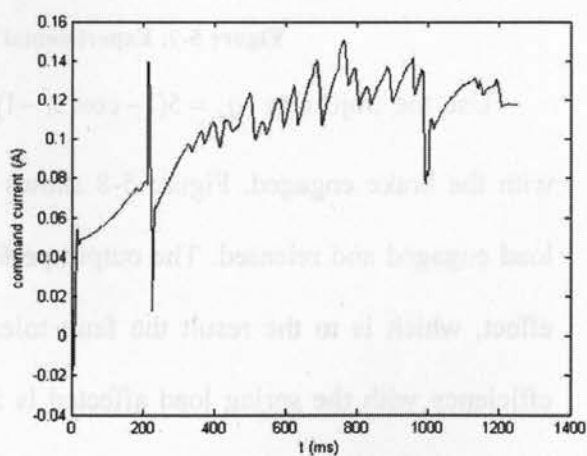
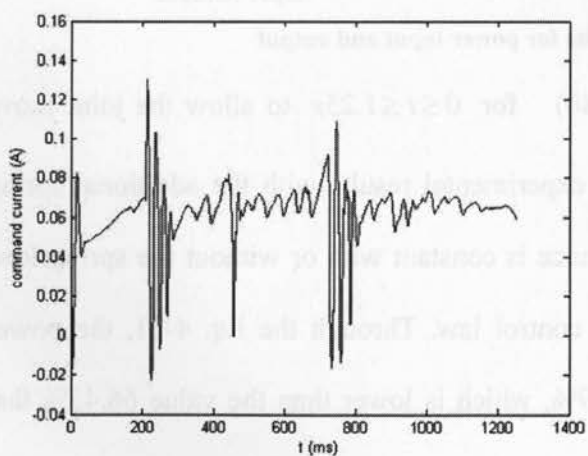
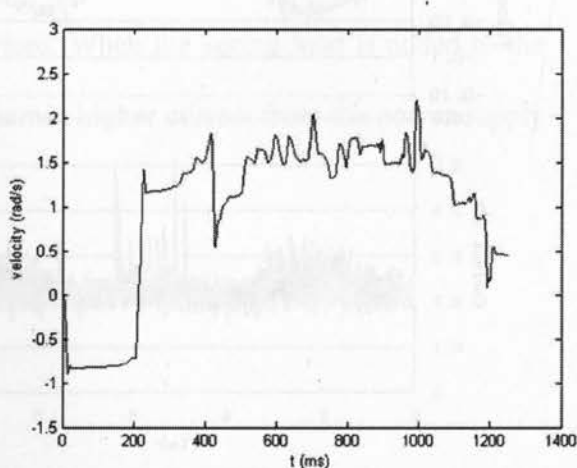
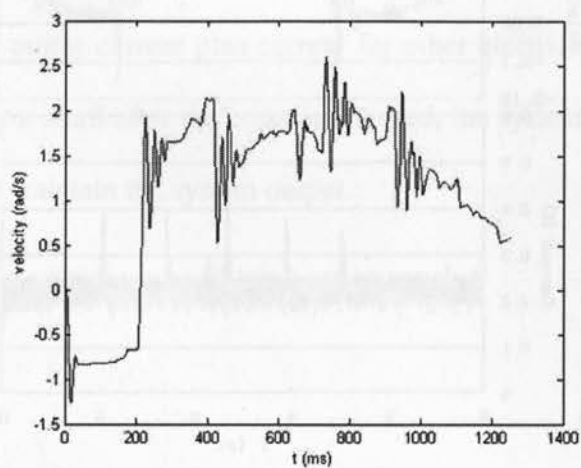
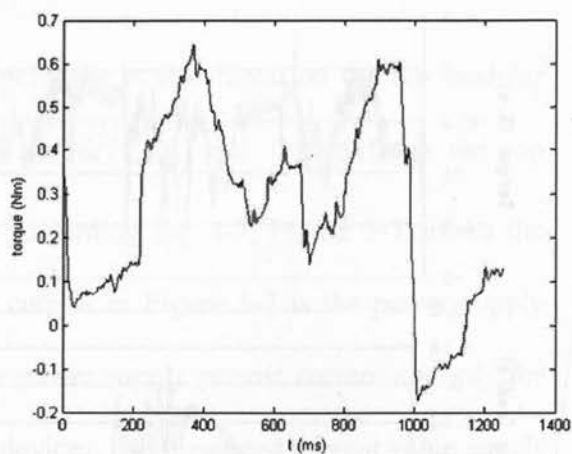
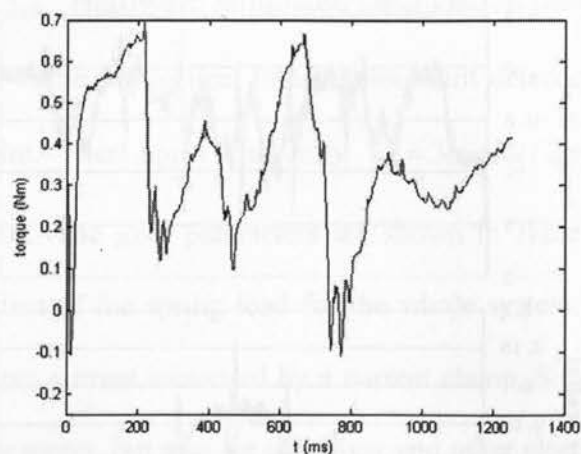
As two joints moving together, the coupling effect from lower joint has influence on the top joint, and makes the measured torque value of the top joint remains positive, as shown in Figure 5-4.



**Figure 5-7: Experimental results for power input and output**

Use the trajectory  $q_d = 5(1 - \cos(2t - 1)\pi / 40)$  for  $0 \leq t \leq 1.25s$  to allow the joint move with the brake engaged. Figure 5-8 shows the experimental results with the additional spring load engaged and released. The output performance is constant with or without the spring load effect, which is to the result the fault tolerant control law. Through the Eq. 4-21, the power efficiency with the spring load affected is 58.39%, which is lower than the value 66.45% that joint has no spring load.





Brake open

Brake engage

Figure 5-8: Experimental results with the additional spring load engaged and released

### 5.3.3 Conclusions

Power efficiency estimation based approach for health monitoring and fault detection is tested in the experiments under the fault tolerant control, the system can tolerate a certain degree of fault and provides proper output. As shown in Figure 5-4 and Figure 5-5, the top joint of the MRR the power efficiency decreases as the introduced degradation factor  $n$  changes from 1 to 0.5. For the experiment with two joints moving simultaneously, the estimated power efficiency drops as the degradation factor  $n$  introduced while the efficiency value is lower than the estimated efficiency in previous experiment with only one joint moving. This is because the simultaneous movement of the two joints brings the vibration to the system, which affects the torque sensor feedback.

The mechanical device fault is simulated through the spring connected to the brake. In order to simulate the friction like additional resistive torque, the joint movement is limited in a certain range and the experiment time is much shorter. The decrease of the integration time leads lower power efficiency accuracy as the noise signal has bigger influence. As shown in Figure 5-8, the output power keeps constant and the command input increases with the additional resistive torque caused by the spring load. As can be seen, the simulated mechanical degradation clearly produces a significant change in the mechanical power transmission efficiency, which can be used for fault detection, as well as the fault isolation in MRR mechanical transmission system. The experimental results have confirmed the effectiveness of the proposed health monitoring and fault detection schemes.

## Chapter 6 Conclusions and Future Work

### 6.1. Conclusions

The thesis work has been devoted to develop a simple, effective fault detection and health monitoring method that facilitates the realization of a fault tolerant MRR system and its low cost maintenance, utilizing as much as possible the existing system hardware has become the subject of this thesis study.

The first half of this thesis work, mainly described in Chapter 3, has been designated to the design aspects of the Ryerson MRR system. Critical components, such as the joint motor, motor driver, harmonic drive, torque sensor, encoder and Hall sensor, and the joint brake, etc., have been selected according to the design requirements. A commercial off the shelf motor controller module and a general DPS controller module have been selected to perform the modular joint controller module functions. Special interfaces between these two controller modules have been designed to take advantage of the compactness and expertise in the driver circuit designs, and the flexibility that the general DSP controller can offer for implementing the customized MRR control and its fault detection and health monitoring algorithms. This modular joint controller module design allows us to achieve the compactness of the controller module and reduced design cycle, which further allows us to have sufficient time and resource to focus on the primary subject of this thesis study, the fault detection and health monitoring of the MRR system. In addition to the hardware designs of the modular joint controller module, dynamic models, as well as the necessary control schemes, to facilitate the construction and control of the MRR are also

presented in Chapter 3.

Based on the MRR system designed in Chapter 3, the proposed fault detection and diagnosis method has been presented in Chapter 4. The potential faults and associated failure modes on mechanical and electrical components in the MRR system are classified and analyzed first. As a result, a fault model including the friction effect for the MRR system has been established. Then, a novel and effective power efficiency estimation based method intended for MRR health monitoring and fault detection has been proposed. This proposed method is based on the fact that the performance degradation and faults of various electrical and mechanical components in each MRR module can be reflected by the power efficiency degradation, and the modular joint sensor signals used for distribute control can be conveniently used for power efficiency estimation. Since the power efficiency for the electrical system and the mechanical system can be estimated separately to determine the health condition of the corresponding system, the fault isolation between electrical and mechanical systems can thus be achieved. This proposed power efficiency estimation based method is straightforward and efficient, as all the necessary information and signals required to support this method can be readily obtained from the existing MRR distributed controls.

Two carefully designed simulated degradations in both mechanical transmission and electric motor have been presented in Chapter 5. These two simulated faults have been used to test the proposed power efficiency estimation based health monitoring and fault detection strategy. The feasibility and effectiveness of this proposed method have been verified through various experiments. The experimental results are presented in Chapter 5.

It is therefore felt by the author that the original objectives of this thesis have been fully achieved.

## 6.2. Major Contributions

The major contributions of this thesis work are mainly in the following two related areas:

- 1) Working with team members, compact MRR joint modules have been developed. My major contributions in this area include development of the electronics system architecture, hardware implementation, software development, as well as integration and testing.
- 2) A novel power efficiency estimation based health monitoring and fault detection approach has been developed particularly for MRR and investigated through experiments. Furthermore, the proposed method is equipped with the ability to isolate the faults on mechanical devices and electrical subsystem. The proposed method has been studied experimentally using the developed MRR modules, and its effectiveness has been demonstrated with experimental results.

## 6.3. Future Work

Based on the research work reported in this thesis, future efforts should be oriented toward the following areas:

1) In the proposed power efficiency estimation based health monitoring and fault detection strategy, the predictable power loss, i.e. the part of the power loss due to normal operation, whose contribution to the power efficiency change should not be taken into account for evaluation of the system degradation, are assumed to be constant. However, this part of power loss does change from time to time during robot movement. More accurate evaluation of this part of power loss and elimination of its effect on the effective power efficiency change due to system degradation will certainly improve the effectiveness of the proposed method.

2) Although the trend of decreased power efficiency can be an effective indication of system





# Appendices

## Appendix A

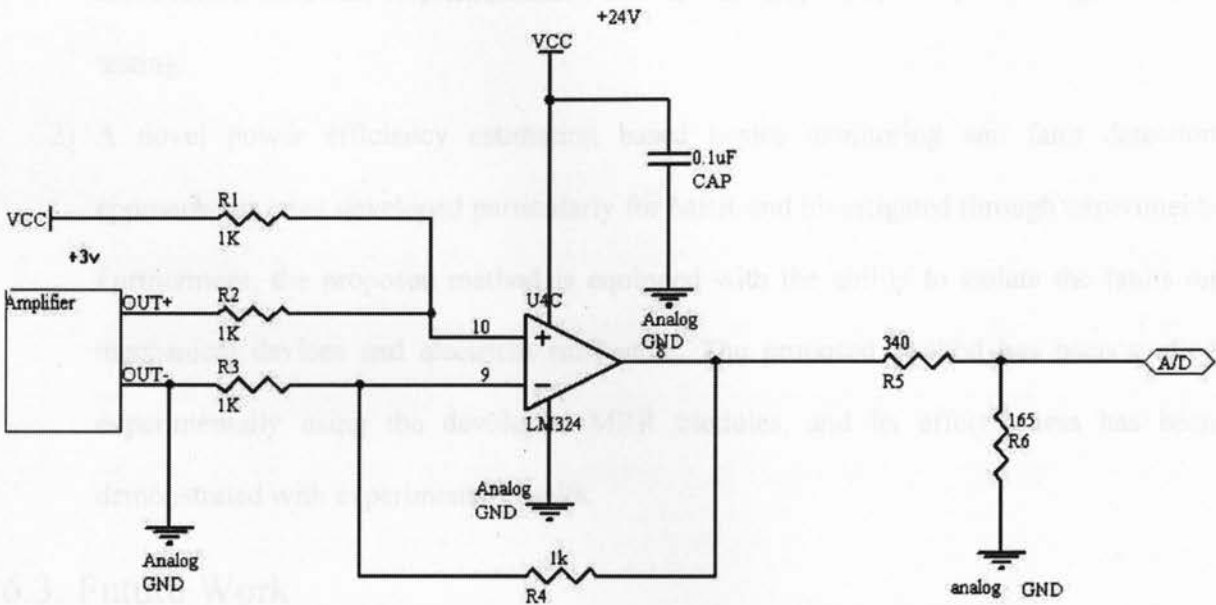


Figure A- 1: Torque sensor amplifier circuit connection

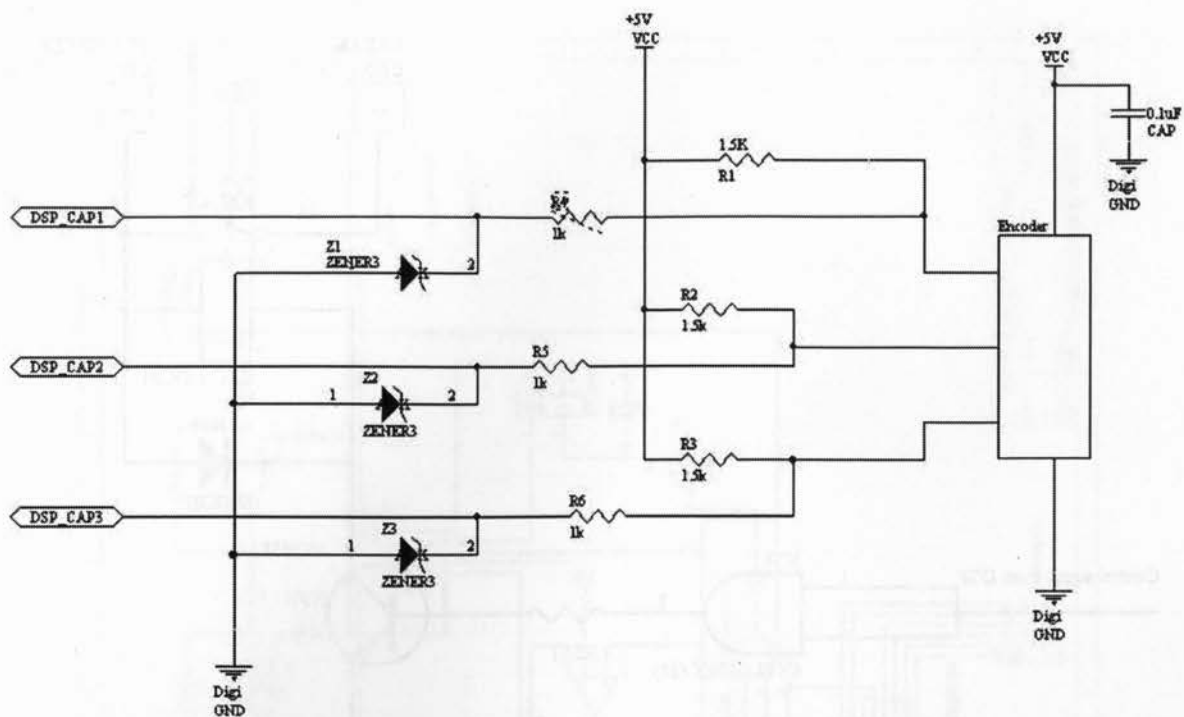


Figure A- 2: Encoder circuit connection

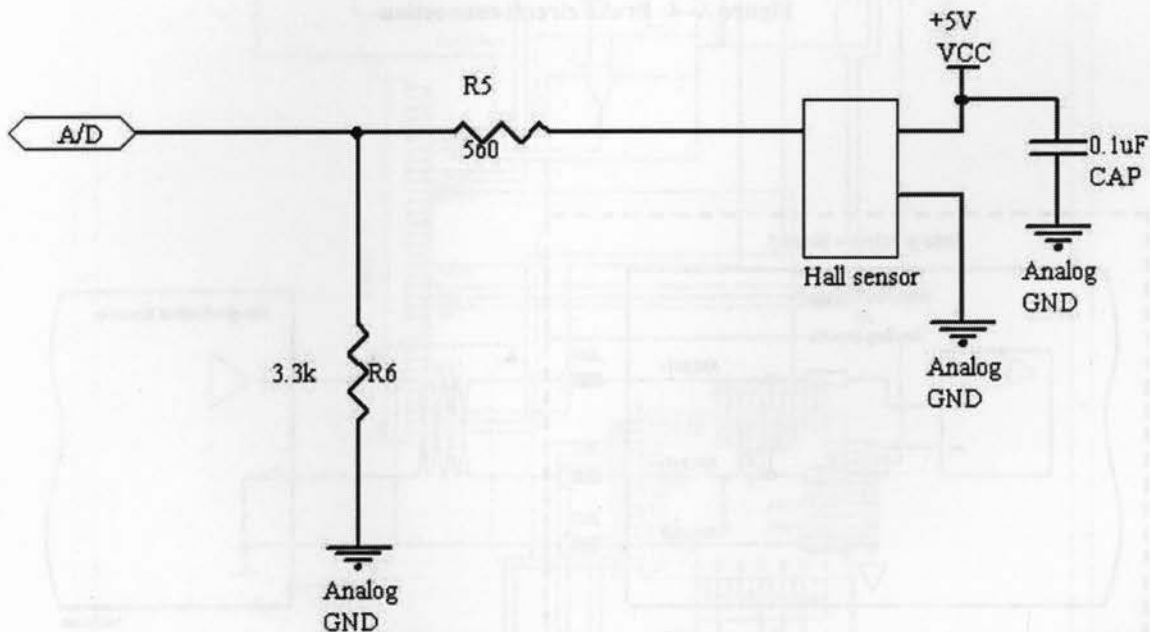


Figure A- 3: Hall sensor circuit connection

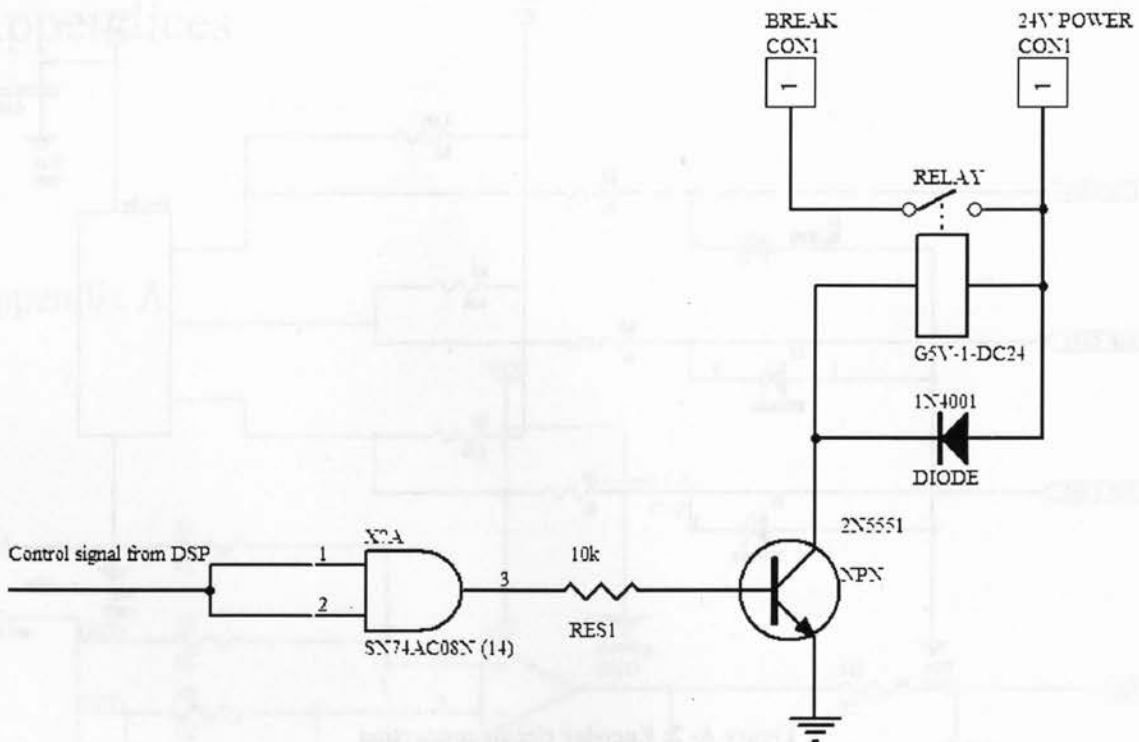


Figure A- 4: Brake circuit connection

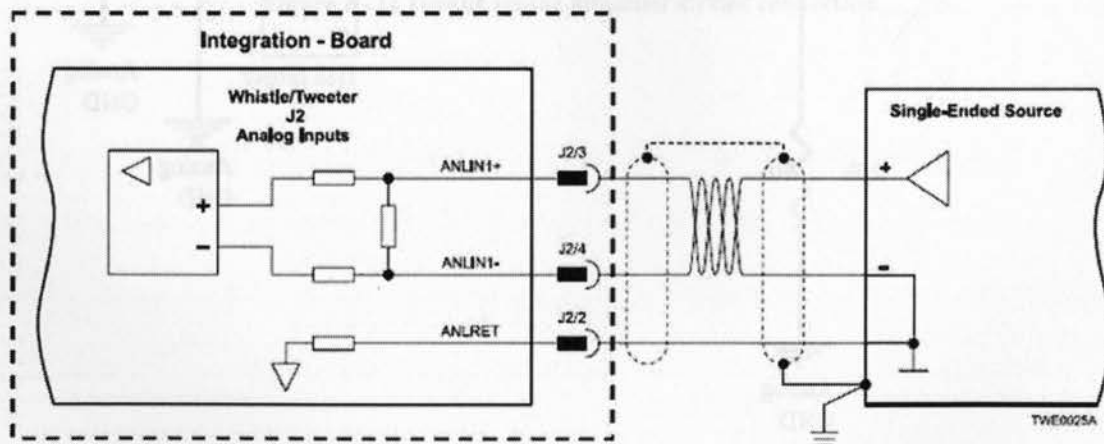


Figure A- 5: Elmo driver circuit connection

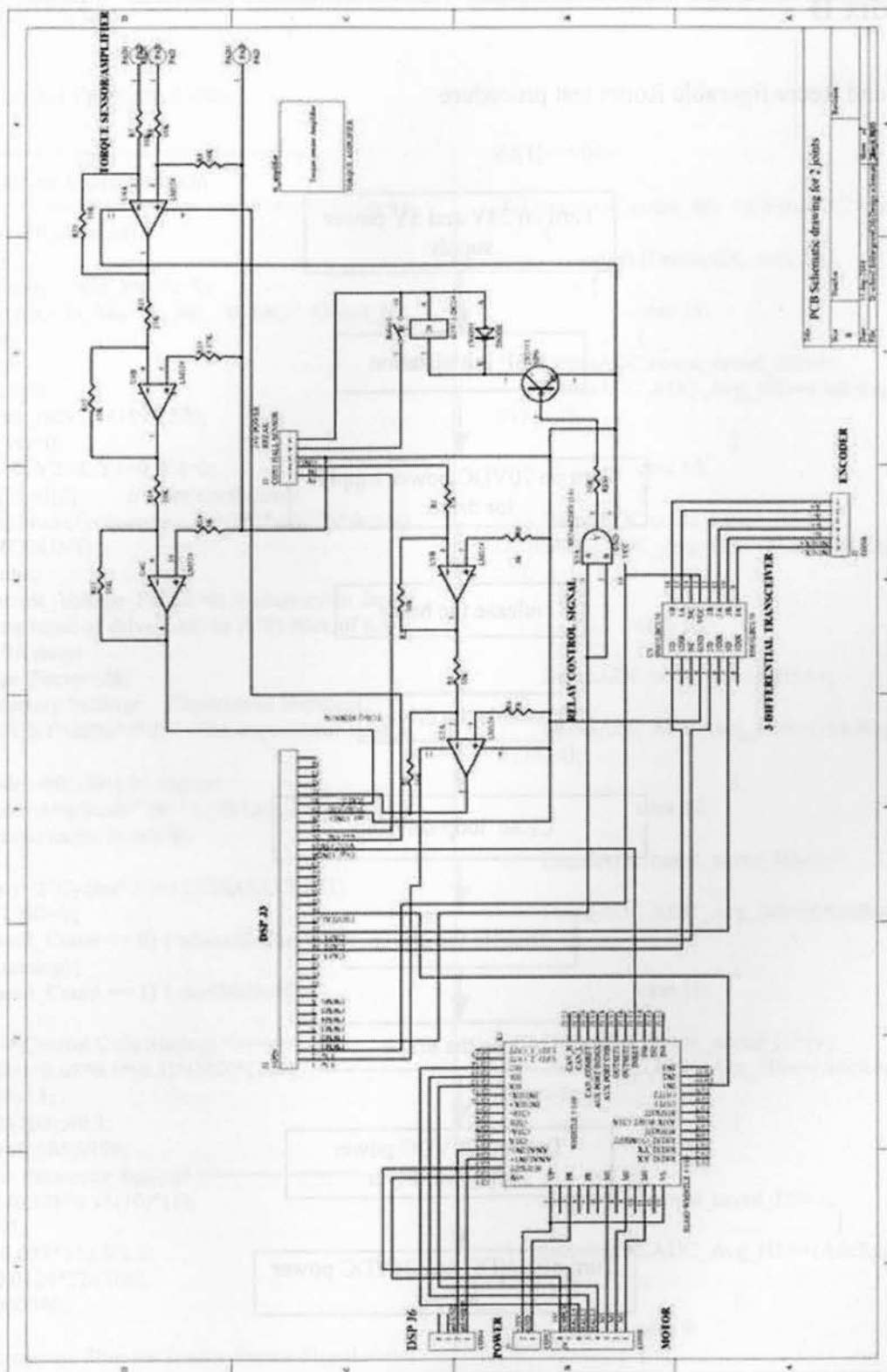
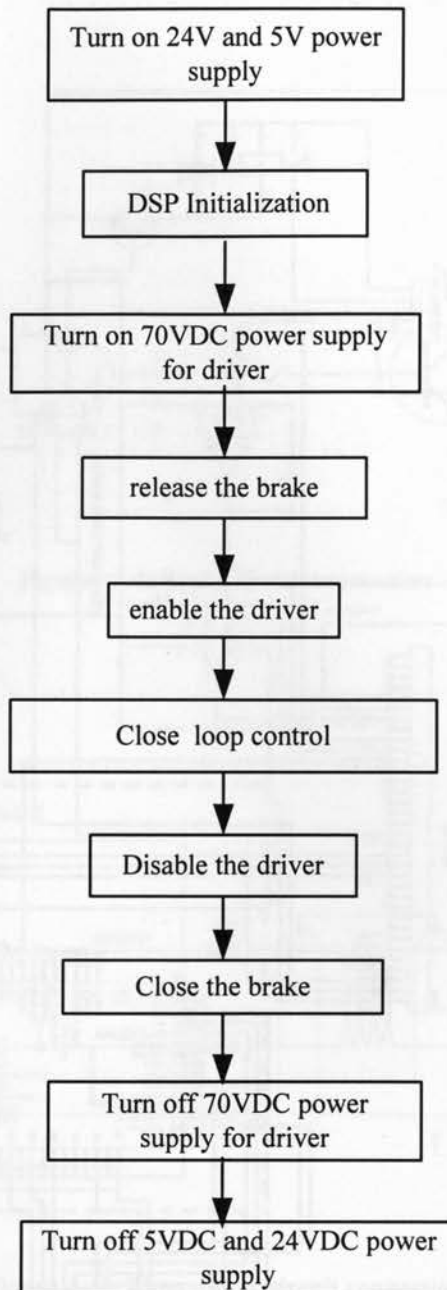


Figure A- 6: PCB board schematic

## Appendix B

### Modular and Reconfigurable Robot test procedure



**Figure B- 1: MRR2 electrical control system test procedure**

# Appendix C

## DSP Control Program Code

```
//=====
// Control Law Implementation
//=====
void ControlLaw(void) {
float K_P;
float lambda, abs_r=0;//a=0,
float Fs_hat, Ft_hat, Fc_hat, B_hat;// Offset_hat,
Gain_hat;
float Im;
float Ku, eps;
float count_rad=3.14159/1270;
int sign_vel=0;
float Y1=0, Y2=0, Y3=0, Y4=0;
float nd[2],dd[2]; // filter coefficients
float Amplitude,Frequency; // 2*pi/(500*factor)
int MAXCOUNT;
float Cycles;
float Current_Voltage_Factor=4; // conversion factor
//at analog input of drive (unit in A/V) Max of 6A/V
//-9A to 9A range
int Torque_Factor=28;
//== Trajectory Settings. Experiment Duration==
MAXCOUNT=2500*2*2; //the experiment time is
//20sec
Amplitude=-60; //link in degrees
Amplitude=Amplitude*101*1270/180; //convert link
in degrees to motor in counts
Cycles=1;
Frequency=2*Cycles*3.14159/MAXCOUNT;
LIMITFLAG=0;
if (Transmit_Count == 0) { releaseBrake(); }
if (expRunning){
if (Transmit_Count == 1) { startMotion(); }

//=====Control Gain Settings =====
K_P=0.001*0.01*0.1*(0.1)*(100)*(100);/
Ku=0.1/4;//.1;
eps=5*2/(100);//0.1;
lambda=70;//85//100;
//===== Parameter Settings =====
Fc_hat=.02558*0.1*(10)*(1);
Ft_hat=15;
Fs_hat=0.027*5*0.5/2.5;
B_hat=0.0124*22/(100);
Im=0.0000346;

//=====Low pass filter for Torque Sensor Signal =====

nd[0]=.04762; //50 rad/s
nd[1]=.04762;
```

```
dd[1]=-0.9048;

if (DemoADC.count_adc < DemoADC.n_samples)
{
switch (DemoADC.max_ch)
{
case 15:
{
DemoADC.count_saved_HS++;
DemoADC.ADC_Avg_HS+=(AdcRegs.RESUL
T15>>4);
}
case 14:
{
DemoADC.count_saved_HS++;
DemoADC.ADC_Avg_HS+=(AdcRegs.RESULT14>
>4);
}
case 13:
{
DemoADC.count_saved_HS++;

DemoADC.ADC_Avg_HS+=(AdcRegs.RESUL
T13>>4);
}
case 12:
{
DemoADC.count_saved_HS++;

DemoADC.ADC_Avg_HS+=(AdcRegs.RESUL
T12>>4);
}
case 11:
{
DemoADC.count_saved_HS++;
DemoADC.ADC_Avg_HS+=(AdcRegs.RESUL
T11>>4);
}
case 10:
{
DemoADC.count_saved_HS++;

DemoADC.ADC_Avg_HS+=(AdcRegs.RESUL
T10>>4);
}
case 9:
{

DemoADC.count_saved_HS++;
```



```

    DemoADC.ADC_Avg_HS+=(AdcRegs.RESUL
T9>>4);
        }
        case 8:
        {
            DemoADC.count_saved_HS++;

            DemoADC.ADC_Avg_HS+=(AdcRegs.RESUL
T8>>4);
        }
        case 7:
        {
            DemoADC.count_saved_TS++;

            DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T7>>4);
        }
        case 6:
        {
            DemoADC.count_saved_TS++;

            DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T6>>4);
        }
        case 5:
        {
            DemoADC.count_saved_TS++;

            DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T5>>4);
        }
        case 4:
        {
            DemoADC.count_saved_TS++;

            DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T4>>4);
        }
        case 3:
        {
            DemoADC.count_saved_TS++;

            DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T3>>4);
        }
        case 2:
        {
            DemoADC.count_saved_TS++;

            DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T2>>4);
        }
        case 1:
        {
            DemoADC.count_saved_TS++;

```

```

    DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T1>>4);
        }
        case 0:
        {
            DemoADC.count_saved_TS++;

            DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T0>>4);
        }
        default: ;
        }

    DemoADC.count_adc++; // increment counter
of A/D made conversions
    }
    else
    {
        ref_pos=0.5*Amplitude*(1-cos(3.14159*(2*Tra
nsmitt_Count-1)/(2*MAXCOUNT)));
        ref_vel=0.5*Amplitude*3.14159*500*sin(3.141
59*(2*Transmit_Count-1)/(2*MAXCOUNT))/(MAX
COUNT);
        REF_POS
=ref_pos*count_rad*180/3.14159/101; // reference
trajectory
        REF_VEL
=ref_vel*count_rad*180/3.14159/101;

//==== obtain position encoder signals =====
if (Transmit_Count<1)
{
    EvaRegs.T2CNT=0;
    Position=0;
    sum_error=0;
}
DemoQEP.position +=(int)(EvaRegs.T2CNT);
EvaRegs.T2CNT=0;
Position = DemoQEP.position;
//===== Torque and hall sensor calculation =====
Torque_Sensor=DemoADC.ADC_Avg_TS/DemoAD
C.count_saved_TS*3/4095; //average torque
sensor A/D
Torque_Sensor=((Torque_Sensor-1.5)/1.5); //217.34
Torque_Sensor=(Torque_Factor*(Torque_Sensor+0.7
35*0)); // -Offset/Gain;
Torque_Sensor=Torque_Sensor+3.5077;

hall_sensor=DemoADC.ADC_Avg_HS/DemoADC.c
ount_saved_HS*3/4095; //hall sensor
hall_sensor_deg=((2.5*3.3/4.8)-hall_sensor)*180/(4*
3.3/4.8); //hall sensor in degree

//===== buffering previous position values =====
if (Transmit_Count<=1)

```

```

{
Velocity=0;
Position=0;
ref_acc=0;
Prev_ref_vel=0;
ref_vel=0;
Nom_acc=0;
Prev_ref_pos=0;
a_1=Torque_Sensor;
Prev_velocity=0;
t_1=a_1;
PPosition=0;
prev_vel=0;
prev_num_vel=0;
}
if (Transmit_Count>1)
{
for (ii=0;ii<BCKSTEPS;ii++)

{Prev_Position[BCKSTEPS-ii]=Prev_Position[
BCKSTEPS-ii-1];
Prev_ref_Position[BCKSTEPS-ii]=Prev_ref_Pos
ition[BCKSTEPS-ii-1];}
Prev_Position[0]=Position;
Prev_ref_Position[0]=ref_pos;
}
//===== Velocity Estimation routine=====
if (Step_Est==0)
Start_Pos=Position;
Pos_Diff=Position-Start_Pos;
if ((Step_Est==2) || (abs(Pos_Diff)>3))
{
Velocity2=Pos_Diff/(Step_Est*.002)*count_rad;
Step_Est=0;
}
else Step_Est++;
ii=1;
while (ii<=BCKSTEPS)
{
Pos_Diff2=((float)(Position-Prev_Position[ii]))*
count_rad*180/3.14159/101;
if((ii==BCKSTEPS)||((abs((float)(Position-Prev
Position[ii])*count_rad*180/3.14159/101)>50 ))
{
Velocity=((float)(Position-Prev_Position[ii])/(ii*0.00
2));
DenIndex=ii;
new=(float)(Prev_Position[ii]);

break;
}
else ii=ii+1;
}

//===== numerical differentiation to obtain
//desired acceleration =====

```

```

if (Transmit_Count>=1)
{
ref_acc=count_rad*((ref_vel-Prev_ref_vel)/.002);
Prev_ref_vel=ref_vel;
num_vel=(Position-PPosition)/.002;
Nom_acc=count_rad*((Velocity-Prev_velocity)/.002)
;
PPosition=Position;
}
else
{
ref_acc=0;
num_vel=0;
Nom_acc=0;
Prev_velocity=0;
}

Velocity3=
0.000999*num_vel+0.000999*prev_num_vel +
0.998*prev_vel;
prev_num_vel = Velocity3;
prev_vel = num_vel;
//=====Calculate the Filtered Torque =====
Torque_filt =
nd[0]*Torque_Sensor+nd[1]*t_1-dd[1]*a_1;
a_1=Torque_filt;
t_1=Torque_Sensor;

//=====Calculate velocity and position errors as
//well as mixed tracking errors (NL)=====
vel_error=(-ref_vel+Velocity)*count_rad/101; // on
joint side side in rads
pos_error= (-ref_pos+Position)*count_rad/101;
r = vel_error+lambda*pos_error;
a=ref_acc/101-2*lambda*vel_error-(lambda*lambda)
*pos_error;

VEL_Link=Velocity*count_rad*180/3.14159/101;
VEL_Link2=Velocity2*180/3.14159/101;
VEL_NUM=Velocity3*count_rad*180/3.14159/101;
POS_Link=Position*count_rad*180/3.14159/101;
if ((Velocity*count_rad)==0)
sign_vel=0;
else if ((Velocity*count_rad)<0)
sign_vel=-1;
else
sign_vel=1;

//== Calculate the nominal torque =====
Nom_Tau=Im*101*a+B_hat*Velocity*count_rad/10
1+(Fc_hat+Fs_hat*exp(-Ft_hat*Velocity*count_rad*
Velocity*count_rad/(101*101)))*sign_vel;
Nom_Tau1=Im*101*a;
Nom_Tau=Nom_Tau/0.167;

```

```

DemoADC.ADC_Avg_HS+=(AdcRegs.RESUL
T9>>4);
    }
    case 8:
    {
        DemoADC.count_saved_HS++;

        DemoADC.ADC_Avg_HS+=(AdcRegs.RESUL
T8>>4);
    }
    case 7:
    {
        DemoADC.count_saved_TS++;

        DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T7>>4);
    }
    case 6:
    {
        DemoADC.count_saved_TS++;

        DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T6>>4);
    }
    case 5:
    {
        DemoADC.count_saved_TS++;

        DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T5>>4);
    }
    case 4:
    {
        DemoADC.count_saved_TS++;

        DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T4>>4);
    }
    case 3:
    {
        DemoADC.count_saved_TS++;

        DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T3>>4);
    }
    case 2:
    {
        DemoADC.count_saved_TS++;

        DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T2>>4);
    }
    case 1:
    {
        DemoADC.count_saved_TS++;

```

```

DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T1>>4);
    }
    case 0:
    {
        DemoADC.count_saved_TS++;

        DemoADC.ADC_Avg_TS+=(AdcRegs.RESUL
T0>>4);
    }
    default: ;
    }

    DemoADC.count_adc++; // increment counter
of A/D made conversions
    }
    else
    {
        ref_pos=0.5*Amplitude*(1-cos(3.14159*(2*Tra
nsmi
t_Count-1)/(2*MAXCOUNT)));
        ref_vel=0.5*Amplitude*3.14159*500*sin(3.141
59*(2*Transmit_Count-1)/(2*MAXCOUNT))/(MAX
COUNT);
        REF_POS
=ref_pos*count_rad*180/3.14159/101; // reference
trajectory
        REF_VEL
=ref_vel*count_rad*180/3.14159/101;

//==== obtain position encoder signals =====
if (Transmit_Count<1)
{
    EvaRegs.T2CNT=0;
    Position=0;
    sum_error=0;
}
DemoQEP.position +=(int)(EvaRegs.T2CNT);
EvaRegs.T2CNT=0;
Position = DemoQEP.position;
//===== Torque and hall sensor calculation =====
Torque_Sensor=DemoADC.ADC_Avg_TS/DemoAD
C.count_saved_TS*3/4095; //average torque
sensor A/D
Torque_Sensor=((Torque_Sensor-1.5)/1.5); //217.34
Torque_Sensor=(Torque_Factor*(Torque_Sensor+0.7
35*0)); // -Offset)/Gain;
Torque_Sensor=Torque_Sensor+3.5077;

hall_sensor=DemoADC.ADC_Avg_HS/DemoADC.c
ount_saved_HS*3/4095; //hall sensor
hall_sensor_deg=((2.5*3.3/4.8)-hall_sensor)*180/(4*
3.3/4.8); //hall sensor in degree

//===== buffering previous position values =====
if (Transmit_Count<=1)

```

```

{
Velocity=0;
Position=0;
ref_acc=0;
Prev_ref_vel=0;
ref_vel=0;
Nom_acc=0;
Prev_ref_pos=0;
a_1=Torque_Sensor;
Prev_velocity=0;
t_1=a_1;
PPosition=0;
prev_vel=0;
prev_num_vel=0;
}
if (Transmit_Count>1)
{
for (ii=0;ii<BCKSTEPS;ii++)

{Prev_Position[BCKSTEPS-ii]=Prev_Position[
BCKSTEPS-ii-1];
Prev_ref_Position[BCKSTEPS-ii]=Prev_ref_Pos
ition[BCKSTEPS-ii-1];}
Prev_Position[0]=Position;
Prev_ref_Position[0]=ref_pos;
}
//===== Velocity Estimation routine=====
if (Step_Est==0)
Start_Pos=Position;
Pos_Diff=Position-Start_Pos;
if ((Step_Est==2) || (abs(Pos_Diff)>3))
{
Velocity2=Pos_Diff/(Step_Est*.002)*count_rad;
Step_Est=0;
}
else Step_Est++;
ii=1;
while (ii<=BCKSTEPS)
{
Pos_Diff2=((float)(Position-Prev_Position[ii]))*
count_rad*180/3.14159/101;
if((ii==BCKSTEPS)||((abs((float)(Position-Prev
Position[ii]))*count_rad*180/3.14159/101)>50 ))
{
Velocity=((float)(Position-Prev_Position[ii])/(ii*0.00
2));
DenIndex=ii;
new=(float)(Prev_Position[ii]);

break;
}
else ii=ii+1;
}

//===== numerical differentiation to obtain
//desired acceleration =====

```

```

if (Transmit_Count>=1)
{
ref_acc=count_rad*((ref_vel-Prev_ref_vel)/.002);
Prev_ref_vel=ref_vel;
num_vel=(Position-PPosition)/.002;
Nom_acc=count_rad*((Velocity-Prev_velocity)/.002)
;
PPosition=Position;
}
else
{
ref_acc=0;
num_vel=0;
Nom_acc=0;
Prev_velocity=0;
}

Velocity3=
0.000999*num_vel+0.000999*prev_num_vel +
0.998*prev_vel;
prev_num_vel = Velocity3;
prev_vel = num_vel;
//=====Cacluate the Filtered Torque =====
Torque_filt =
nd[0]*Torque_Sensor+nd[1]*t_1-dd[1]*a_1;
a_1=Torque_filt;
t_1=Torque_Sensor;

//=====Calculate velocity and position errors as
//well as mixed tracking errors (NL)=====
vel_error=(-ref_vel+Velocity)*count_rad/101; // on
joint side side in rads
pos_error= (-ref_pos+Position)*count_rad/101;
r = vel_error+lambda*pos_error;
a=ref_acc/101-2*lambda*vel_error-(lambda*lambda)
*pos_error;

VEL_Link=Velocity*count_rad*180/3.14159/101;
VEL_Link2=Velocity2*180/3.14159/101;
VEL_NUM=Velocity3*count_rad*180/3.14159/101;
POS_Link=Position*count_rad*180/3.14159/101;
if ((Velocity*count_rad)==0)
sign_vel=0;
else if ((Velocity*count_rad)<0)
sign_vel=-1;
else
sign_vel=1;

//== Calculate the nominal torque =====
Nom_Tau=Im*101*a+B_hat*Velocity*count_rad/10
1+(Fc_hat+Fs_hat*exp(-Ft_hat*Velocity*count_rad*
Velocity*count_rad/(101*101)))*sign_vel;
Nom_Tau1=Im*101*a;
Nom_Tau=Nom_Tau/0.167;

```

```

//=====Calculate Parametric Control Value =====
Y1=Velocity*count_rad/101;
Y2=sign_vel;
Y3=exp(-Ft_hat*Velocity*count_rad*Velocity*count_rad/(101*101))*sign_vel;
Y4=(-Fs_hat*Velocity*count_rad*Velocity*count_rad/(101*101))*exp(-Ft_hat*Velocity*count_rad*Velocity*count_rad/(101*101))*sign_vel;
if (Transmit_Count>1)
{
sumY1+=r*Y1*.002;
sumY2+=r*Y2*.002;
sumY3+=r*Y3*.002;
sumY4+=r*Y4*.002;
}
else
{
sumY1=0;
sumY2=0;
sumY3=0;
sumY4=0;
}
Up=sumY1*Y1;
Up+=sumY2*Y2;
Up+=sumY3*Y3;
Up+=sumY4*Y4;
Up=Up*(-K_P);
Up=Up/0.167;
//===== Calculate Robust Control Value =====
if (r<0) abs_r=r*-1;
else abs_r=r;
if (abs_r>=eps)
Uu=-Ku*r/abs_r;
else
Uu=-Ku*r/eps;
Uu=Uu/0.167;
//===== Torque Calculation =====
Error_amp=-(Nom_Tau+Uu+Up+Torque_filt/(101*0.167))*0.5;
//===== Y() x Theta =====
Up = a*sumY1_FTC + vel_sig*sumY2_FTC + sign_vel*sumY3_FTC + Torque_Sensor*sumY4_FTC + 1*sumY5_FTC+ exp(-0.5*Velocity*Velocity/(101*101))*sign_vel*sumY6_FTC;
//=====Calculate Robust Control Value=====
//=====compensation for friction term and delta
//Fq(q,qdot) + delta ==

```

#### Initial reference Position Detection Program Code

```

//=====
//Interrupt service routine to find reference //pulse
//=====
void marking_pulse_isr(void)
{
float Kp_MP=0.5;

```

```

if (r<0) abs_r=r*-1;
else abs_r=r;
if (abs_r>=eps)
Uf=-(Ku/2)*r/abs_r;
else
Uf=-(Ku/2)*r/eps;

```

```

/***** send torque control command through
DAC*****

```

```

DAC_voltage=(Error_amp/Current_Voltage_Factor+1.5);
DAC_volt_temp=DAC_voltage;
if (DAC_voltage>=3)
DAC_voltage=3;
if (DAC_voltage<=0)
DAC_voltage=0;
DAC_output=DAC_voltage*4095/3;
//velocity control current command V
SendDACAImm(DAC_output);
//send command to DAC A with immediate update

```

```

if (Transmit_Count>MAXCOUNT )//
LIMITFLAG==1
{

```

```

DAC_output= 0x07FF; //
engageBrake();
runLevel=0;
}

```

```

else
{
Transmit_Count++;
logging();
}

```

```

/*****Reset ADC variables for next
iteration*****/

```

```

DemoADC.count_adc=0;
DemoADC.count_saved_TS=0;
DemoADC.count_saved_HS=0;
DemoADC.ADC_Avg_TS=0;
DemoADC.ADC_Avg_HS=0;
}

```

```

}
}

```

```

float count_rad=3.14159/1270;
float Current_Voltage_Factor=4;
int MAXCOUNT=100;
float Amplitude,Frequency;
Amplitude=15; //link in degrees
Amplitude=Amplitude*101*1270/180;

```



```

Amplitude=Amplitude/1.3;
Frequency=2*1*3.14159/2500;
if (Transmit_Count == 0)
{ releaseBrake(); }
if (Transmit_Count == 1)
{ startMotion(); }
if(DemoADC.count_adc<DemoADC.n_samples)
{
    switch (DemoADC.max_ch)
    {
        case 15:
        {
            DemoADC.count_saved_HS++;
            DemoADC.ADC_Avg_HS+=(AdcRegs.RESULT15>
>4);
        }

        case 14:
        {
            DemoADC.count_saved_HS++;
            DemoADC.ADC_Avg_HS+=(AdcRegs.RESULT14>
>4);
        }

        case 13:
        {
            DemoADC.count_saved_HS++;
            DemoADC.ADC_Avg_HS+=(AdcRegs.RESULT13>
>4);
        }

        case 12:
        {
            DemoADC.count_saved_HS++;
            DemoADC.ADC_Avg_HS+=(AdcRegs.RESULT
T12>>4);
        }

        case 11:
        {
            DemoADC.count_saved_HS++
            DemoADC.ADC_Avg_HS+=(AdcRegs.RESULT
T11>>4);
        }

        case 10:
        {
            DemoADC.count_saved_HS++;
            DemoADC.ADC_Avg_HS+=(AdcRegs.RESULT
T10>>4);
        }

        case 9:
        {
            DemoADC.count_saved_HS++;
            DemoADC.ADC_Avg_HS+=(AdcRegs.RESULT
T9>>4);
        }

        case 8:
        {
            DemoADC.count_saved_HS++
            DemoADC.ADC_Avg_HS+=(AdcRegs.RESULT

```

```

T8>>4);
        }

        case 7:
        {
            DemoADC.count_saved_TS++;
            DemoADC.ADC_Avg_TS+=(AdcRegs.RESULT
T7>>4);
        }

        case 6:
        {
            DemoADC.count_saved_TS++
            DemoADC.ADC_Avg_TS+=(AdcRegs.RESULT
T6>>4);
        }

        case 5:
        {
            DemoADC.count_saved_TS++;
            DemoADC.ADC_Avg_TS+=(AdcRegs.RESULT
T5>>4);
        }

        case 4:
        {
            DemoADC.count_saved_TS++
            DemoADC.ADC_Avg_TS+=(AdcRegs.RESULT
T4>>4);
        }

        case 3:
        {
            DemoADC.count_saved_TS++;
            DemoADC.ADC_Avg_TS+=(AdcRegs.RESULT
T3>>4);
        }

        case 2:
        {
            DemoADC.count_saved_TS++;
            DemoADC.ADC_Avg_TS+=(AdcRegs.RESULT
T2>>4);
        }

        case 1:
        {
            DemoADC.count_saved_TS++;
            DemoADC.ADC_Avg_TS+=(AdcRegs.RESULT
T1>>4);
        }

        case 0:
        {
            DemoADC.count_saved_TS++;
            DemoADC.ADC_Avg_TS+=(AdcRegs.RESULT
T0>>4);
        }

        default: ;
        }

        DemoADC.count_adc++;
    }
}

```



```

else
{
ref_pos=-Amplitude*(sin(Frequency*Transmit_Count)-0.5*sin(2*Frequency*Transmit_Count));
REF_POS=ref_pos*count_rad*180/3.14159/101;

ref_vel=-500*Amplitude*Frequency*(cos(Frequency*Transmit_Count)-cos(2*Frequency*Transmit_Count));
REF_VEL=ref_vel*count_rad*180/3.14159/101;
//== obtain position encoder signals ==
if (Transmit_Count<1)
{
EvaRegs.T2CNT=0;
Position=0;
Position_vel=0;
sum_error=0;
}
DemoQEP.position +=(int)(EvaRegs.T2CNT);
EvaRegs.T2CNT=0;
Position = DemoQEP.position;
hall_sensor=DemoADC.ADC_Avg_HS/DemoADC.count_saved_HS*3/4095;
hall_sensor_deg=((2.5*3.3/4.8)-hall_sensor)*180/(4*3.3/4.8); //hall sensor in dgree
POS_Link=Position*count_rad*180/3.14159/101;
if (Transmit_Count<1)
{
Velocity=0;
Position=0;
}
//send torque control command through //DAC
if (markingpulse_FLAG==1)
{Error_amp=Kp_MP*(-ref_pos+Position)*count_rad;
};
DAC_voltage=(Error_amp/Current_Voltage_Factor+1.5);
DAC_volt_temp=DAC_voltage;
if (DAC_voltage>=3)
DAC_voltage=3;
if (DAC_voltage<=0)
DAC_voltage=0;
DAC_output=DAC_voltage*4095/3;
SendDACAImm(DAC_output);

if (Transmit_Count>(MAXCOUNT*25) || markingpulse_FLAG==0)
{
DAC_output= 0x07FF;
runLevel=0;
StartPos_joint=Position;
RefPulse_Pos=hall_sensor_deg;
}

```

```

else
{
Transmit_Count++;
logging();
//==Reset ADC variables for next //iteration ==
DemoADC.count_adc=0;
DemoADC.count_saved_TS=0;
DemoADC.count_saved_HS=0;
DemoADC.ADC_Avg_TS=0;
DemoADC.ADC_Avg_HS=0;
}
}

void FindMarkingPulse(void)
{
marker = 0;
notTransition = 1;
marking_pulse=0;
markingpulse_FLAG=1;
ref_po=0;
initial_mark=0;
EALLOW;
if((GpioDataRegs.GPADAT.bit.GPIOA10==1)&&notTransition)
{

dif = marker - EvaRegs.T2CNT;
marker = EvaRegs.T2CNT;
marking_pulse++;
notTransition = 0;
}
if(GpioDataRegs.GPADAT.bit.GPIOA10==0)
{
notTransition = 1;
}
EDIS;
if ((marking_pulse==0)&&markingpulse_FLAG)
{
maker_po=0;
initial_mark=0;
}
if (marking_pulse>0&&markingpulse_FLAG){
maker_po=hall_sensor;
ref_po((((2.5*3.3/4.8)-maker_po)*180/(4*3.3/4.8)+0.5)/3.5644);
initial_mark=ref_po*3.5644;
EvaRegs.T2CNT=0;
DemoQEP.position = 0x0000;
Position=0;
markingpulse_FLAG=0;
}
exitedFMPfunc=1;

```

# References

1. T. Fukuda and S. Nakagawa, "Dynamically Reconfigurable robotic system," IEEE Proc. of Int. Conf. on Robotics and Automation, Vol. 3, pp. 1581-1586, 1988.
2. K. H. Wurst, "The Conception and Construction of a Modular Robot System," Proc. of 16th Int. Symposium on Industrial Robots (ISIR), pp. 37-44, 1986.
3. G. Liu, S. Abdul, and A. A. Goldenberg, "Distributed control of modular and reconfigurable robot with torque sensing," Robotica, Vol. 26, pp. 75-84, 2008.
4. G. Liu, X. He, J. Yuan, S. Abdul, and A.A. Goldenberg, "Development of Modular and Reconfigurable Robot with Multiple Working Modes," Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 3502-3507, Pasadena, California, USA, 2008.
5. M. Yim, D. G. Duff and K. D. Roufas, "PolyBot: a modular reconfigurable robot," IEEE Robotics and Automation Magazine, Vol. 9, no. 4, pp. 49-53, 2002.
6. A. Kamimura, H. Kurokawa, E. Yoshida, S. Murata, K. Tomita and S. Kokaji, "Automatic Locomotion Design and Experiments for a Modular Robotic System," IEEE/ASME Trans. on mechatronics, Vol. 10, no. 3, pp. 314-325, 2005.
7. A. A. Goldenberg, N. M. Kircanski, P. Kuzan, J. Wiercienski, R. Hui and C. Zhou, "Modular, Expandable and Reconfigurable Robot," Robotics and Computer-Integrated Manufacturing, Vol. 13, no. 2, pp. 173-174(2), June 1997.
8. B. Khoshnevis, R. Kovac, W. Shen and P. Will, "Reconnectable Joints for Self-reconfigurable Robots," Proc. of 2001 IEEE/RSJ Int. conf. on Intelligent Robots and Systems, pp. 584-589, 2001.
9. J. W. Suh, M. Yim and S. B. Homans, "Telecubes: Mechanical Design of a Module for Self-Reconfigurable Robotics," Proc. of the 2002 IEEE Int. Conf. on Robotics and Automation, Vol. 4, pp. 4095-4101, 2002.
10. T. Matsumaru, "Design and Control of the Modular Robot System: TOMMS," Proc. of IEEE Int. Conf. on Robotics and Automation, Vol. 2, pp. 2125-2131, 1995.
11. W. H. Lee and A. C. Sanderson, "Dynamics and Distributed Control of Tetrobot Modular

- Robots,” Proc. of IEEE Int. Conf. on Robotics and Automation, Vol. 4, pp. 2704-2710, 1999.
12. R. Hui, N. Kircanski, A. Goldenberg, C. Zhou, P. Kuzan, J. Wiercienski, D. Gershon and P. Sinha, “Design of the IRIS Facility – a Modular, Reconfigurable and Expandable Robot Test Bed,” Proc. of the IEEE Int. Conf. on Robotics and Automation, Vol. 3, pp. 155-160, 1993.
  13. T. Strasser, M. Rooker and G. Ebenhofer, “Distributed Control Concept for a 6-DOF Reconfigurable Robot Arm Innovative,” Production Machines and System 2008 Conf.
  14. G. Gustavo, C. Angulo and C. Raya, “A Multi-Agent-Based Management Approach for Self-Health Awareness in Autonomous Systems,” IEEE Int. Workshop on Engineering of Autonomic and Autonomous Systems, pp. 79-88, 2007.
  15. C. Wilkinson, D. Humphrey, B. Vermeire and J. Houston, “Prognostic and health management for avionics,” 2004 IEEE Proc. Aerospace Conf., Vol. 5, pp. 3435-3447, 2004.
  16. N. M. Vichare and M. G. Pecht, “Prognostics and Health Management of Electronics,” 2006 IEEE Trans. on Components and Packaging Technologies, Vol. 29, no. 1, pp. 222-229, March 2006.
  17. D. S. Bodden, W. Hadden, B. E. Grube and N. S. Clements, “Prognostics and Health Management as Design Variable in Air-Vehicle Conceptual Design,” IEEE Journal of Aircraft Vol. 43, no. 4, pp. 1053, July-August 2006.
  18. M. Pecht, B. Tuchband, N. Vichare and Q. J. Ying, “Prognostics and Health Monitoring of Electronics,” 2007 IEEE Int. Conf. on Mechanical and Multi-Physics Simulation Experiments in Microelectronics and Micro-Systems, pp. 1-8, 2007.
  19. M.A. Schwabacher, “A Survey of Data-Driven Prognostics,” Proc. of AIAA-2005-7002, Arlington, Virginia, September 2005. .
  20. M. Pecht, B. Tuchband, N. Vichare and Q. J. Ying, “Prognostics and Health Monitoring of Electronics,” 2007 IEEE Int. Conference on Mechanical and Multi-Physics Simulation Experiments in Microelectronics and Micro-Systems, pp. 1-8, 2007.
  21. K. Keller, K. Sweringen, K. L. Swearingen and J. Sheahan, “Aircraft Electrical Power Systems Prognostics and Health Management,” IEEE Aerospace Conf., pp. 4-11, March 2006.

22. D. C. Swanson, "A General Prognostic Tracking Algorithm for Predictive Maintenance," IEEE Proc. on Aerospace Conf., Vol. 6, pp. 2971–2977, March 2001.
23. J. Yuan, G. Liu and B. Wu, "Power Efficiency Estimation Based Health Monitoring and Fault Detection of Modular and Reconfigurable Robot," IEEE Int. Conf. on Automation and Logistics, pp. 629-634, 2008.
24. S. Abdul and G. Liu, "Decentralised Fault Tolerance and Fault Detection of Modular and Reconfigurable Robots with Joint Torque Sensing," IEEE Int. Conf. on Robotics and Automation, pp. 3520-26, Pasadena, California, USA, 2008.
25. M. Mitsuishi, "Diagnostic system for robot using a force-torque sensor," Robotersysteme Journal, Vol. 5, no. 1, pp. 40-46, 1989.
26. O. Moseler and R. Isermann, "Application of model-based fault detection to a brushless DC motor," IEEE Trans. on Industrial Electronics, Vol. 47, no. 5, pp. 1015 – 1020, 2000.
27. M. A. Masrur, Z. Chen, B. Zhang and Y. L. Murphey, "Model-Based Fault Diagnosis in Electric Drive Inverters Using Artificial Neural Network," IEEE Power Engineering Society General Meeting, pp. 1 - 7, 2007.
28. R. L. de Araujo Ribeiro, C. B. Jacobina, E. R. C. da Silva and A. M. N. Lima, "Fault detection of open-switch damage in voltage-fed PWM motor drive systems," IEEE Trans. on Power Electronics, Vol. 18, no. 2, pp. 587–593, 2003.
29. K. N. Srinivas and R. Arumugam, "Static and Dynamic Vibration Analyses of Switched Reluctance Motors Including Bearings, Housing, Rotor Dynamics, and Applied Loads," IEEE Trans. on Magnetics, Vol. 40, no. 4, pp. 1911–1919, 2004.
30. T. D. Popescu, "New Approach for Machine Vibration Analysis and Health Monitoring," IEEE Int. Symposium on Industrial Electronics, Vol. 4, pp. 2874 – 2879, 2006.
31. G. S. Maruthi and K. P. Vittal, "Electrical Fault Detection in Three Phase Squirrel Cage Induction Motor by Vibration Analysis Using MEMS Accelerometer," IEEE Int. Conf. on Power Electronics and Drives Systems, Vol. 2, pp. 838 - 843, 2005.
32. J. J. Gertler, "Survey of model-based failure detection and isolation in complex plants," IEEE Control System Magazine, pp. 3–11, 1988.
33. P. M. Frank, "Fault diagnosis in dynamic system using analytical and knowledge-based redundancy- a survey and some new results," Automatica Vol. 26, pp. 459–474, 1990.
34. J. Chen and R. J. Patton, "Robust model-based fault diagnosis for dynamic systems,"

Kluwer Academic Publishers, Boston, Dordrecht, London, 1999.

35. G. Guglielmi, T. Parisini and G. Rossi, "Fault diagnosis and neural networks: a power plant application," *Control Engineering Practice*, Vol. 3, no. 5, pp. 601-620(20), 1995.
36. R. Tinos, M. H. Terra, and M. Bergerman, "Fault detection and isolation in cooperative manipulators via artificial neural networks," *IEEE Int. Conf. Control Applications*, pp. 492 - 497, 2001.
37. A. T. Vemuri and M. M. Polycarpou, "Neural-Network-Based Robust Fault Diagnosis in Robotic Systems," *IEEE Trans. on Neural Networks*, Vol. 8, no. 6, pp. 1410-1420, 2007.
38. S. N. Huang, K. K. Tan and T. H. Lee, "Automated Fault Detection and Diagnosis in Mechanical Systems," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 37, no. 6, pp. 1360-1364, 2007.
39. H. D. Taghirad and P.R.Belanger, "Intelligent built-in torque sensor for harmonic drive systems," *IEEE Trans. on Instrumentation and Measurement*, Vol. 48, no. 6, pp. 1201-1207, 1999.
40. I. Godler, T. Ninomiya, M. Horiuchi and M. Hashimoto, "Improved performance of built-in torque sensing for harmonic drives," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vol. 2, pp. 1249-1254, 2000.
41. J. Imura, T. Sugie, Y. Yokokohji, and T. Yoshikawa, "Robust control of robot manipulators based on joint torque sensor information," *Proc. of IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, Osaka, Japan, pp. 344-49, 1991.
42. G. Liu and A. A. Goldenberg, "Robust control of robot manipulators based on dynamics decomposition," *IEEE Trans. on Robotics and Automation*, Vol. 13, no. 5, pp. 783-89, 1997.
43. G. Liu and A. A. Goldenberg, "Comparative study of robust saturation control of robot manipulators: analysis and experiments," *Int. Journal of Robotics Research*, Vol. 15, no. 5, pp. 473-91, 1996.
44. J. Lunze and M. Staroswiecki, "Fault-diagnosis systems: an introduction from fault detection to fault tolerance," *Springer-Verlag Publishers*, pp. 3, 2006.
45. T. Kenjo, "Permanent magnet and brushless dc motors," *Oxford Publishers*, pp. 58, 1985.

② BL-70-63