INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

NOTE TO USERS

The original manuscript received by UMI contains pages with slanted print. Pages were microfilmed as received.

This reproduction is the best copy available

UMI

A NOVEL, NEURO-MUSCULAR, ACTUATOR CONTROL SCHEME

By

ZHAN SHENG, B.Eng.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of
Master of Engineering

Department of Mechanical and Aerospace Engineering
Ottawa-Carleton Institute for Mechanical and Aerospace Engineering

Carleton University
Ottawa, Ontario
September 8, 1997
© copyright
1997, Zhan Sheng

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-27005-X

Canada

The undersigned recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis:

“A Novel, Neuro-Muscular, Actuator
Control Scheme”

Submitted by Zhan Sheng
in partial fulfillment of the requirements for
the degree of Master of Engineering

A handwritten signature in black ink, appearing to read 'Donald L. Russell', written over a horizontal line.

Professor D. L. Russell
Thesis supervisor

A handwritten signature in black ink, appearing to read 'Robert Bell', written over a horizontal line.

Professor R. Bell
Chair, Department of Mechanical and Aerospace Engineering

Carleton University
September 8, 1997

ABSTRACT

This thesis presents the design and analysis of a novel actuator trajectory control scheme that draws on the strategies used by the human central nervous system (CNS) to deal with constraints. The hypothesis is that, while the manipulator moves closely along the desired path, it may take advantage of the constraint (if it exists) to minimize its own power output to the environment. A muscle-like model of the actuator is developed in order to demonstrate the motion, while the controller is structured in a manner similar to the human CNS with both an alpha and a gamma command path. The function of the alpha global task control is accomplished by a proportional plus integral (PI) controller, whereas the local stability function of the gamma system is emulated with a novel perturbation controller. The perturbation controller minimizes work output by continuously testing the response of the system to an application of a minute force pulse. The single degree of freedom systems are modelled using MATLAB® and SIMULINK®. The control scheme utility is tested by running simulations of trajectory control systems configured with both an unconstrained single actuator system, and a cooperative dual actuator system. The simulation results demonstrate that the perturbation controller shows promise for effectively resolving the common difficulties realized when coordinating a pair of independent actuators.

ACKNOWLEDGMENTS

I wish to thank my thesis supervisor, Dr. D. Russell, for his assistance and support in my graduate studies.

A special thanks to my sister, Joy, and my brother-in-law, Lawrence, for their unlimited support.

Finally, I owe my heart and gratitude to my husband, Steven, who has always been there to show me his love and understanding.

DEDICATION

To my parents, Chengde Sheng and Zhuohua Yue.

TABLE OF CONTENTS

Acceptance Form	ii
Abstract	iii
Acknowledgments	iv
Dedication	v
Table of Contents	vi
List of Tables	viii
List of Figures	ix
 CHAPTER 1 INTRODUCTION	1
1.1 Robotic Systems	1
1.2 Motivation	3
1.3 Objectives	4
1.4 Overview	6
 CHAPTER 2 THE NEURO-MUSCULAR SYSTEM: SOME BACKGROUND INFORMATION	8
2.1 Introduction.....	8
2.2 Muscle Mechanics	8
2.3 Nervous System	13
2.4 Motor Control System: Neural Control Aspects	18
 CHAPTER 3 SOME IMPORTANT TOPICS IN ROBOTIC CONTROL SYSTEMS: A LITERATURE SURVEY	22
3.1 Introduction	22
3.2 Constrained Manipulation	23
3.3 Contact Stability	27
3.4 Redundancy	30
3.5 Comparison Between Robotic and Human Manipulation.....	32
 CHAPTER 4 A MECHANICAL SYSTEM MODEL FOR THE ROBOTIC CONTROL SYSTEM	36
4.1 Introduction	36
4.2 Muscle Models	38
4.3 Environment Structure	45
4.4 Model Parameters Identification	46
4.5 Analysis of The Open Loop Muscle-Environment Models..	47
4.6 Conclusions	64

CHAPTER 5 DESIGN OF A NOVEL ROBOTIC CONTROL SYSTEM	66
5.1 Introduction	66
5.2 An Emulation Model of Neuro-Muscular Control System ..	67
5.3 Design of the Control Strategies	72
5.4 Design of the Observers	85
5.5 The Combination of Control Law and Observers	90
5.6 Summary	91
CHAPTER 6 SIMULATION	94
6.1 Introduction	94
6.2 Simulation Parameters	95
6.3 Control System Parameters	97
6.4 Cooperative System	114
6.5 Result Summary	136
CHAPTER 7 CONCLUSIONS AND RECOMMENDATIONS.....	142
7.1 Summary.....	142
7.2 Conclusions.....	143
7.3 Recommendations.....	144
BIBLIOGRAPHY.....	146
APPENDIX A MATLAB MFILES FOR CHARACTERIZING MODEL 1 THROUGH 4.....	150
APPENDIX B MATLAB MFILES DEFINING THE SYSTEM AND SIMULATION PARAMETERS FOR SINGLE ACTUATOR SYSTEM.....	153
APPENDIX C SIMULATION DIAGRAMS FOR SINGLE ACTUATOR SYSTEM	156
APPENDIX D MATLAB MFILES DEFINING THE SYSTEM AND SIMULATION PARAMETERS FOR DUAL ACTUATOR SYSTEM.....	162
APPENDIX E SIMULATION DIAGRAMS FOR DUAL ACTUATOR SYSTEM	165

LIST OF TABLES

4.1	Final selection of values for the parameters of the mechanical model	47
6.1	Final selection of values for the control system parameters	112

LIST OF FIGURES

1.1	Block diagram of a basic robotic system	1
1.2	Comparison of human and robotic manipulation	2
2.1	Simplified anatomy of the muscle and the tendon	10
2.2	A simplified mechanical model of muscle and tendon	12
2.3	Muscle proprioceptors and their pathways	14
2.4	A simplified “wiring diagram” of the alpha and gamma motor systems in the central nervous system	16
2.5	Block diagram of neuro-muscular feedback control system	18
3.1	Two manipulators handling a common object	26
3.2	Binding of the manipulators (left) and crushing of the object (right)	26
4.1	Hill’s functional model of the muscle	39
4.2	Muscle Model 1	41
4.3	Muscle Model 2	42
4.4	Muscle Model 3	44
4.5	Muscle Model 4	44
4.6	The environment structure	45
4.7	An s-plane plot for a general pair of complex conjugate poles and a single pole on the real axis	49
4.8	The mechanical system as a combination of Model 1 and the environment structure	51
4.9	The step response of Model 1	54
4.10	The mechanical system as a combination of Model 2 and the environment structure	55
4.11	Comparison of step response of Model 2 with Model 1	57
4.12	The mechanical system as a combination of Model 3 and the environment structure	58
4.13	Comparison of step response of Model 3 with Model 1	60
4.14	The mechanical system as a combination of Model 4 and the environment structure	61
4.15	Comparison of step response of Model 4 with Model 2	63
5.1	A simplified block diagram of our robotic control system	68
5.2	A block diagram of the actuator system	70
5.3	Block diagram of a proportional controller	74
5.4	Step response of the proportional controller with $K = 5$ and $K = 10$	75
5.5	Block diagram of a proportional plus integral (PI) controller	76

5.6	Step response of the proportional plus integral controller with $K = 5$, $K_i = 50$	78
5.7	Root locus in terms of K	78
5.8	Block diagram of the perturbation controller	83
5.9	Block diagram of a system and its observer	86
5.10	A block diagram representation of the closed-loop observer (Observer1) ...	88
5.11	A block diagram representation of Observer2	89
5.12	Block diagram of the whole system which includes controllers, actuator system, and observers	90
6.1	The reference path command	96
6.2	Step response of the proportional controller with $K = 5$	99
6.3	Step response of the proportional plus integral controller with $K = 5$ and $K_i = 50$	101
6.4	Step response for other value set of K and K_i	102
6.5	Typical closed-loop response to the path command, x_d , with $K = 5$, $K_i = 50$	103
6.6	Comparison of the actual system output with the estimated output state	105
6.7	Description of the force pulse train	104
6.8	Response of the single actuator system with the perturbation controller $G = -1$, $G = -10$, and $G = -100$	108
6.9	Single actuator system response to the path command for $G = -100$	111
6.10	Single actuator system response to the path command for $G = -10$	113
6.11	Block diagram of two parallel robots configured in a cooperative system	115
6.12	Dual actuator response for Test 1: balanced path tracking	117
6.13	Dual actuator response for Test 2: unbalanced static position holding for $G = -10$	119
6.14	Dual actuator response for Test 2: unbalanced static position holding for $G = -50$	122
6.15	Dual actuator response for Test 2: unbalanced static position holding for $G = -80$	124
6.16	Dual actuator response for Test 3: PI-only controller path tracking	126
6.17	Dual actuator response for Test 4: path tracking with an initial unbalance	129
6.18	Dual actuator response for Test 5: non-identical robots in a cooperative system with $B_{p2} = 60\text{Ns/m}$	131
6.19	Dual actuator response for Test 6: non-identical robots in a cooperative system with $G2 = -50$	134
6.20	Dual actuator response for Test 7: output position calibration error with the perturbation controllers deactivated	137
6.21	Dual actuator response for Test 7: output position calibration error with active perturbation controllers	139

CHAPTER 1

INTRODUCTION

1.1 ROBOTIC SYSTEMS

Most industrial robotic systems consist of four subsystems as follows:

- 1) the mechanical arm (manipulator) subsystem;
- 2) the controller subsystem;
- 3) the drive (actuator) subsystem, and;
- 4) the recognition (sensor) subsystem.

As shown in Figure 1.1, the manipulator motion is effected by the force (or torque), $F(t)$, generated by the actuator. Furthermore, the controller determines the command input to the actuator, $U(t)$, by comparing the detected manipulator output with the desired robot output. The actuator converts the command input, $U(t)$, into force, $F(t)$, to move the manipulator. The recognition subsystem uses various sensors to detect the robot's state and feeds this information back to the controller.

Figure 1.1 Block diagram of a basic robotic system.

The performance of industrial robots can be enhanced through the improved design of manipulators, actuators, controllers, sensors, and, of course, any combination of these.

Robots may resemble humans in their ability to carry out various activities, as illustrated with Figure 1.2. A person's arm is moved by the muscles that are controlled by the brain. The brain assembles data from the senses, processes that information, and transmits command signals along the nerves to the muscles, causing the arm to move. For the robot, the manipulator is moved by the actuator and the function of the brain is fulfilled by the controller which sends signals to the actuator.

Figure 1.2 Comparison of human and robotic manipulation (source unknown).

1.2 MOTIVATION

The neuro-muscular control system used by the human to perform tasks is hypothesized to be close to optimal in design. By “optimal”, we mean that the manner by which a human performs a task appears to tend towards maximum efficiency of effort, energy, time, space or some combination of these. For example, a human being can generally lift a cup of drink to his mouth in a single, rapid and fluid motion without spilling any of the drink. One goal in the field of robotics is that robots could be this smooth, flexible and adaptive. Unfortunately, modern conventional mechanisms fail to meet this level of performance. For example, it is difficult for a robot to perform a complex task in a smooth, fluid motion. Rather, the robotic motion tends to be jerky as it is usually comprised of a series of distinct motions.

Most successful applications of industrial robots have drawn on position control strategies, in which the robot is treated, essentially, as an isolated system. However, many manipulation tasks to be performed by one or more robots fundamentally require dynamic interaction. In the example of lifting a cup of liquid, experience allows us to predict that controlling the action by merely specifying the start and finish points would inevitably lead to spillage of the drink. This is because we have failed to take account of the dynamics of the fluid system within the cup. Robotic experience has clearly demonstrated that contact between the robot and its environment can result in unstable behaviour, which is also called “contact instability” [9]. Contact instability, as its name implies, is the phenomena of a loss of stability when the robot contacts, or couples to its environment. This can happen because the interaction between the robot and its environment changes the dynamics of overall system such that the response is much different from that of the robot alone. Yet, human beings may contact environments and manipulate objects with no evidence of contact instability. So, the desire is to make robots function as well as humans do from this perspective. From our discussions, we recognize that this will fundamentally require the advancement of contemporary robot control strategies.

If one of the goals in robotics today is to design a robot that interacts well with its environment, it seems only natural to study human behaviour in this respect. However, the design of the human structure and the identification of its control system have proven to be very complex. This ultimately sets a limitation on the ability to match human manipulation capability in robotic systems. Our state of knowledge of human structure and control systems is one key to enhanced robot performance. So, the human muscle, its functions, the mechanics of its motion, and, the neuro-control system are all topics that, if well understood, could aid in advancing the state of the art in robotics today. The work in this thesis is based on the notion of designing a robot system that emulates some of the characteristics that make human control strategies superior to those of contemporary robotic systems.

1.3 OBJECTIVES

In practice, the potential instability of dynamic systems is often the most troublesome aspect of control system design, particularly for robots. Because stability is of such importance, the focus of this thesis will be on investigating controller and actuator subsystems for robots that avoid contact instability, using a structure similar to that found in human neuro-muscular system.

The control subsystems of robots will be categorized into two classes:

- 1) the global task control subsystem, and;
- 2) the local actuator control subsystem.

The global task control system functions to ensure that the manipulator successfully completes the various desired tasks. A large amount of research has been done in this field, and we will review this effort in Chapter 3. The local actuator control system

generates the necessary signals to drive any individual actuator. The actuator control signals should be adjusted in such a way that the individual actuator effort is minimized. This should ensure that the whole system remains stable. This second class of the control subsystem has been addressed by relatively few researchers. A significant portion of this thesis is devoted to addressing this second class of subsystem and the interaction of the two classes.

When a human performs a task within a given environment, he does so using a complex control loop. We recognize that the human controller is an efficient system, and we wish to emulate it for robot control. So, it is necessary to know, in detail, the functions of the human motor control loop. One objective of this thesis is to investigate the human's neuro-muscular system and understand the features that make human performance superior to conventional control systems. These characteristics are to be modelled so that they can be incorporated into an improved design of a robot actuator and control system.

Two major tasks are undertaken with respect to understanding the human neuro-muscular system. The first task is to discuss the mechanism of muscle contraction, and develop some mechanical models of the muscle actuator. The resulting model should emulate certain human characteristics and enable the incorporation of these characteristics into robotic actuator system design. The second task, which is also the most important, is to deal with how the movements of a human being are controlled. This includes a model of how the neural control system is organized to control movement, and a description of how the sensory information from the muscle is used in the control of movement. Using the knowledge learned from the investigation of the neural control system, we propose a strategy of control which is based on certain features of the neuro-muscular control system. The development of this control strategy represents progress in robot control subsystem design. Finally, a feedback control system, as a whole, is designed in which the characteristics of human muscles are used in

the design of the actuator, and in which the neural control algorithms are used as a basis for controlling the actuator.

Simulation is an important tool in the modern design and testing of control systems. We will make use of this tool extensively in this work. By simulating the whole system, we find that the knowledge gained from the study of human systems can be used to improve robot design.

1.4 OVERVIEW

The thesis is divided into three phases. The first phase consists of the review of research into the neuro-muscular system. From this, we gain an understanding of the mechanics and control of human movement. An overview of these findings is presented in Chapter 2. It includes a summary of the investigations of human muscle properties, some background information on the human nervous system and a description of the function of the motor control system.

The second phase, which is presented in Chapter 3, reviews the work which has been done by scientists and engineers in the study of the human motor system and in robotics. The last phase includes two components, the first being the analysis of models of muscle, (i.e. the definition of the actuator configuration), which is covered in Chapter 4. The second component involves the design of the control system which is detailed in Chapter 5. The design is then simulated by using the computer software packages of MATLAB ® and SIMULINK® on a UNIX computer system in Chapter 6. The graphical results obtained from our simulations are used to evaluate the behaviour of our controller and actuator systems.

We recognize that the improvement of robotic systems must be made in small, manageable steps. This work explores one possibility for emulating human neuromuscular control systems in robot design. The work is summed up in Chapter 7 with conclusions and recommendations for future work that will assist in applying the results of this work to the practical advancement of the state of the art of robotic system design.

CHAPTER 2

THE NEURO-MUSCULAR SYSTEM: SOME BACKGROUND INFORMATION

2.1 INTRODUCTION

One of the most characteristic features of human beings is their fluid and agile mechanical motion. We know that control of movements requires the cooperation of the nerves, the muscle and the skeleton. Human movements are effected by the muscles which are innervated by the motor nerves which are excited by the nervous system in response to some sensory input to the brain. Excitation of the motor nerves is followed by contraction of the muscles. So, we see that the neuro-muscular system is a closed-loop control system. Hence, the engineering model of the neuro-muscular system will involve both mechanics and control theory. Before human characteristics can be applied to the design of a robot, the mechanisms and control of human muscles must be reasonably well understood. This chapter primarily describes the neuro-muscular system in order to develop this necessary understanding. It contains some background information on mechanical models of human muscles, the nervous system, as well as the motor control system.

2.2 MUSCLE MECHANICS

The muscles can be broadly separated into three types [40]:

- 1) Striated (skeletal) muscle: it is responsible for movement. The skeletal muscle fibers appear striated in the light microscope, hence we call it the striated muscle.
- 2) Cardiac muscle: it moves blood through the heart.
- 3) Smooth muscle: it helps to maintain the internal control of the body.

In this thesis, we only concern ourselves with the skeletal muscle, because it is the actuator which drives limb movements.

A great deal of muscle-related research has been done since the late 1930's, however most of it was for medical purposes and is too in-depth to be of interest here. The construction of a mechanical-mathematical model of the muscle is the topic we are concerned with. In the past, many models have been suggested for the description of skeletal muscle. The existing models of the muscle can be divided into two basic groups:

- 1) Structural models: based on well-known and hypothetical notions about the micro-structure of the muscle and interaction between its basic biological components.
- 2) Phenomenological models: based on the principle of behavioural representation of muscle mechanics.

Theoretically, the muscle models that are built up directly from the structural considerations should represent the most accurate choice because of their explicit nature. However, these kinds of models involving detailed physiological parameters are too intricate for our purpose as they result in overwhelming complexities for practical application. Consequently, our concentration is on the more tractable, phenomenological, models of the muscle. In the phenomenological models, some components which serve to represent the observed behaviour of the muscle are not necessarily realized from biological micro-structure of real muscle. However, we do not wish to imply that the biological system is purely mechanical; with the phenomenological models of the muscle, our goal is to provide a convenient frame of reference for understanding the mechanics of muscle to improve the design of robotic systems.

The phenomenological muscle models range in structure from simple models such as simple linear viscoelastic models to extremely complex combinations of linear and nonlinear elements. Although the more elegant muscle models hold promise for accurately describing many of the recognized characteristics of muscle, their use in even moderately sized problems appears to lead to large computational costs. In this thesis, we choose a well-known muscle model originally based on Hill's model [19], developed in 1938, which is physically and mathematically representative of the relevant muscle characteristics and yet simple enough as to be tractable in analysis. We begin the description of this model by first examining the biophysical structure of a muscle.

Figure 2.1 Simplified anatomy of the muscle and the tendon. The spindle and Golgi tendon organ are shown to a greatly enlarged scale, and only one of each organ is represented.

The root “-fusal” means fusiform or spindle-shaped; so intrafusal fibers are muscle fibers within the spindle, and the extrafusal fibers are those outside the spindle -- i.e., the fibers of the muscle in which the spindle is embedded.

Figure 2.1 shows the basic anatomy of the skeletal muscle and the tendon. As we see, the whole muscle is composed of the extrafusal and the intrafusal fibers. The extrafusal fibers contain a large number of myofibrils, which are the contractile elements of the muscle that produce the force which is transmitted to the bone. Hence, it is this part of the muscle that is primarily responsible for the generation of force in the human limbs.

Situated between the extrafusal fibers of the muscle are the spindle organs that are connected in parallel with the muscle and attached at both ends to the main muscle mass, (i.e. the extrafusal fibers). The spindle is made up of the intrafusal fibers which contain relatively little contractile material. As shown in Figure 2.1, the spindle includes two major parts: nuclear bag and nuclear chain. Regardless, the basic function of the spindle is to transduce the muscle stretch and the rate of the stretch. Another organ set in the muscle structure are the Golgi tendon organs which are located in the tendon near the muscle-tendon junction. These organs lie in series with (i.e. they are between) the muscle and the tendon that attaches the muscle to the bone and serve to transduce the total force generated by the muscle.

The spindle and the Golgi tendon organ are sensory receptors. They modify and even control many aspects of the muscle behaviour, as we will see in next two sections. In the meanwhile, we are now prepared to present a phenomenological mechanical model of the muscle.

A conceptual model of a muscle is shown in Figure 2.2. This four element model assumes that the whole muscle is comprised of a simple force generator (FG) in parallel with both a spring (PS--parallel stiffness) and a dashpot (PD--parallel damping) and in series with a second spring (SS--series stiffness). Comparing Figure 2.1 and Figure 2.2, we can see that the FG represents the gross effect of the extrafusal contractile elements in the muscle, because it gives rise to the force. We also recognize that the muscle

behaviour is not consistent with a pure force generator, so we introduce some additional elements to the model. The PS represents the spring-like elastic properties of the muscle and its behavioural presence is thought to arise from the sarcolemma (muscle cell membrane) and the connective tissue surrounding the muscle fibers. The PD represents the viscous mechanism in the muscle for damping oscillations caused by the PS. Of course, the dashpot element will dissipate energy just like human muscles also dissipate heat (which, of course, makes one perspire when the muscles are exerted). Finally, the SS represents the elastic qualities of the tendon which is in series with the muscle mass of Figure 2.1.

Figure 2.2 A simple mechanical model of muscle and tendon.

This model is simple but well approximates the results of actual experiments on whole muscles [34]. This model also incorporates the passive and active characteristics of the muscle behaviours. The FG is active and the parallel and series elastic and damping components account for the passive properties of the muscle. In the active mode, the FG is generating a force. In the passive mode, the FG generates no force, yet the other components may still be dilating, which implies a change in the energy state of the system.

Obviously, we can add much more detail to this model in an effort to more closely represent specific muscular behaviour. In fact, we will analyze some other models in Chapter 4. However, we find that the above model is adequate as a basic foundation for our study. Other models may result in a more complex system, with little added benefit towards understanding the fundamental dynamic behaviour of the muscle-like actuator system.

2.3 NERVOUS SYSTEM

When muscles operate in a human being, they are controlled by the nervous system. The nervous system includes the central nervous system and the peripheral nervous system. The central nervous system (CNS) includes the brain and the spinal cord, and it has the primary function of control. The peripheral nervous system consists of *afferent* (sensory) nerves which convey information to the CNS from the sensory organs (including those within the muscle structure), and the *efferent* (motor) nerves which convey instructions from the CNS to the muscle.

As we know, the control of the human movement is closed-loop, and the physiological sources of the feedback signals are the various sensory receptors. The CNS collects information from these sensors that respond to internal and external stimuli and uses it to generate motor commands. There are three main kinds of sensory receptors in human beings [40]:

- 1) exteroceptors on the outer surfaces of the body, for sensing stimuli arising from the outside environment;
- 2) interoceptors on the body's inner surfaces, whose stimuli are primarily chemical (an example of such a stimulus would be "hunger pains"); and,
- 3) proprioceptors, which lie between the inner and outer surfaces of the body within the muscles and tendons, whose stimuli are mechanical forces and movements.

The following text concentrates on the proprioceptor; particularly as it deals with the muscle and has, primarily, motor functions. As shown in Figure 2.3, the proprioceptors within the muscle are the spindles and the Golgi tendon organs. Because the muscle spindle is ordinarily attached at both ends to the main muscle, it will experience the same relative length changes as the overall muscle. Therefore, when muscle is stretched, the spindle will be activated too. Then, information related to the length and the rate of change in length of the muscle are sent along the sensory nerves, called *Ia* afferents to the CNS. On the other hand, the Golgi tendon organ is in series with the force-producing contractile elements in muscle, so it plays a role in the perception of the force in the contracting muscle and behaves more like a force transducer for the muscle. Therefore, when the muscle contracts, information related to the force in the muscle is sent along the sensory nerves, called *Ib* afferents, to the CNS.

Figure 2.3 Muscle proprioceptors and their pathways.

(The term "*Ia*" provides two pieces of information; the "*I*" refers to the size of the sensory (afferent) fiber (Type *I* fibers are larger than Type *II*, and are our primary focus), and the "*a*" refers to the fact that this fiber comes from the spindle. The term "*Ib*" refers to the fact that the sensory fiber emerging from the Golgi tendon organ is a large Type *I* afferent, and the "*b*" refers to the fact that the fiber originates at the Golgi tendon organ.)

Figure 2.4 represents the “wiring diagram” of the neuro-muscular system [1]. As we mentioned earlier, the central nervous system (CNS) includes two parts: the spinal cord and the brain. The spinal cord, which is in the lower level of the CNS, is the major pathway for transmitting signals (information) from the muscle to the brain, and vice versa. In other words, the CNS uses the spinal cord as an interface between the brain and the muscle. The brain, the higher level of the CNS, is the centre of the control system and we will discuss how it functions in the next section.

As shown in Figure 2.4, a periphery nerve of the peripheral nervous system is used to collect the individual neurons (both sensory, or afferent, and motor, or efferent) which carry information toward and away from the spinal cord. Near the spinal cord, the peripheral nerve is separated into two roots: the dorsal (back) and the ventral (front) roots, where they contact the spinal cord separately. The afferent neurons which carry the information about the states of muscle enter the CNS via the dorsal roots. On the other hand, the efferent neurons which carry the CNS's motor command leave the cord via the ventral roots. Although the dorsal roots contain only afferents, and the ventral root only efferents, both afferents and efferents run together in a single peripheral nerve serving the muscle.

From the Figure 2.4, we also can see that there are two different colors of matter inside the spinal cord: grey and white. The grey matter, in the shape of a shield, is made up of nerve cells and their short interconnecting extensions. The white matter, surrounding the grey matter, is mainly composed of long nerve fibers running up and down the cord. Therefore, the main function of the white matter is to bring information up to the brain and motor commands down to the cord.

Figure 2.4 A simplified “wiring diagram” of the alpha and gamma motor system in the central nervous system. Although many of these circuit operate in parallel in the CNS, we only show one such circuit here.

Inside the spinal cord, the efferent motoneurons (alpha and gamma) are all found in a specific portion of the grey matter in the ventral roots. However, the afferent neurons can either connect with other neurons whose cell bodies are in the grey matter, or they can travel up to the brain through the surrounding white matter. As shown in Figure 2.4, the information from the *Ia* afferents is not only sent to the higher regions in the CNS but also to an interconnection with the alpha motoneurons in the spinal cord. The flow of the information from the *Ib* afferents is analogous to that from the *Ia* afferents, except it must pass through a relay neuron which is called an interneuron in the spinal cord. Therefore, the alpha motoneurons are innervated by two separate sources: the brain as well as the *Ia* and *Ib* afferents from muscles. After all, it is the alpha motoneurons that carry the efferent signals to the main muscle, causing it to contract. The interconnection of the afferent signals to the efferent motoneurons within the spinal cord allow for certain reflex actions in the biological system.

Figure 2.4 also shows gamma motoneurons as another efferent output. Gamma motoneurons are influenced by the brain and the gamma efferent connects directly to the muscle spindle. Therefore, the muscle spindle is not only affected by the length of the muscle but by the gamma system as well. That means the activity in the *Ia* afferent is determined by two things: the length and rate of stretch of the extrafusal muscle fibers as well as the amount of force supported in the intrafusal fibers, which is determined by the firing rate of the gamma motoneurons. However, as we mentioned before, the intrafusal fibers support relatively small amount of force in comparison with the extrafusal fibers.

Without going into any further unnecessary detail, we may summarize: the CNS receives information about the states of the muscle from the combination of two input sources: (1) the peripheral sensing of the muscle length and the rate of change of the length by the muscle spindle (*Ia*); (2) the peripheral sensing of the muscle force generated by the Golgi tendon organ (*Ib*). And there are two neural pathways that affect the muscle from the CNS: the alpha system and the gamma system. This information exchange makes up the sensory feedback branch of our control loop (refer to Figure 1.1), as well as

the communication link from the controller (brain) to the muscle system. It now remains to describe the controller portion of the neuro-muscular system.

2.4 MOTOR CONTROL SYSTEM: NEURAL CONTROL ASPECTS

Human motor response is based on the simultaneous cooperation of many anatomical parts (the muscles, the brain, the receptors that send information about the muscles, and so on). In this sense, it is useful to consider all of these parts as making up a system whose behaviour can be studied as a whole. Our attention will be focused on how the brain uses the input and output devices of the CNS, (muscle spindle, Golgi tendon organ, alpha motoneurons, gamma motoneurons) to control the muscle actuator.

Figure 2.5 Block diagram of neuro-muscular feedback control system [5].

The control flow of the neuro-muscular system is presented in Figure 2.5. Alpha and gamma systems are the final pathways from the CNS to the muscle and they adjust

the length, the velocity and the force of the muscle with the aid of the proprioceptors in the muscle.

As we know, there are two different kinds of movement involved in the human motor control system: reflex and voluntary. First, let us consider how the reflex function works. When the muscle is stretched by an outside agency, i.e. an external force, the spindle is stretched along in parallel with it. Then the information about the length and the rate of change in length of the stretched muscle is sensed by the sensory region of the spindle, the *Ia* afferent. Notice that the information from the *Ia* afferent travels to two places: the higher centre of the CNS, the brain, as well as to an interconnection with the alpha motoneurons. Therefore, when the *Ia* afferent feeds directly back to the alpha motoneurons, the firing rate of the alpha motoneurons will be increased to make the muscle that was stretched increase its contraction force and regain its original length. This is the so-called “stretch reflex”.

For voluntary movement, it is mainly the CNS that controls the activity of the alpha motoneurons. However, if the muscle movement were commanded only by a change in the alpha motoneurons activity, it would be difficult for the CNS to control the movement. Consider that, the activity of the alpha motoneurons can only cause the extrafusal muscle fibers to contract and the contraction of the muscle fibers will result in slackening the intrafusal fibers of the spindle. This is because the spindle is a biological component which does not have purely elastic properties. Therefore, the slackened spindle would make the *Ia* afferent (attached to the spindle) drop in its activity which, in turn, would cause the CNS control system to decrease the activity of the alpha motoneurons, and eventually, turn the muscle force off. However, this problem can be avoided by adding a process termed “alpha-gamma coactivation” in the CNS. As we mentioned before, the gamma motoneurons are also activated by the CNS and their activity can cause the intrafusal fibers of the spindle to contract. So, if the CNS can control both alpha and gamma motoneurons in a way of coordinating their action, a contraction of the extrafusal fibers will not be accompanied by a slackening of the

spindle. Instead, the contraction of the spindle through the alpha-gamma coactivation will maintain its sensitivity.

One function of the alpha and gamma command pair which was demonstrated above is for the voluntary movement event. It is also believed that these two command pathways play an even greater role in the neuro-muscular control system. But, as we know for biological systems, it is difficult to understand every detail since it is impossible to isolate the neural components without violating the integrity of the whole system. Although the full utility of the alpha and gamma systems is not yet well understood, it is recognized that the existence of these *two* distinct pathways is one of the critical features of the neuro-muscular control system. The alpha signal appears to play a major role in the global task control system (refer to Chapter 1); that is, the alpha commands are primarily responsible for the successful tracking of a desired path by the limb. On the other hand, the role of the gamma system is not so tractable, apart from the function of maintaining the spindle sensitivity. Since the observed feature of near-optimal performance of the neuro-muscular system is not encompassed by the alpha system, we believe that it may be the domain of the gamma system. That is, the gamma signal may play a role in the optimization of the muscle energy output in such a way to adjust the muscle effort in response to the (changing) demands of the environment. We will return back to this concept of the two distinct command systems when we develop our robotic control strategies in Chapter 5.

Our description of the neuro-muscular feedback control system, above, arms us with some knowledge to design our robotic control system. Even though, in Figure 2.5, we only present a simplified description of the neurological connection of the sensory receptors (spindles and Golgi tendon organs) to the CNS, we still can see that the neuro-muscular control system includes multiple control signals and feedback loops. And we also realize that the biological system is complex and some structural components and features are still not well understood. However, our goal, which will be addressed in Chapter 5, is how to use our present knowledge of these special organs to guide the

design of our robotic control system. At that time, we will attempt to assimilate the control characteristics of the neuro-muscular system as described above and base our robotic control system on the structure of the block diagram shown in Figure 2.5.

CHAPTER 3

SOME IMPORTANT TOPICS IN ROBOTIC CONTROL SYSTEMS: A LITERATURE SURVEY

3.1 INTRODUCTION

In the previous chapter, we discussed the structure and behaviour of the neuro-muscular system. We wish to apply this knowledge to the design of robotic systems. In order to do this successfully, it is first necessary to review some important fields of research in robotics. This will make us aware of the obstacles that may be encountered in robot design and will help guide our effort. In this chapter, we will present a brief overview of the literature that is devoted to some key topics in the field of robotics research.

A robot control system is the brain which drives the manipulator to complete its task. The ability of a robot to execute a task is largely dependent on the performance of its control system. Therefore, the design of the control system is a central issue in the development of robotic systems, and we will thus focus our effort in this direction.

Currently, depending upon the task performed by a robot system, the problem of control of robotic manipulators generally falls into two classes: *unconstrained* and *constrained*. In the first class, the manipulator is driven freely in its workspace and does not experience contact with the (hard) environment while it completes its task. Spray painting by a robot is an example of this class where the manipulator travels through its workspace without any significant restriction. On the other hand, for the second class, the manipulator is constrained by interaction with its environment while completing its

task. Typical examples are the “peg-in-hole insertion” task (e.g., inserting a computer board into a slot) and two robots manipulating a common object. These are tasks in which the manipulator is in contact with its environment and cannot move freely. The classification of manipulation as unconstrained and constrained is similar to the classification introduced by Whitney [44], who categorized manipulation into “rearrangement” and “force” tasks.

The first class of manipulation has been subject to extensive research and application over the past two decades [3, 4, 28, 49]. In general, these kinds of tasks can be specified by a preplanned motion trajectory of the manipulator which contains only position information. Therefore, existing position control strategies can be used successfully for manipulator control. The second class, which is equally important, has been studied by relatively few researchers. The number of applications and amount of experimental work carried out on constrained manipulation is correspondingly smaller.

This thesis deals with the second class of robotic manipulation in which the manipulator is constrained by some environmental contact. The range of available research topics within this area is wide and varied, including analyzing and designing interaction controllers, generating coupled stability criteria and so on.

3.2 CONSTRAINED MANIPULATION

In constrained manipulation, the interaction force results from contact between the manipulator and the environment. For example, robotic deburring is a task that requires control of the interaction forces. This contact force cannot necessarily be determined by the movement of the manipulator. Therefore, to accomplish a constrained task successfully, a designer must implement a strategy that controls the interaction force as well as the motion of the manipulator.

Previous researchers have suggested two approaches for controlling the constrained motion of a manipulator. The first approach is aimed at controlling the force and position of the manipulator in a nonconflicting way. This means that force is commanded along those directions constrained by the environment, and position is commanded along those directions in which the manipulator is free to move. The second approach is aimed at developing a relationship between the interaction forces and manipulator position. By controlling the manipulator position and specifying its relationship with the interaction force, a control system can allow the manipulator to complete the task while maintaining appropriate contact force.

The first approach was addressed by many researchers. In 1981, Mason [33] made the observation that for a general N -degree of freedom manipulator, each degree of freedom, (in the appropriate frame of reference), will be either position constrained or force constrained. Another important paper investigating the control of a robotic manipulator in contact with the environment was written by Raibert and Craig in 1981 [39]. In this paper, the authors proposed a control scheme termed “hybrid position / force control”. This control scheme was based upon the theoretical framework of Mason: the robotic manipulator’s workspace is divided into orthogonal directions based upon the environmental constraints. In those directions that are position constrained, the interaction force is controlled, and in those directions that are force constrained, the manipulator position is controlled. For example, in a surface grinding task, force would be controlled normal to the surface, and position would be controlled tangential to the surface. This control scheme requires a good working knowledge of the environment in order to associate precise directions with constraints and use the appropriate type of control for each degree of freedom. The need for detailed knowledge of the constraints is a serious limitation of this control scheme. For example, if the manipulator travels from one constrained point to another such that the environment at the new point exerts constraints that differ (particularly in direction) from the constraints at the previous point, a new control scheme with a different structure must be designed to accommodate the new constraints.

The second approach has been termed “impedance control”. It is a unified and general approach that focuses on the characterization and control of *interaction*. Impedance control was first proposed by Hogan [20]. The fundamental concept is based upon the observation that the environmental constraints impose a relationship between the position of the manipulator and the interaction force exerted by the environment. This control scheme does not directly specify force and position commands, but rather the dynamic relationship between the force and position. This gives rise to the concept of mechanical stiffness or “impedance”.

Hogan pointed out that this relationship may be modelled by a generalized nonlinear impedance consisting of inertial, damping and compliance characteristics. He then hypothesized that a feasible control strategy for a contact task could be to give the manipulator an impedance that is complimentary to that of the environment. The combined system of the manipulator and environment are predicted to be well behaved, in much the same manner as an electrical power source that is matched to the impedance of its load. Therefore, by controlling the manipulator position by specifying its relationship with the interaction force, we find that the interaction force between the manipulator and its environment can be controlled without the need for force feedback.

Recently, a number of investigators have addressed the control of the robotic mechanisms from a more general perspective. Dubrowsky and Desforges [12] have developed the model referenced adaptive control system technique in an effort to achieve quality performance in robotic manipulators even in the face of complex nonlinear dynamics. The most significant feature of their algorithm is that it does not require a-priori knowledge of the environment, nor a complex mathematical model of the system dynamics. However, the basic result of their work is the position control of single, open-chain manipulators.

Figure 3.1: Two manipulators handling a common object.

When two robots manipulate a single object, a closed kinematic chain is formed as illustrated by Figure 3.1. The control of the force distribution in a manipulation system that contains a closed kinematic chain is an important problem. The articulated links are individually powered by actuators, and each link and actuator comprises a subsystem. One of the most difficult problems involved in coordination of these subsystems is the necessity to actively control the force distribution within the mechanism. Failure to do so may result in the binding of the mechanism or a crushing load on a link as illustrated with Figure 3.2.

Figure 3.2: Binding of the manipulators (left) and crushing of the object (right).

The problem of force control in closed chain dynamic systems has been addressed by Hemami and Wyman [18], as well as by Orin and Oh [36]. Hemami and Wyman [18]

linearized their dynamic systems and used linear feedback to maintain the forces of constraint. The forces do not need to be measured nor be directly involved in the feedback loop. The latter authors proposed an approach based upon the minimization of the energy consumption and optimization of the load sharing between the several chains of the mechanism. This approach was applied to a nonlinear system where the forces were always specified a-priori.

Another important problem for the control of a constrained task is that, when the manipulator comes into contact with the environment, it will exhibit different dynamic characteristics. It is difficult, and often impossible, to maintain stable movement under all circumstances [9]. However, it is known that stability of the robot system is prerequisite for any robot control algorithm. Control creates a closed loop in which command gives rise to response, which in turn give rise to a new command. In such a situation, it is possible for instability to arise, especially for interaction tasks. Therefore, the design of a control system for constrained motion robots must address the stability issue.

3.3 CONTACT STABILITY

Currently, one of the significant issues in robot control is the stability analysis of the interaction between two manipulators or between a manipulator and its environment. The term “contact stability” or “coupled stability” refers to the stability of the robotic manipulator when it is in contact with some environment. Although contact stability is a critical issue for control of interaction, relatively little information or practical results can be found on this subject.

As we know, for a conventional control system, in order to track a trajectory (in force or position) more quickly, the closed loop bandwidth must be increased. To accomplish this, the control gains are generally raised. However, the gains can not be

increased beyond some limits, otherwise the system will usually become unstable. So, the application of high-bandwidth, high-accuracy feedback control can cause contact instability under the constrained condition. This is a major obstacle in the development of robot control systems.

The influence of the feedback gain on the stability is not as easy to quantify theoretically. The contact stability of a force controlled manipulator was analyzed by Whitney [45]. He modelled the manipulator as a velocity-input integrator, and used a proportional position, velocity, and force controller in discrete time. He modelled the environment as a spring and derived the following stability result:

$$0 < TG_F K_E < 1 \quad (3.1)$$

where T is the sample rate, G_F is the force feedback gain, and K_E is the environment stiffness. So, from his result, we can see that there is a tradeoff between G_F and K_E for a fixed T . In other words, a higher force feedback gain can be used without causing contact instability if the environment is more compliant (i.e., K_E is reduced). On the other hand, stability will be difficult to achieve if the environment is very rigid (i.e. $K_E \rightarrow \infty$).

Since Whitney published his result, various explanations for contact instability have been offered. Using one-dimensional linear analysis, Eppinger and Seering [13] concluded that a typical proportional force control system can become unstable when it is in contact with a rigid environment. An and Hollerbach [2] also realized this problem and proposed that in order to achieve steady-state accuracy without affecting the stability of a closed-loop system, a low-pass filter has to be added to the control architecture such that the system appears to be less “stiff”.

In another paper, Eppinger and Seering [14] have illustrated that stability problems can arise due to non-colocated sensors and actuators. They found that for the

case of colocation of sensors and actuators (which means actuators and sensors are attached to the same point on manipulator), it is always possible to design stabilizing controllers. But colocation of force sensors and actuators may not be physically possible or practical. Hence, non-colocation of sensors and actuators must be considered on a practical basis. The authors observed that for a system with non-colocated sensors and actuators, instability will usually occur under some given condition such that a universal controller design may not be simple. Mathematically, this is due partly to the unstable zeros inherent in non-colocated system and also because some eigenvalues of the non-colocated control system may move into the right-half complex plane. In 1992, Yang and Mote [48] introduced the use of specific time delay blocks in the control system to eliminate the instability of non-colocated system. The time delay constants in the blocks depend on the system eigenstructure, and so, are not general.

Kazerooni, Sheridan and Houpt [23] have demonstrated stability of target impedance dynamics that have been specified for a linear controller in contact with an environment. They treated the environment as a second order linear system to which the manipulator is directly coupled. However, the result was derived from local, linear analysis which required that the environment be either linear or at least differentiable. In contrast, the analysis presented by Hogan [20] required no such assumptions. In his studies, Hogan examined the impedance control scheme which preserves system stability in the face of interaction between the manipulator and its environment. He showed that contact stability may be guaranteed if the control system provides the manipulator with an appropriately structured dynamic response to the environmental input.

Another more general stability analysis for the system that models the manipulator and the environment as two coupled linear systems was discussed by Colgate and Hogan [11]. They restricted the environment to be a passive system (i.e., the environment is not allowed to contain any source of power) and defined the “driving point impedance” of the closed loop robotic system (while not in contact with the environment). They then used the Nyquist criteria to show that the combined system of

the manipulator and its environment will be stable provided that the driving point impedance presented to the environment is strictly positive real. Once again, this proof has the inherent limitation of assuming that contact interaction is linear in nature.

After some thought, it becomes apparent that the contact dynamics are not necessarily a linear form of interaction. Kazerooni [24] has developed a continuous time stability proof that makes no assumption about the linearity of the contact dynamics. He showed that there will be some upper limit on the size of the force feedback gain for which the stability of the system will no longer be guaranteed.

More recently, Colgate [10] generated the “coupled stability criteria” which are conditions that guarantee the stability of a manipulator in contact with (i.e., coupled to) any member of a well defined class of physical environments. This work stemmed from the previous results in which the class of the environments was assumed to comprise only of passive systems [11]. Recently, however, it has been argued that passivity is an unattainable or, perhaps even an undesirable goal in practice. This is because it is unrealistic to assume that all motion in an environment is damped. This has motivated the investigation of “nearly passive” robots. In their paper, Chapel and Su [7] have introduced a theoretical framework to examine the coupled stability of the nearly passive response. They defined a metric that bounds the difference between the achieved, non-passive response, and a passive target response. Conditions for coupled stability are developed based upon this measurement.

3.4 REDUNDANCY

Manipulators with more degrees of freedom than necessary to complete a specified class of tasks can be termed “redundant” for those tasks. When two manipulators (or one manipulator and its environment) interact in performing a constrained motion task, the problem of redundancy usually arises. These systems can

be regarded as redundant manipulator systems. Redundant manipulators have a number of potential advantages over non-redundant systems in many aspects, such as collision avoidance with obstacles in the work space, and avoiding manipulator singularities.

The redundancy of robotic manipulators plays an important role in increasing their flexibility, reliability, and versatility. However, because the interaction will make some joints and actuators redundant, complicated control algorithms are required. So, in order to take full advantage of the redundancy, more analysis must be done and effective control strategies should be developed. The study of redundant manipulators is an important research topic in advanced manipulation. The actuator forces in a redundant system cannot be found uniquely from the system's dynamic equations (i.e. there are more variables than equations). There are some common and important techniques developed to solve these problems. These techniques include the use of singular value decompositions, pseudoinverses, variational methods and Pontryagin's maximum principle.

Klein and Huang [27] have discussed the pseudoinverse control of a simple redundant manipulator. Their paper reviewed the pseudoinverse control method in terms of the singular value decomposition representation. The authors found that although the pseudoinverse allows the system to avoid singularities, it imposes limits on the configuration of the manipulator which may not be desirable.

Yoshikawa [50] has proposed a control scheme that utilizes redundancy in order to avoid singularities and obstacles. He proposed a quantitative measure of the ability of the redundant system to maneuver the manipulator. Using this measure, he discussed two different approaches for developing a control algorithm. The first approach makes use of an additional performance criterion, whereas the second approach specifies an additional desired trajectory of the manipulators.

In 1990, Kazerooni, Bouklas, and Guo [25] presented a control methodology for constrained motion in redundant robotic manipulators. This control approach actually takes advantage of the redundancy in the robot's degrees of freedom. The authors have assumed that the redundant manipulator has two mechanical parts. The first part is the primary manipulator used solely as a positioning system used to control the manipulator's position. The secondary part, which encompasses the remaining degrees of freedom is used to impose a desired force on the environment. The authors verified the applicability of this control method, analyzed the contact stability of the system, and derived a sufficient condition for stability.

Since the redundant system is powered by more actuators than there are degrees of freedom in the system, the equation matrix that represent the inverse dynamics is underdetermined. Therefore, there is no unique solution. In an effort to address this difficulty, Nahon and Angeles [35] have investigated two approaches: one for minimizing the power losses in the system, and another for minimizing internal forces in the system. They used a linear quadratic optimization criteria to form a complete system of equations. They also introduced local and global performance indices to compare the results obtained from these two approaches to decide which one yielded better results under certain situations.

3.5 COMPARISON BETWEEN ROBOTIC AND HUMAN MANIPULATION

In comparison with robots, it is widely recognized that humans have much superior capability when controlling the execution of a task. The human arm is a superb manipulator, capable of interacting with a diverse set of environments in a well behaved manner. In particular, it has been observed that whereas robotic systems that interact with objects may exhibit unstable behaviour, the human manipulator shows no such difficulty (if we do not consider the case of overloading in this sense). This may suggest that the structure of the biological actuator and the biological control strategy can

guarantee the stable completion of a contact task. Therefore, a better understanding of the human manipulator system is valuable to the goal of expanding the capability of robotic manipulators.

Analyses of the control of human arms in contact with an environment have rarely been presented in the past. In 1978, Chao and An [6] discussed the human musculoskeletal system. They pointed out that the problem of redundancy (in which the unknown variables inherent to the system exceed the available equations of equilibrium and constraint) is also encountered in the human manipulator. They chose the pinch function of the thumb and forefinger as an example in order to formulate a general theory for the solution of the redundancy problem. They did not, however, consider the neurological factors which may, in fact, be dominant in the musculoskeletal system. That is to say that the human control strategy may be of such sophistication to allow it to adapt to and control just about any actuator successfully.

A skeletal muscle is the actuator which drives the human arm to move. How the muscle is controlled by the central nervous system to produce movement is a fundamental question of neurophysiological research. It is also fundamental to this work. Hogan [21] has proposed the impedance control scheme which was developed from an investigation of the strategies that may be used by the human brain to control the arms. The approach observes the similarity between complex biological systems and robotic systems with respect to their physical function and interaction with the environment. He also explored the use of redundancies in the human neuro-muscular system: “excess” actuators or “excess” skeletal degrees of freedoms, and suggested that the redundancies in the neuro-muscular system may in fact play an essential functional role in controlling interaction behavior.

Hogan first defined the mechanical impedance of the neuro-muscular system as the relationship between muscle force and neural inputs. He then pointed out that the adaptive control of the mechanical impedance may be one of the ways in which the

central nervous system controls the musculoskeletal system. He also suggested that the central nervous system can preserve the muscle's simple spring-like behaviour during a contact task.

Inspired by Hogan's concept of impedance control, Wu, Young and Houk [46] have studied the unique nonlinear damping feature of the neuro-muscular system. They utilized the phase plane approach to evaluate the characteristics of this nonlinear damping and noted that the nonlinear dynamics may assist the human arm to adapt to different tasks without the stability problems which are typical of linear damping systems. Then, they suggested that a suitable model of the neuro-muscular system would be comprised of a linear spring in series with a nonlinear damping element. However, a thorough understanding of the motor signals in the neuro-muscular system is crucial to the effective use of this sort of model.

In their later research, Wu, Young and Hwang [47] analyzed the motor signal patterns. Based upon the identified signal patterns, they proposed the strategy of "muscle modulation control". This control strategy may then be applied to the neuro-muscular model of Wu et. al. [46]. Their results demonstrate that the neuro-muscular model can adapt to the transition from position control to force control without a stability problem. This capability is very desirable for a robotic system.

Some other similarities in the control behaviour of the human and robotic systems have also been researched. In their paper, Iqbal, Dogruel, and Hemami [22] have briefly discussed the stability problems that may occur in both musculoskeletal and robotic systems due to the presence of feedback delays. These feedback delays have been known to destabilize control systems as a result of the inherent addition of phase lag. A feedback delay is usually a consequence of neural transmission times in the biological system, and of computation time in the robotic system. The authors investigated the stability by the polynomial methods of Nyquist and Popov. However, the analysis

applies only to a linearized system, whereas an analysis based on nonlinear systems theory does not yet exist.

A further example of the research into human-robot similarities is the work of Kazerooni and Guo [26]. These authors have studied a new class of manipulators called “extenders” which are human-controlled robotic manipulators. These kind of manipulators benefit from both the strength advantage of mechanical systems and the neurological advantage of the human system. A robotic manipulator has difficulty in achieving both the speed and accuracy of the human limb, but it may readily exceed the limited strength of the biological actuator. So, the extenders can amplify the strength of the human, while utilizing the intelligence (i.e., the control strategies) of human to generate the command signal to the system. The authors analyzed the stability of the system that is comprised of human, extender, and manipulated object and found that the better the required performance, the narrower the stability range is.

From our discussion above, we observe that the field of robotics may benefit from the analysis of the human neuro-muscular system. We see that the human manipulator is a physically complex manipulator that embraces the advantages of redundancy, and yet the control system manages to avoid contact instability. A thorough understanding of the topics of redundancy, stability and biological neural-motor control strategies may assist in the advancement of the state of the art of robot design.

CHAPTER 4

A MECHANICAL SYSTEM MODEL FOR THE ROBOTIC CONTROL SYSTEM

4.1 INTRODUCTION

In comparison with robots, it is widely acknowledged that the human being has capacity to perform variable, fast, and delicate tasks in a manner which far exceeds robotic ability. Considering an assembly process operated by a human worker, we can see that this worker can pick up the parts and assemble them together with an apparently seamless and continuous series of motions. This ability suggests that a versatile control mechanism exists within the human being. This mechanism also maintains stability during constrained manipulation tasks, as well as in high speed tasks, and in interactions with unknown environments. Complexities in the design of both the physical components and the control systems have prevented the ultimate goal of matching human skill in robotic systems. Specifically, with the existing state of technology, we do not have a robot that can work as successfully as the human being in an unstructured environment.

A major challenge facing the development of sophisticated robotic control systems is to capture the superior function demonstrated by human beings. Our long range goal is to develop a control algorithm that emulates the biological approach. The superior performance may be due in part to the human sensors as well as control strategies generated from the brain. However, we must recognize that the human being also has a unique neuro-muscular actuation system that responds to the command signals

from the CNS. This physical system is quite different in many respects from typical robotic actuators. Today's robotic actuators do not have the same capabilities of muscle.

The study of the human neuro-muscular system can be frustrating due to the intricate complexity of the system. One potentially powerful avenue for gaining insight into such a complex system is to develop an appropriate model of the system. In this introductory work, we concern ourselves only with the control system interacting with an actuator which has basic muscle-tendon features. Our model should include two aspects of the neuro-muscular system: the muscle actuation system and neural control system.

It is commonly assumed [41] that, for a robotic system, the control system design process is independent of the mechanical design process, and usually follows it. That means the mechanical system usually already exists when the design of the controller is attempted. Hence, the controlled system's performance is strongly dependent upon the mechanical system (i.e., actuator and manipulator) dynamics. In our case, we are simultaneously proposing an actuator and controller in the sense that our actuator model will be kept general when we develop the controller. Therefore, one initial and critical job is to describe the actuator (i.e., the muscle and tendon) with an appropriate model. We also consider a rigid connection between the actuator and the environment so that there is no distinct manipulator component. In this way, we consider a one-dimensional mechanical model with no loss of generality. With this, we may avoid unnecessarily complicated geometries, but still capture the lessons to be learned from the neuro-muscular system.

In this chapter, we will study the mechanical part (i.e., the actuator) of the neuro-muscular system. Our effort will be directed at the analysis of several different models of the muscle-tendon structure. After comparing these models, we will propose a simple muscle-like robot actuator model for the control system which will be developed in next chapter. On the other hand, in order to develop and analyze our control scheme, we need to define a system mechanically comprised of both actuator *and environment*. So, in this

chapter, we will also define a general environment which couples to the actuator, thereby building a whole basic mechanical system model for the controller.

4.2 MUSCLE MODELS

One of the major problems in modelling the human neuro-muscular system is to describe the actuator, i.e., the muscle and the tendon, with an appropriate model. During the past century, numerous investigations have sought to determine the structure and function of skeletal muscle and many models of muscle have been suggested. The most famous muscle model was developed by Hill in 1938 [19], and this model has dominated the field for years. Although some elegant muscle models hold promise for much more accurately describing many of the recognized characteristics of muscle, they are generally too intricate for our purpose.

This section deals with models which physically and mathematically represent the relevant human muscle characteristics. Our goal is to develop a simple model that adequately represents the observed phenomenological behavior of the muscle. However, we can anticipate that a model which is too simple will reveal little insight into the system or its performance, and may even generate misinformation. In order to model the human muscle successfully, we must understand explicitly which characteristics are relevant and which may be neglected or simplified. For example, we know that the behaviour of muscle is nonlinear. Since the form of the nonlinearity is not well known, the inclusion of some arbitrary nonlinearity would not contribute much to our understanding of the muscular system. On the other hand, we are aware that the principal phenomenological features may be described approximately by a linear model under some conditions [8]. It is useful to neglect the nonlinearity of the muscle behaviour initially in the analysis and, in future work, seek errors induced by this assumption. Therefore, we choose to use a linear model in this work since the complexities added by non-linear features will not necessarily result in a significant or valid gain of insight.

Before we propose a model of a muscle system, it is important to capture the physical behaviour of the muscle with experimental data. We may then develop our model with a focus on how well it represents the physical data. As such, we will only address the models which are based on Hill's model because he derived it from the famous quick-release experiments on muscle [34]. That is to say that the structure of Hill's model is well-founded in its representation of real muscle behaviour and we may proceed to use the models which are based upon Hill's model without having to reproduce the experiments on real muscle. We first introduce Hill's model to be our baseline or starting point.

Figure 4.1 Hill's functional model of the muscle. This three-element model includes the contractile element (CE), the series elastic element (SE), and the parallel elastic element (PE) which bears resting tension. N is the neural input.

Hill's Model

This is the first and simplest muscle model that can adequately simulate the principal muscle characteristics. As shown in Figure 4.1, an active muscle is represented by three elements. Two elements are arranged in series: a contractile element and an elastic element. The contractile element (CE) is capable of generating a force when the muscle is activated by the neural input (N). The tendon along with the connective tissues that are in series with the contractile component are considered collectively to be the series elastic element (SE) which transfers the force generated by the contractile element.

On the other hand, if an inactivated muscle is stretched, tension is observed which is due to the connective tissues that surround the muscle. Therefore, the parallel elastic element (PE) is added to the model to account for the elasticity of the muscle at rest.

Although Hill's model may be simplified by assuming that the series elastic element has infinite stiffness (i.e., zero compliance) or the parallel elasticity approaches zero stiffness, we observe that such simplifications are seldom sufficiently realistic. Hence, Hill's model may be regarded as the simplest representation of the muscle structure. For nearly 60 years, since 1938, many ideas have been added to Hill's model and modifications have been proposed in order to accommodate newly discovered facts. In some models, the series and parallel elastic elements are given viscoelastic behaviour. In another, the elements are arranged differently, such as one elastic element in parallel with the contractile element, and another in series with the combination. These features are incorporated to more realistically represent the muscle behaviour and structure. For our work we wish to include some improvements to Hill's model, while at the same time, maintain a degree of simplicity. Given this, we are now prepared to propose some models for this work. Our first model, which we term "**Model 1**" incorporates the former idea of viscoelastic behaviour. Our second model, "**Model 2**" also incorporates the viscoelastic behaviour as well as a relocation of the elastic tendon structure. Let's first begin with a complete description of our **Model 1**.

Model 1

As shown in Figure 4.2, this is a four element model of the muscle. The elements FG, SE, PE and PD are, respectively, the force generator, the series elastic element, the parallel elastic element and the parallel damping element. Compared to Hill's model, we can see that the complexity has been increased by adding the damping element (PD). The contractile component of a real muscle is known to demonstrate some relationship between force and velocity (i.e., the muscle force depends upon the velocity of contraction) under dynamic conditions. This relationship has been approximated in the linear muscle model and is achieved by a parallel combination of a force generator and a

viscous damper. So, adding the damping element (PD) accounts for the velocity dependent behaviour of the muscle.

Our force generator (FG) is also more versatile than the muscular contractile element (CE), in that we allow it to give rise to extensile forces as well as contractile forces. The CE component of muscle does not have this ability for the simple reason that: “It is impossible to push on a rope.” In muscular physiology, the bi-directional motion of limbs is achieved through a pair of complementary, opposed muscles (“antagonist muscles”), each capable of generating contractile forces only. In a system of mechanics, we can easily realize a bi-directional force actuator (with, for example, a two port hydraulic cylinder), and therefore, we assign this ability to our FG component. With this, our model is fully functional, and yet it maintains a practical simplicity.

Figure 4.2 Muscle Model 1. This model consists of the force generator (FG), and a parallel damping element (PD) in series with an elastic element (SE). This series combination is in parallel with another elastic element (PE). The parameters representing the damping and elastic elements in their linear constitutive relations are identified by B_{p1} , K_{s1} , and K_{p1} respectively. The subscripts “P” and “S” refer to parallel and series respectively, whereas the subscript “1” identifies **Model 1**.

Model 2

Compared to **Model 1**, this model considers the muscle and tendon separately. It assumes that the whole muscle is a single force generator (i.e., contractile element) in parallel with a damping element and an elastic element. These parallel elastic and

damping elements reflect the “damped spring” characteristics of the muscle. The tendon, which lies in series with the muscle, is represented by an elastic element in order to model its force transducer behaviour. Therefore, it also bears resting tension, as well as the parallel elastic element (PE). Figure 4.3 shows a diagrammatic representation of the model. The reason for modelling the muscle and tendon separately is that, from the muscle’s anatomical structure, the tendon is in series with the whole muscle, not just the contractile element. The muscle connective tissues, which are in series with the contractile element as in **Model 1**, also contributes to the muscle elasticity, but their influence is small and can often be neglected. That is to say that it may not be necessary to include an elastic element that is strictly in series with the FG and PD parallel combination. Therefore, for **Model 2**, we choose to ignore this relative small connective tissue compliance, which allows us to maintain the order of **Model 1** in **Model 2**.

Figure 4.3 Muscle Model 2. This model consists of a force generator (FG), a parallel damping element (PD) and, elastic element (PE) in series with an elastic element (SE). In the same manner as Figure 4.2, the constitutive relation parameters are identified as B_{p2} , K_{p2} and K_{s2} .

Some evidence has verified that **Model 2** is actually equivalent to **Model 1** in its representation of muscle behaviour. In 1971, Fung [17] has shown that these models

are, in fact, mathematically equivalent if we choose the parameters of the linear passive elements to satisfy the following relations:

$$K_{P1} + K_{S1} = K_{S2} \quad (4.1)$$

$$K_{P1} = \frac{K_{P2}K_{S2}}{K_{P2} + K_{S2}} \quad (4.2)$$

$$\frac{B_{P1}}{K_{S1}} = \frac{B_{P2}}{K_{P2} + K_{S2}} \quad (4.3)$$

If the above relations are met, then both models are exactly equivalent. That means that if one model is known, the other equivalent model can replace it and will have the same dynamic characteristics. We will verify this statement in section 4.6. Now, we will introduce a third model.

Although we wish to maintain simplicity in our actuator model, we must take account of all significant physical features of a real actuator. As such, we also recognize that any real actuator has mass, and we should consider including this mass in our model. In the biological actuator, the muscle fibres comprise the bulk of the mass, with the tendon adding a small amount of mass. In reality, the mass of a robotic actuator (for example, a hydraulic cylinder) can be insignificant compared to the load mass. Nevertheless, we shall examine the effects of including the actuator mass (i.e., muscle mass) in each of our **Models 1 and 2**. This gives rise to our **Models 3 and 4** as illustrated with Figure 4.4 and 4.5 respectively.

We define our elastic elements to be linear (i.e., Hookean) mechanical springs, and our viscous elements to be linear dashpot dampers in the usual sense (i.e., force proportional to extension rate) for both models. The springs and dampers represent the passive properties of the muscle. Despite the fact that none of these assumptions of linearity agrees particularly well with physiological reality, the models are treated as linear here because the nonlinear muscular system can be always linearized [8]. The

simple linear models often capture the essential behaviour and will be used in the interest of simplicity. For all of the models, the active force generator (i.e., the contractile element) is known to be affected by the extension of the muscle, the extension rate (i.e., velocity) of the muscle as well as the degree of excitation (which is under neural control). The workings of the force generator will not be discussed in this chapter, but will be described in detail in next chapter. For analysis purposes in this chapter, the role of the FG is to allow us to enter an arbitrary force input to the system.

Figure 4.4 Muscle Model 3. This model consists of a force generator (FG) in parallel with a damping element (PD) connected to an element representing the muscle tissue mass (M). This combination is in series and in parallel with elastic elements (SE) and (PE) respectively.

Figure 4.5 Muscle Model 4. This model consists a parallel structure of a force generator (FG), a damper (PD) and an elastic element (PE), all connected to an element representing the muscle tissue mass (M). This group is in series with an elastic element (SE) representing the compliant tendon.

4.3 ENVIRONMENT STRUCTURE

Now that we have described our actuator models, we need to couple them to an environment to form a complete mechanical system. Only with this complete system may we proceed to do a mathematical analysis. In this section, we focus on the model of the environment. There is no specific model for the environment dynamics: we require that the environment model be general to ensure that our analysis of the muscle and control system is not limited in application. Once we recognize that a general environment is simply a mechanical system, we can readily surmise that it should contain all of the basic features of mechanics. This implies that the model should have inertia (i.e., mass), and compliance (i.e., spring-like features), as well as an energy dissipating feature (i.e., a damper). We wish to maintain simplicity, so we choose linear time-invariant features. This allows us to neglect non-linear mechanical behaviour such as hysteresis. We need not be disturbed by this since a great many practical environments may be represented by a subset of the mass-spring-damper combination. Our environment model may be depicted by Figure 4.6.

Figure 4.6 The environment is represented by a mass (M_E) which is supported by a spring (K_E) and a damper (B_E) to the ground.

4.4 MODEL PARAMETERS IDENTIFICATION

In order to do a numerical analysis, we must attach values to our system parameters. Normally, we might expect to match the values of the actuator parameters to those of experimental results on human muscle. However, we recognize that a robotic actuator may have practical stiffness, mass and damping values that are much different from human muscular features. Therefore, it is not necessary to match exactly our actuator parameters to those of a human muscle model. What we should attempt to do is to make the relative values of the parameters similar to human muscular features. For example, experimental evidence of muscle [42] has shown that the tendon stiffness is somewhat larger than the muscle tissue stiffness, that is, $K_s > K_p$. Therefore, if we have an actuator with an internal stiffness $K_p = 10 \text{ N/m}$, then we might expect the series stiffness, K_s , to be, say, 3 times larger.

In the similar sense, it is not critical how we choose the parameters values for the environment structure. This is because we can conceive of any number of different practical environments. We do, however, recognize that the environment mass is typically much larger than the actuator mass (in fact, in our **Models 1 and 2**, we neglect the actuator mass). Hence, we might practically assign $M_e = 10 \times M = 10 \text{ kg}$. The environmental stiffness would be large for many constrained manipulation tasks (e.g., pole-in-hole insertion for which the environment is “hard”) but may conceivably be small for other tasks (e.g., for a door opening task for which the environment is “soft”). For purpose of investigating contact stability, we would choose a large value of the environment stiffness K_e (say $5 \times K_p$). The environmental damping may also shows a wide range of practical values. To limit our scope, we arbitrarily choose the damping of the environment to be 10 Ns/m and we chose the damping of the actuator to be comparable in magnitude, say, 6 Ns/m .

Although, we may spend much effort in investigating various values of our parameters, we need only choose one set as a reference. We will use this set of values of parameters as a basis for all of our subsequent analysis. For convenience, we list our chosen values that are eventually used in our simulations in Table 4.1.

Table 4.1 This table shows the final selection of values for the parameters of our model. The values are chosen somewhat arbitrarily but with some reflection on reality. The third column indicates a set of mathematically consistent units that may be associated with the parameters.

Parameter	Value	Unit
K_s	30	N/m
K_p	10	N/m
B_p	6	Ns/m
M	1	kg
M_E	10	kg
B_E	10	Ns/m
K_E	50	N/m

4.5 ANALYSIS OF THE OPEN LOOP MUSCLE-ENVIRONMENT MODELS

We have specified the structure of our actuator and environment models, along with nominal values for the parameters, and now we may proceed to analyze the four different muscle (i.e., actuator) models. In this way, we can gain an understanding of the dynamic behaviour of the basic muscle structure, and observe the difference between our model structures. From these we will choose one single muscle model structure as our actuator for use in the development of our control system in Chapter 5. We recognize

that the response of our actuator model should also incorporate phenomenological muscle behaviour. We will attempt to identify these features in our analysis.

It is helpful to outline the steps that we will follow in the analysis of the models. In this way, we maintain an orderly flow to our analysis, and we will be able to clearly identify the results. Let us list our procedure as follows:

1)

Write the differential equations of motion that describe the mechanical system represented by our model. We will present the equations in state space form:

$$\dot{x} = Ax + Bu \quad (4.4)$$

$$y = Cx + Du \quad (4.5)$$

where the vector x represents the states of the system, vector u is the input, y is the output vector and A , B , C , D are coefficient matrices.

2)

From the coefficient matrix A , we determine the eigenvalues of the system. The eigenvalues, of course, identify the modes (i.e., the poles) of the dynamic system in the s -plane. As shown in Figure 4.7, the position of these poles in the s -plane can provide the following information:

- i. The open loop system is stable if all eigenvalues lie within the left half plane.
- ii. The speed of response of the system is dominated by the poles that are closest to the imaginary axis (IM) (e.g., in the case of Figure 4.7, the dominant poles are the complex pair). The wider the separation from the Im axis (i.e., the greater σ or $\zeta\omega_n$ is), the quicker will be the response associated with that pole or pole pair. The system response is dominated by the slowest component response.

- iii. The damping ratio (ζ) of each pair of complex-conjugate poles is readily derived from the eigenvalues. The damping ratio influences the response time, the overshoot and the amount of oscillation in the response. For very low damping the response is oscillatory and has much overshoot, while for large damping (i.e., ζ greater than 0.707) the response shows almost no overshoot and oscillation.

Figure 4.7 An s-plane plot for a general pair of complex conjugate poles and a single pole on the real axis (RE). The damping ratio (ζ) is equal to $\sin \theta$, and ω_n is called the undamped natural frequency of the second order component of this system (i.e., the pair of imaginary space poles). The parameter, σ , is the reciprocal of the time constant, τ , of the first order component of this system (i.e., the single real pole).

3)

The system is then analyzed for controllability and observability.

Controllability of the system is required since we intend to develop a controller for our system which uses state feedback, albeit in a novel manner. The criteria for controllability is stated as [15]:

$$\det[B \ AB \ A^2B \ \dots \ A^{n-1}B] \neq 0 \quad (4.6)$$

where n is the rank of the A matrix ($A = n \times n$ matrix). That means the rank of the controllability matrix $[B \ AB \ A^2B \ \dots \ A^{n-1}B]$ must be equal to n , the order of the system.

Observability of the system is necessary because we wish to deduce information about the states of the system through an observer system, and then provide this information to the control system. The criteria for observability is stated as [15]:

$$\det \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \neq 0 \quad [4.7]$$

4)

In order to simulate a response from our system, it is necessary to define an input to the system. As described in section 4.2, the input is provided through the force generator component (FG) of our system model. And, for analysis of a open loop system which has complex poles, it is most informative to present the step response, so, we simply define the input to be a unit step (force) input.

5)

The step response of the open loop system is simulated using Matlab. We study the step response to ensure that the behaviour of the system will correspond with our predictions made when investigating the eigenvalues. We also attempt to recognize the phenomenological muscle behaviour within our actuator model response.

Now, let us proceed with the analysis of each of our actuator-environment models.

Model 1

Let us first describe the complete model explicitly with Figure 4.8. We can see that the motion of the system is considered in one-dimension only. States x_1 and x_2 are defined on the figure. The position, x_2 , of the environment mass is the state of ultimate interest as it will be required to track a given path. And we may use the figure to assist in directly writing down the state space equations of motion.

Figure 4.8 This figure illustrates the total mechanical system as a combination of **Model 1** (see Figure 4.2) and our environment structure (see Figure 4.6).

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -K_S/B_P & K_S/B_P & 0 \\ 0 & 0 & 1 \\ K_S/M_E & -(K_S + K_E + K_P)/M_E & -B_E/M_E \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1/B_P \\ 0 \\ 0 \end{bmatrix} \cdot [F] \quad (4.8)$$

$$y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (4.9)$$

where x_1 and x_2 are the position state variables as shown in Figure 4.8 and x_3 is the velocity state variable for x_2 (i.e., $x_3 = \dot{x}_2$). F is our arbitrary force input signal from the FG. Output, y , is seen to be the position of the environment mass (i.e., x_2).

After inserting the appropriate values from Table 4.1 into the A matrix, we may obtain the system eigenvalues using the Matlab instruction “eig(A)”. For this case we find the eigenvalues to be:

$$\lambda_{1,2} = -0.8164 \pm 2.4906i \text{ (due to environment)}$$

$$\lambda_3 = -4.3672 \text{ (due to actuator)}$$

Immediately, we can state that the open loop system is stable since all of these eigenvalues are in the left half plane. We see that the system is dominated by the complex poles. That is to say that the response is expected to be primarily second order with a slight influence of the first order component. The damping ratio of the complex pair of eigenvalues is readily calculated by:

$$\zeta = \frac{0.8164}{\sqrt{0.8164^2 + 2.4906^2}} = 0.3115$$

And the undamped natural frequency is calculated by:

$$\omega_n = \frac{0.8164}{0.3115} = 2.621 \text{ s}^{-1}$$

Since these poles have a damping ratio smaller than 0.707, we anticipate a system response with some overshoot and oscillation.

Our next step is to examine the controllability of this system. We implement our criteria in Matlab and find that:

$$\det[B \ AB \ A^2B] = -0.0417$$

From this, our criteria states that controllability is assured since the determinant is nonzero. That means the system is completely state controllable. We then proceed to examine the observability, and find that:

$$\det \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} = 3$$

So, we conclude that the system is observable since the determinant above also has a nonzero value. This means that if some of the state variables are not directly measured (in this case, only x_2 is directly measured with a position sensor), we may use an observer to estimate the others for our control system.

Finally, we simulate the step response which is illustrated in Figure 4.9. We see from this figure that indeed the response has a dominant second order shape with some overshoot and oscillation. In order to more clearly illustrate the procedure in our analysis we present a simple Matlab instruction set (Appendix A) that contains all of our analysis steps. Our simulation was executed with the Matlab function “step”, with a specified time step of 0.05 sec..

It still remains to suggest that the step response of our actuator system is representative of the phenomenological behaviour of muscular systems. It is not necessary to go into excessive detail on this issue especially since the coupled environment influences the system response and we cannot isolate strictly the muscle actuator model itself. Also, we note that phenomenological muscle behaviour is practically influenced by neural control signals, and this prevents us from observing the free response of a real muscle-environment system. However, there is evidence that our model response may well represent the response of a real muscle-environment system that lacks the benefit of control signals from the brain. Specifically, a person who suffers from a neural disorder may have difficulty in limiting the overshoot (i.e., flailing) and

Figure 4.9 Step response of Model 1.

oscillation (i.e., tremoring) of his limbs when attempting to perform a task. Therefore, we may simply assert that a second order, asymptotic, response is consistent with the expected open loop behaviour of muscle.

We recognize that the single pole on the real axis represents the natural mode of the actuator. The complex pair of poles represent the natural modes of the environment. We see that the actuator system pole is widely separated from the origin and this satisfies the requirement that the actuator be fast enough to adapt to the control command with no lag. The slower environment structure dominates the response with its second order behaviour.

Model 2

The analysis of our next model follows in the same manner as **Model 1**. For brevity we will just state the key results without description.

Model:

Figure 4.10 This figure illustrates the total mechanical system as a combination of **Model 2** (see Figure 4.3) and our environment structure (see Figure 4.6).

Equations:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -(K_S + K_P)/B_P & K_S/B_P & 0 \\ 0 & 0 & 1 \\ K_S/M_E & -(K_S + K_E)/M_E & -B_E/M_E \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1/B_P \\ 0 \\ 0 \end{bmatrix} \cdot [F] \quad (4.10)$$

$$y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (4.11)$$

Eigenvalues:

$$\lambda_{1,2} = -0.6810 \pm 2.3699i \text{ (due to environment)}$$

$$\lambda_3 = -6.3047 \text{ (due to actuator)}$$

Damping ratio and undamped natural frequency:

$$\zeta = 0.2762$$

$$\omega_n = 2.466 \text{ s}^{-1}$$

Controllability criteria:

$$\det[B \ AB \ A^2B] = -0.0417 \neq 0$$

Observability criteria:

$$\det \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} = 3 \neq 0$$

Step response: (see Figure 4.11)

Figure 4.11 Comparison of step response of Model 2 (solid line) with Model 1 (dashed line).

Again we present a listing of our Matlab command lines used to generate the above results (Appendix A). Our results show that the behaviour of this system is very much similar to that of **Model 1**, as evidenced by the similar undamped natural frequency and small influence of the first order pole. On the other hand, our **Model 2** has a somewhat smaller damping ratio than our **Model 1**, and hence, the step response exhibits relatively more overshoot and oscillation. Since, the response is otherwise similar to that of **Model 1**, we need not describe it any further.

Model 3

Model:

Figure 4.12 This figure illustrates the total mechanical system as a combination of **Model 3** (see Figure 4.4) and our environment structure (see Figure 4.6).

Equations:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -K_s/M & K_s/M & -B_p/M & 0 \\ K_s/M_E & -(K_s + K_p + K_E)/M_E & 0 & -B_E/M_E \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/M \\ 0 \end{bmatrix} \cdot [F] \quad (4.12)$$

$$y = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (4.13)$$

where x_1 and x_2 are the position state variables as shown in Figure 4.12, x_3 and x_4 are the velocity state variables for x_1 and x_2 respectively (i.e., $x_3 = \dot{x}_1$ and $x_4 = \dot{x}_2$). F is our arbitrary force input signal from the FG. Output, y , is seen to be position of the environment mass (i.e., x_2).

Eigenvalues:

$$\lambda_{1,2} = -0.8789 \pm 2.2982i \text{ (due to environment)}$$

$$\lambda_{3,4} = -2.6211 \pm 4.7813i \text{ (due to actuator and its mechanical inertia)}$$

Damping ratio and undamped natural frequency:

$$\zeta = 0.3572 \quad \omega_n = 2.4605s^{-1} \text{ (calculated from } \lambda_{1,2} \text{)}$$

$$\zeta = 0.4807 \quad \omega_n = 5.4526s^{-1} \text{ (calculated from } \lambda_{3,4} \text{)}$$

Controllability criteria:

$$\det \begin{bmatrix} B & AB & A^2B & A^3B \end{bmatrix} = -9 \neq 0$$

Observability criteria:

$$\det \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} = -9 \neq 0$$

Step response: (see Figure 4.13)

Figure 4.13 Comparison of step response of **Model 3** (solid line) with **Model 1** (dashed line).

We see again that the step response is similar to that of **Model 1**. This is not surprising since the dominant pair of poles have similar characteristics (i.e., ζ and ω_n) as those of **Model 1**. However, we see that the overshoot of our **Model 3** is increased by the presence of the second pair of poles. This influence is only seen in the early part of the response because the relative damping (i.e., $\zeta\omega_n$) of this second pair of poles is greater than that of the dominant pair. Ultimately, we observe that the addition of the actuator model element (M) representing the muscle tissue mass has only a very small influence upon the behaviour of the system.

Model 4

Model:

Figure 4.14 This figure illustrates the total mechanical system as a combination of **Model 4** (see Figure 4.5) and our environment structure (see Figure 4.6).

Equations:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -(K_s + K_p)/M & K_s/M & -B_p/M & 0 \\ K_s/M_E & -(K_s + K_E)/M_E & 0 & -B_E/M_E \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/M \\ 0 \end{bmatrix} \cdot [F] \quad (4.14)$$

$$y = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (4.15)$$

Eigenvalues:

$$\lambda_{1,2} = -0.6923 \pm 2.2787i \text{ (due to environment)}$$

$$\lambda_{3,4} = -2.8077 \pm 5.7150i \text{ (due to actuator and its mechanical inertia)}$$

Damping ratio and undamped natural frequency:

$$\zeta = 0.2907 \quad \omega_n = 2.3815 \text{ s}^{-1} \text{ (calculated from } \lambda_{1,2} \text{)}$$

$$\zeta = 0.4409 \quad \omega_n = 6.3674 \text{ s}^{-1} \text{ (calculated from } \lambda_{3,4} \text{)}$$

Controllability criteria:

$$\det \begin{bmatrix} B & AB & A^2B & A^3B \end{bmatrix} = -9 \neq 0$$

Observability criteria:

$$\det \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} = -9 \neq 0$$

Step response: (see Figure 4.15)

We see from the step response of our **Model 4**, that the comments that were expressed regarding our **Model 3** also apply in this case.

(Refer to Appendix A for a listing of the Matlab commands for **Model 3** and **Model 4**.)

Figure 4.15 Comparison of step response of Model 4 (solid line) with Model 2 (dashed line).

4.6 CONCLUSIONS

At last we must choose a single model for usage while developing our controller. We have seen from the above results that the behaviour of all four models are very similar. And, the results of the simulation of the actuator-environment systems indicate that the muscle-like actuator system is capable of generating stable movement. Therefore we may arbitrarily choose any one of the models for development of our control system.

Immediately, we observe that **Models 3 and 4** each have the complexity of an additional pole, and yet this imposes only a minor influence on the system dynamics. That is, the additional mass elements in **Models 3 and 4** are not significantly reflected in the system response. Therefore, we reject these two models in order to avoid unnecessary complication in our analysis. Our choice is then between **Models 1 and 2**.

We have already stated in Section 4.2 that **Models 1 and 2** are equivalent if the conditions of equations 4.1 through 4.3 are satisfied. Although, we have already chosen standard values for our parameters, it is worthwhile to verify the equivalency. We find that if we chose the actuator parameters of **Model 2** to be:

$$K_s = 40; \quad K_p = 40/3; \quad B_p = 32/3,$$

then the conditions of equations 4.1 through 4.3 are fulfilled if **Model 1** maintains the standard parameters. And we may calculate the new eigenvalues of **Model 2** to be:

$$\lambda_{1,2} = -0.8164 \pm 2.4906i$$

$$\lambda_3 = -4.3672$$

We observe that these eigenvalues are identical to those of **Model 1**, and we conclude the two models are essentially equivalent in their dynamics.

Given the dynamic equivalence, we recognize that the structure of **Model 2** more closely represents the physical components that make up the muscle system. Also, the actuator of **Model 2** (using our standard parameters) is somewhat faster than that of

Model 1. As previously stated, we prefer a fast actuator for our system, so that the control signals will be followed with high fidelity. On these bases, we select **Model 2** as our reference structure. Our task is now to develop a control system that provides an input command to the force generator element. We shall begin this development in the next chapter.

CHAPTER 5

DESIGN OF A NOVEL ROBOTIC CONTROL SYSTEM

5.1 INTRODUCTION

Many practical tasks to be performed by the industrial robot fundamentally require dynamic interaction between the robot and its environment. The environment might be a passive object or, say, an active power tool which is manipulated by the robot. Alternatively, the environment may be another robot working cooperatively, or it may even be some system with variable properties. As we mentioned before, the interaction between the robot and its environment will, in general, change the performance of the robot and may also cause contact instability. Therefore, the capability of a robotic system to execute its tasks is strongly dependent upon its ability to achieve the desired performance in a stable mode, particularly when the environment makes up part of the system.

Our ultimate goal is to investigate the design of a general control scheme for robotic systems that guarantees no degradation of performance and stability when the robot is isolated from, or coupled to, a wide variety of environments. That is, by using this control scheme, the robot can follow a desired command “closely” and the local stability of the system is also guaranteed whether the environment is passive or active, known or uncertain.

Although the conventional closed-loop feedback control system may be used to influence the behaviour of the robotic system, it creates an essential trade-off between performance and stability. That is, for better performance, we can choose high gains for

the feedback control system, but this may destabilize the robotic system. Hence, control of the robotic system is not exclusively a matter of good performance, but preservation of stability must also be achieved.

Stability problems emerged immediately after the development of the robotic control system, especially when dealing with contact tasks. Although it has been addressed with more sophisticated control algorithms, faster computation, and better sensors, we still lack a general control scheme capable of generating the desired and stable movement required by all kinds of tasks in industry. In this thesis, we will mainly discuss one possible solution of the contact stability problem as the basis for obtaining an acceptable performance.

It is a matter of common observation that human beings have no difficulty maintaining stability as well as desired performance when interacting with all kinds of environments. This ability makes us believe that there may be a consistent criteria that human beings utilize to make the decisions to control their movements. In order to improve the robotic control system design towards solving the stability and performance problems of contact systems, and eventually make robot operate as well as human beings, we require an understanding of two important issues. The first of these, which was introduced in Chapter 2, is how the human being normally controls muscle movements via the neural pathways. The second, which will be investigated in this chapter, is how we may best implement a control strategy drawing on the features of the human control strategy.

5.2 AN EMULATION MODEL OF NEURO-MUSCULAR CONTROL SYSTEM

When a novel control system is to be investigated for a complex system such as robot, a methodical course of action must be planned first. In this thesis, we are interested in how the robotic system might be controlled in ways similar to the human

motor control system. Therefore, our first step, which was covered in Chapter 2, is to discuss some basic structures and characteristics of human neuro-muscular system. From there, we learned that human motor response is based on the simultaneous cooperation of many anatomical components (the muscles, the brain, the receptors that provide information about the states of the muscles, and so on). As such, we introduced a block diagram of neuro-muscular feedback control system, which is shown in Figure 2.5. Based on this diagram, we will propose our own robotic control system in a way to consider the biological components so that each corresponds to a robotic part; muscle to actuator, brain to controller system, receptors to sensors or observers.

We present the basic structure of our proposed control system in Figure 5.1. This control system consists of “hardware” components (e.g., sensors or observers, actuator and its structure) combined with a controlling “software” component. In the following part of this section, our concern will be with how our robotic control system corresponds to the human motor system, particularly, what the difference is between these two, and why it exists.

Figure 5.1 A simplified block diagram of our robotic control system

Compared with Figure 2.5, we can see that our model, Figure 5.1, is a relatively simple control system. This is because our work is a preliminary study of robotic control system design based on the human CNS such that an unduly complicated model is not required to improve our insight. On the other hand, our goal is to implement the most desirable features of the strategies used by the CNS in our robotic control system. Hence, in our design process, we may add, omit, or interpret in a manner not precisely consistent with the biological system, the existing characteristics of the human neuro-muscular system. We therefore recognize that the robotic system is not expected to perform exactly as would a human being.

When selecting the desirable features of the CNS strategies, we first must recognize that, for robotic applications, some neurological behaviours are unacceptable. As we know, certain neuro-muscular feedback loops which involve the interneurons within the spinal cord provide for the reflex action, while others involve the higher centers of the CNS (the brain). But for a robotic system, we would only be interested in commands that are generated directly from the higher level of the CNS. This is because we do not wish the robotic system to have spinal-cord-like reflexes otherwise it might be able to run amok like a barn-yard chicken with its head cut off. Therefore, we may omit the intersection of afferent to efferent pathway within the spinal cord in our emulation model of the biological nervous system. Hence, we consider the spinal cord and the brain collectively as a single controller system. Therefore, all of the sensory sources (spindles and Golgi tendon organs) will provide information directly to the controller system (i.e., the CNS).

From Figure 2.5, we recall that there are two neural pathways: the alpha system and the gamma system. The alpha system was identified to play a role in the reflex action. But from our discussion above, we see that this feature of the alpha system will not be required for our robotic system. We retain the alpha system for its role in global task performance. One of the functions of the gamma signal was identified to maintain

the sensitivity of the inelastic spindle. On the other hand, our robotic system need not be limited by the mechanical features of biological tissue. That is to say that the function of the spindle in our robotic system can be assumed to be accomplished by an elastic element. Therefore, our gamma pathway need not provide a signal to the spindle sensor for the purpose of keeping the spindle sensitive. Instead, we retain the gamma pathway for its role in maintaining stability in our system by receiving signals from the brain to coactivate with the alpha system. We believe that it is this alpha-gamma coactivation which exists in the CNS that provides an important, unique, and unifying motor control principle to the human limb. Since each of our control pathways (α and γ) may function somewhat differently from the biological analog, we will discuss each in detail in the next section. After all, for robotic control system, the CNS would be represented by a computer controller system made up of chips (instead of neurons) whose purpose is to process information and make command decisions for the actuator to control the movements.

Figure 5.2 A block diagram of the actuator system representing equations 4.10 and 4.11

As we know, the CNS determines the command to the muscles that generate the force required to effect the desired motion. In our emulation model, we use an actuator to replace the muscles. It will transfer the controller command into the force to perform the

tasks. The details of the actuator system design were discussed in the previous chapter. The actuator block of Figure 5.1 may therefore be replaced by a structure representing the equations of dynamics of the actuator. A block diagram of the actuator system described by equations 4.10 and 4.11 is shown in Figure 5.2, where the signal flow is more clearly illustrated.

In the neuro-muscular system, there are force, position and rate sensitive components inside the muscles that provide the CNS with information about what the muscles are doing. The sense organs within the tendons can signal how hard the muscle are pulling, i.e., the force generated inside the muscle. The sense organs within the muscles can signal how much and how fast the muscle are being stretched, i.e., the position and rate of the muscle stretch. However, we must realize that, for a mechanical robotic system, it is not necessarily straightforward or desirable to directly measure each system state variable. Fortunately, we may implement an observer (or estimator) system to gain access to all of the system's state variables. As shown in Figure 5.1, we replace the muscle sensing elements with a so-called recognition system which includes observers and sensors. Because the main function of an observer system is to provide the information needed by the controller system to generate a command for the actuator, we can only design the details of it after having designed the controller. Therefore, we will discuss the detail structure of our observer system in section 5.4.

For the human motor system, we assume that the desired command, (i.e., the trajectory specification), is supplied from higher levels of the CNS and will drive the limb muscles in some purposeful way. Particularly, for the voluntary motion, we believe that the desired position is stored within the CNS and that the CNS must be able to interpret this into a corresponding muscle length. At the same time, the actual length of the muscle fibers is sensed by the receptors and fed back to the CNS. By comparing with the desired state, the CNS generates an error signal to the nervous system and evokes a correcting command to the muscles. Therefore, the CNS not only effects the control of movement but also takes the lead in the initiation of a movement. In our robotic closed-

loop system shown in Figure 5.1, the feedforward command for intended movements (i.e., task) is “stored” in the controller and is compared with the information about the movements as they actually occur which arrive from the recognition system via the feedback loops. On the basis of this comparison, the controller system will make a compensatory command to the actuator. Our robotic system is considered to be a so-called “tracking system” because its output, y , is equal to a position state (x_2). The path that the robot is required to track is analogous to the trajectory specification generated by the CNS.

From the above, we have proposed a neuro-muscular-like model that can emulate the functional capability of a biological system for robotic control system, and which is also suitable comprise between the necessary depth of information and the desired simplicity and clarity. However, until now, we have only provided a block diagram of our control system. In next two sections, we will discuss the detail design of the controller and recognition (observer) systems.

5.3 DESIGN OF THE CONTROL STRATEGIES

In this section, we will develop a formal description of the two controller parts (α and γ) of the emulation model, Figure 5.1. This means that the control strategy (i.e., the control “law”) must be rigorously defined for its application in robotics.

Until now, we have discussed the neural control scheme in general terms such as “CNS”, “higher center”, “brain” and etc. This generality was necessary because we know that the CNS is the foundation of human control system which has to make use of the proprioceptors’ information during the course of movement to appropriately adjust the activation state. But the way in which the CNS controls human movement, or, which variables are controlled by the CNS in the movement, is not so well understood.

Researchers [34] have found that the control variables may include the muscle length, or the acceleration and velocity, or the effort and/or the energy which is needed in order to execute a task, amongst others. This is primarily due to the fact that the workings of the brain cannot be readily measured (i.e., a dump of the brain source code is not available).

However, what we have identified as unique to the neuro-muscular system is the existence of the two distinct neural command pathways: alpha and gamma. And there exists a direct control of muscle activation from the CNS to the alpha system (to innervate the extrafusal muscle fibers), and to the gamma system (to innervate the intrafusal muscle fibers). Researchers [29] have observed that the extrafusal activation of muscle (i.e., the alpha command) is always accompanied by the intrafusal activation (i.e., the gamma command). Hence, we believe that the alpha and gamma systems share responsibility for the performance of tasks, i.e., two different types of control systems are involved in human movements. On the other hand, it is also observed that the CNS commands exert their predominant effects on the alpha system, with subsidiary adjustment of the gamma system. Based on this, we may infer that the alpha command is responsible for global task control (i.e., regulation or tracking).

On the other hand, the strategy of the gamma system is still somewhat mysterious with the exception of its role in the maintenance of the spindle sensitivity (discussed in Chapter 2). In fact, the intrafusal muscular activity appears to be modulated in ways somewhat *independent* of the extrafusal fibers, which may suggest that the gamma system may operate locally, at times, in opposition to the alpha system. Further, the existence of the gamma system is theorized to have some influence upon the stabilization of the muscular system, and allows for the performance of coordinated tasks in a dynamic environment. This is because the presence of the gamma system allows for smooth movements even in the face of variable environmental conditions. Based on this, we assume that the CNS must include some dynamic optimization criteria in the gamma system that adapts to the environmental load, and it is this feature that makes the

neuromuscular control system so distinctive, although there is no experimental evidence to confirm or deny this theory.

We may use these phenomenological observations to guide the design of our robotic control strategy. That is to say that we need not understand explicitly the control strategy employed by the brain, as long as we can impose a well-understood strategy that appears to generate an equivalent phenomenological results. Furthermore, we may do well to explore complex control systems, such as those where more than one control law may exist at different hierarchical levels. As such, we will proceed to discuss our robotic control scheme in terms of the two components that individually represent the functions of the alpha (global control) and gamma (local control) system.

The Alpha System: Proportional-Integral (PI) Controller

Since the alpha system in the CNS regulates the limb movement, (i.e., ensuring that the muscle contracts along a path that is commanded by the brain), it is reasonable to assume that a “proportional” control law is in effect. For a proportional control law, as shown in Figure 5.3, the command signal (F) grows in proportion to the difference (i.e., error) between the desired system output and the actual system output.

Figure 5.3 Block diagram of a proportional controller. The proportional gain is represented by K .

The explicit position control law:

$$F = K \times (x_d - x_2) \quad (5.1)$$

Figure 5.4 Step response of the proportional controller with $K = 5$ (solid line) and $K = 10$ (dashed line).

may only be the starting point in an actual controller design. As shown in Figure 5.4 (step responses for $K = 5$, $K = 10$), the proportional control system demonstrates a steady-state error that is inversely proportional to the feedback gain. But our robotic controller should guarantee that the end-point of the actuator (i.e., x_2) follows the trajectory command very closely, and that the steady-state error cannot be compensated by re-calibrating our reference scale since the system may also be affected by disturbances. Hence, a proportional-only control law simply will not satisfy our phenomenological requirements. In particular, we may observe that the human is capable of maintaining a desired limb position even when subjected to disturbances. Therefore, in order to ensure that the manipulator follows the desired path in the limit, a higher-order control law is needed for the global task control.

In order to make the steady-state output equal to the command input, i.e., zero steady-state error, an integral control term is mandatory. This idea is a classical one; we simply place an integrator in the forward path in series with the system.

Figure 5.5 Block diagram of a proportional plus integral (PI) controller. The proportional gain and integral gain are represented by K and K_i respectively.

Based on the above, a proportional plus integral (PI) controller is naturally considered for implementation. The structure of the PI control is illustrated in Figure 5.5 below. Compared with Figure 5.3, an integrator is inserted in the feedforward path and its command is proportional to the time integral of the error quantity. This PI structure will allow us to achieve zero steady-state error. Such a configuration of placing a PI controller in the feedforward path is one of the most simple control designs and is commonly encountered in practice. The advantage of this method is the simplicity of the design, and it is appropriate for most industrial applications. Additionally, its simplicity will not overly confuse the situation when later developing the gamma system model.

By implementing a PI control law

$$F = \left(K + \frac{K_I}{s}\right) \times (x_d - x_2) \quad (5.2)$$

the output of the PI controller (F) should be a non-negative feedback gain K times the difference between the desired path x_d and the actual output position x_2 plus the difference integral multiplied by a gain K_I .

As shown in Figure 5.6, by choosing the proper gains, $K = 5$ and $K_I = 50$, the PI controller guarantees that the end-effector of the robot will follow the trajectory closely regardless of the initial error. Hence, it is an effective method of resolving the steady-state accuracy issue and response speed. Therefore, a simple proportional-integral (PI) controller is chosen for the positioning control purpose which represents the alpha system in our robotic control system.

From the above, it looks like that to design a controller which approaches the performance criteria is a fairly simple matter. However, from the system's root locus plot, Figure 5.7, we can see that the system is only conditionally stable. For low values of K , the system is stable; for high values of K , the system is unstable. That means, for an

Figure 5.6 Step response of the proportional plus integral controller with $K = 5$ and $K_i = 50$.

Figure 5.7 Root locus in terms with K .

increased feedback gain K , the system response will be quickened, but a loss of stability may result, specifically when the lag (the integral term) is added. Therefore, as a result, the maximum feedback gains which may be used to achieve the best performance of the closed-loop system are limited.

Further, the robotic system (including actuator and controller) is not necessarily restricted to working by itself against a passive environment; it should be capable of working in conjunction with other systems, (i.e., to execute constrained tasks). If only a position controller is used in constrained maneuvers, the robot-environment interaction forces are treated as disturbances and the controller rejects them, thus aggravating the interaction forces. The usual consequences of this type of interaction are saturation, instability, and physical failure. Therefore, the interaction forces in constrained maneuvers must be accommodated rather than resisted. We realize that an explicit position control law is inadequate; control of dynamic behaviour is also required, particularly when dealing with constrained tasks. Stability of the isolated system is of course primarily essential, however, as the manipulator is expected to interact extensively with its environment, coupled stability is equally as important.

As we seek to achieve more capability from biological emulating robots, an additional control scheme is deemed necessary. We now turn our attention to the analysis of the gamma controller.

The Gamma System: Perturbation Controller

Compared to the alpha system, relatively little has been qualified about the gamma system properties. How does the gamma system fit into the control of intended movement? We do not have clear answers to this fundamental question, but new theories are coming forward. With the specific aim of identifying possible control principles derived from the gamma system, let us first consider the robotic application.

In order to find the actuator force which should be commanded to achieve a given prescribed payload motion, there is a unique solution when a single robot grasps the object. When two or more robots grasp the object, the total energy imparted to the payload system must be a certain amount in order to make the payload follow the prescribed path. But the situation that arises is described as a redundant actuation, meaning that the individual actuator forces required of two robots are not unique. Instead, there are infinitely many combinations of the individual forces that will allow for a particular motion of the object. A classical method in robotics is to specify a cost function and use the redundancy of the manipulator in an attempt to minimize this cost function.

A human being has no difficulty maintaining stability when interacting with all manner of environments, including redundant tasks. Since the role of the alpha system is defined to singularly ensure the global task performance, we believe that when dealing with an interactive (or redundant) task, the gamma system will be active in the neuro-muscular control system with the purpose to reconcile the redundancy. Therefore, in some way, we can say that the gamma system provides a strategy that may use some kind of optimizing criteria in order to achieve the system stability.

Our central hypothesis is then that the gamma system provides some sort of optimal function to the neuro-muscular control system to stabilize the performance, especially when dealing with contact tasks. Therefore, our emulation of the gamma control strategy should be based on the notion of optimizing the processing of the sensory information in order to achieve a stable movement. The immediate problem, then, is to determine which measure of the actuator characteristics, such as power, or energy, or velocity, or any combination, should be used in our optimizing strategy.

Researchers suggest [31] that a biochemical reaction may control the rate of energy (i.e., the power) released in order to minimize the energy cost. In fact, we observe that a human can “sense” the change in the load as a task is being performed, and can

adapt in a manner that minimizes the expenditure of energy. For example, if a person who is pushing a stalled car at a steady rate is joined by a second individual, then the first person will quickly reduce his effort in order to maintain a steady rate of total work done on the payload. This, in fact, maintains the completion of the task at a reduced energy rate of the individual labourers. If the first person did not reduce his effort, then the load (i.e., the car) would accelerate under the application of effort by the second individual. Consequently, the velocity would increase and the energy output of the first person would therefore increase, since

$$E = \int F \cdot V dt \quad (5.3)$$

where E is the work (energy) output, F is the applied force, and V is the speed of the load.

Hence, we suggest that a minimal-energy scheme may be one of the ways that the CNS controls the behaviour of the muscular system through the gamma system. Where the alpha system ensures completion of the global task, the gamma system may minimize the energy output of the muscle, thereby stabilizing the system. As evidence, it is observed that the gamma system is not well developed in infants, and, as a result, their limbs tend to flail wildly before finally grasping an object in their sight. An energy-efficient control system simply should not allow such behaviour.

This energy-minimization theory is also applicable for robotic systems. For contact robotic systems, to ensure coupled stability, usually each dynamic systems must be, in itself, internally stable. Physically, the stability of each system in isolation guarantees that its energy is always decreasing in the absence of a driving command. Since coupling the two or more systems together does not generate any energy, then, the total energy of the coupled system is also decreasing and the coupled system is stable. Hence, it is sensible to use some form of energy-minimization controller to implement the gamma system function into our robotic control strategy.

In recent years, there have been a number of attempts to apply optimal control theory to the analysis of human movements [37]. All of these attempts have been motivated by the belief that optimal control theory is a useful tool for modelling the control of the human neuromuscular system. However, the modern optimal control methodology (e.g., the Linear Quadratic Regulator, “LQR”, method) is often not applicable to the practical design problem. This is simply because the design objective often cannot be expressed in closed mathematical terms (which is needed by the LQR). It is also impractical to predict the effect on closed-loop behaviour of a given pair of weighting matrices [16]. Therefore, we realize that it is difficult from an optimal control standpoint to *directly* achieve the minimum energy. Instead, we will seek a different formulation to develop and implement the gamma system emulation.

In order to fundamentally emulate the CNS ability, in this section, we will propose a new algorithm, the *perturbation controller*, which has a biological analog to describe the function of gamma system that is consistent with the possible optimal control principles in human movement. Based on the observation of human behaviour, a possibility exists that the gamma system persistently tests the response of the environment to an attempt to reduce the energy output. However, as we consider the notion more thoroughly, we realize that we cannot merely state that our controller’s objective is to minimize energy output because it would then only single-mindedly reject any application of effort. Instead, what we propose is a strategy that continuously tests the system response to an incremental application of effort (hence the term perturbation controller). And, depending upon the states of the system at the instant of application of the effort increment, the incremental work output will either be positive or negative. We then use this criteria of positive or negative work output to decide whether the effort should be increased or decreased further to reduce the energy output. Practically, we must apply the effort increment in a discrete, regular, period and we call this increment a “force perturbation”. Essentially, the force perturbation represents a very light “tapping” on the system. Each perturbation adds a minute amount of energy to the system, but we

may realize the desired minimization of work by implementing our feedback criteria as described.

Ultimately, by constantly testing the work output by adding a small amount of perturbation force to the system, and striving to minimize the energy output, the gamma controller will provide a stabilizing influence on the system, especially for contact tasks. By this, the gamma system is used to modify the commands given to the muscle when it faces environmental changes. Additionally, the gamma system may function when the human is uncertain about the load to be handled and work towards a smooth and energy-efficient completion of the task.

To illustrate the perturbation control scheme more clearly, a block diagram of the perturbation controller is shown in Figure 5.8. In the figure, the path-tracking command signal, α , follows directly from the PI controller. This signal is summed with a force pulse train to drive the actuation system. The result is that the system globally follows the desired path, but is locally perturbed by the light tapping of our force pulse train.

Figure 5.8 Block diagram of the perturbation controller

Since power is simply the time derivative of energy, the system states $[x_1, x_2, x_3]$ are used to measure the power first and this measurement is represented by the “Power Calculation” block in Figure 5.8. Even though the power supplied by the actuator is consumed in a number of ways, the useful component is that which is used to displace the load. This means the power to be considered is:

$$P = K_s \times (x_1 - x_2) \times x_3 \quad (5.4)$$

where P is power and K_s is the series stiffness defined in Figure 4.10. Since we are only interested in the power delivered over the duration of the force pulse, we use the pulse train to trigger the Power Calculation block. After that, the power is integrated over time (Energy Integrator) to yield the work delivered to the environment throughout the duration of the pulse.

Our next block, Energy Comparison, in the figure has the task of comparing the work delivered to the environment during the pulse with the work that the system would have transferred if the pulse had not been imposed on the system. The result is a measurement of the incremental work delivered to the environment by the pulse. In order to know what the work transfer would have been in the absence of the pulse, we need to make a “System Image” model, that represents the identical dynamics of the actuation system, yet is not perturbed by the force pulse train. Then the image power will also be integrated in time, over the pulse period, to yield the work output that would have been delivered to the load if the force pulse had not existed. We then subtract the image work output from the actual work output to predict a measure of the incremental work performed on the environment by the pulse alone, as described earlier above. This work quantity is fed back to the system via a negative feedback gain, G , to yield the gamma signal which essentially represents an influence to drive the aggregate incremental work energy to its minimum.

To accomplish the System Image block's function, we need to introduce an observer system to track the image states rather than actual states while the pulse is imposed upon the real system. Also, the observer can align the image states with the true states after the pulse has terminated to ensure that the image system state error does not propagate. The detail of the observer design will be covered in the next section.

5.4 DESIGN OF THE OBSERVERS

In order to fully implement the control strategies developed in the last section, information about all of the system's state variables (i.e., x_1 , x_2 , and $\dot{x}_2$) should be available for feedback. For the human neuro-muscular control system, the CNS receives information about the muscle states through two sensory receptors: the muscle spindles and Golgi tendon organs. The muscle spindles provide information regarding the muscle length and velocity; the Golgi tendon organs sense the force generated inside the muscles. However, for robotic systems, we realize that it may be physically difficult or otherwise undesirable to directly measure each of the state variables with sensors. For example, when robots work in a space orbit, it may be impractical to measure position or velocity with respect to an earth bound reference.

In the case of our muscle-like actuator system which was discussed in Chapter 4, the output of the system is indicated by x_2 (i.e., the actuator endpoint position). On the other hand, the actuator system has three state variables in total, so that the other two state variables may not be accessible to measurement. In reality, for the human muscle structure, it would be difficult to directly measure the position, x_1 , without violating the integrity of the whole system. However, if the system is observable, then those state variables which are not accessible to measurement can be estimated by a so-called observer or estimator. In some circumstances state-variable estimates may be even

preferable to direct measurements, because the errors introduced by the sensors that provide these measurements may be larger than the errors in estimating these variables (this is commonly the case when measuring velocities such as $\dot{x}_2$).

An observer, shown in Figure 5.9, is a device that uses the inputs and outputs of a system to produce estimates of this system's state variables. By using the information about those state variables that can be measured directly by the sensors, the observer is able to estimate other state variables of the system which may not be directly accessible to measurement. In particular, it has been demonstrated [30] that, for any observable linear system, an observer can be designed having the property that the estimation error (i.e., the difference between the state of the actual system and the state of the observer) can be made to go to zero as fast as one may desire. This may be practical because the nature of a system and its estimator is quite different: the observer is either an electronic unit or an equivalent digital device which can operate beyond the dynamic limitations of typical mechanical devices, hence, the observer gains may be set arbitrarily high.

Figure 5.9 Block diagram of a system and its observer

To design an observer for the control system, the first concern is to determine which measurement opportunities the system indicates. In our case, that means we need to determine which state variables of our muscle-like actuator system are accessible to direct measurement and which will need to be estimated for the control scheme.

Regarding our PI controller which was shown in Figure 5.5, we only need to know the actuator endpoint position x_2 . When a robot is in contact with its environment (i.e., a constrained robotic system), we initially assume that no relative motion between the load

and the actuator exists. Thus, the motion of the load can be determined from the motion of the actuator, and vice versa. Hence, the information about the position, x_2 , can be obtained from a position sensor connected at the load.

To determine the power delivered to the environment (i.e., the load) by the robot, we need information about the force output from the actuator to the load which is equal to $K_s \times (x_1 - x_2)$, as well as the velocity $\dot{x}_2$. Therefore, we need to design an observer which can estimate the two additional state variables x_1 and $\dot{x}_2$ in order to implement the perturbation control scheme. Since our primary concern in this thesis is the control strategy, not the observer details, we will just introduce a simple full-order state observer which is able to accomplish this necessary function. Based on the state-space representation of our actuator system, equations (4.10) and (4.11), the structure of the observer is defined as:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \quad (5.5)$$

where $\hat{x}$ is the estimate of the actual states x , and L is the proportional gain which must be chosen to achieve satisfactory performance of the observer. Since there is only a single output of the system, x_2 , the observer gain matrix, L , becomes a column vector:

$$L = [L1, L2, L3]^T \quad (5.6)$$

By comparing equation (5.5) with the equation (4.10), which represents the dynamics of the robotic system, we notice that these two equations have the same form, except that the expression of the observer has an additional input $L(y - C\hat{x})$. Without this additional part, the observer becomes open-loop. Since it is impossible to always set the observer's initial condition $\hat{x}(0)$ equal to the actual system's initial condition, $x(0)$, the estimated states of an open-loop observer would have either a continually growing error or an error that goes to zero too slowly to be of general use. We can easily solve

this problem by adding the closed-loop part, $L(y - C\hat{x})$, which feeds back the difference between the measured and estimated outputs to correct the system continuously until the estimated states converge to the true states.

A block diagram showing the implementation of the equation (5.5) is given in Figure 5.10 to clarify the structure of the observer. We can see that the upper part of this block diagram has the same structure as the actuator system (Figure 5.2), except that the state variables, $\hat{x}_1$, $\hat{x}_2$ and $\hat{x}_3$, are estimates of those of the actuator system.

Figure 5.10 A block diagram representation of the closed-loop observer (Observer1)

Our perturbation controller has a complication that we must consider: recall that the command is based upon the comparison of two quantities: the first being the actual power output of the actuator system; and the second being the *nominal* power output that the system would deliver were it not influenced by a perturbation. Therefore, we actually need two sets of state variables to calculate these power quantities. Consequently, we

must provide a separate observer (we will call it “Observer2”) for the nominal states in addition to the observer (“Observer1”) that estimates the actual system states. Observer2 has a structure very similar to Observer1, and is shown in Figure 5.11. We note, however, the presence of a switch in the feedback branch. In our control strategy, this switch will open when we impose a perturbation, so that Observer2 will operate in an open-loop mode. At the instant that the switch opens, the states of Observer2 are identical to the states of Observer1, and to those of actual system. Therefore, Observer2 will track the states as if the system were not influenced by the perturbation. This is why we refer to Observer2’s states as the “nominal states”. On the other hand, when the switch closes, Observer2 becomes closed-loop and its states quickly converge to those of Observer1, so that there is no error prorogation prior to the next open-loop event.

Figure 5.11 A block diagram representation of Observer2

5.5 THE COMBINATION OF CONTROL LAW AND OBSERVER

We have proposed a control law which assumes that all of the state variables are available. We have also specified two observers which can estimate the full state when provided with the measurements of the system as indicated by its output. The final step, which is covered in this section, will consist of combining the control law with the observers, where the control law calculations are based on the estimated states rather than the actual states. Hence, we will obtain a general control scheme for robotic systems in which not all of the state variables may be measured.

For a tracking system, there are several methods [15] to introduce the desired command input to the system. We will use the one which is preferred [15] for commanding the system, as illustrated by Figure 5.12, which shows a block diagram of the control system with two full-order state observers. For this configuration, the input, F , will excite Observer1 in precisely the same way that it excites the actuator system; thus the estimator error caused by the initial condition will quickly go to zero by choosing appropriate gains, L . After that, the estimate error will remain zero and the observer dynamics will not be further excited. Hence, the command will excite the system in a manner consistent with the controller only.

Figure 5.12 Block diagram of the whole system which includes controllers, actuator system, and observers.

We see that Observer2 taps into the input signal prior to the pulse generator. In this way, our image system will not be influenced by the pulse train. On the other hand, notice that one of the inputs to Observer2 is essentially taken from the PI controller output, F_pi , which is given by:

$$F_pi = (K + \frac{K_i}{S}) \times (x_d - x_2) \quad (5.7)$$

This signal will be influenced by the perturbation which affects the states of the system. As such, our Observer2 is not totally isolated from the perturbation, however the influence of the perturbation on the signal F_pi will be only very slight. We can still be confident that Observer2 will provide a good estimate of the nominal system states.

5.6 SUMMARY

Finally, it is appropriate to review our design procedure that was developed in the last two chapters. The design process begun with the development of a mathematical model of muscle-like actuator system. This model was used to determine the performance specifications: characteristics of the desired system behaviour as well as a definition of the variables to be measured. Following this, we have covered in this chapter, three steps:

Step 1 (section 5.3). Design the control law, or in other words, decide by what means the control is to be accomplished. This is done under the assumption that all the state variables of the system are available for use.

Step 2 (section 5.4). Design an observer (or, in our case, two observers) to estimate those states which are needed for the control law of Step 1, but are not directly measured.

Step 3 (section 5.5). Combine the control law design of Step 1 with the observer design of Step 2 to obtain a total control system design.

We can see that the robotic control system, as shown in Figure 5.12, is based on the neuromuscular system which was a major aim of this work (i.e., to propose a control scheme which has a biological analog). The fundamental principle behind the control algorithm is that it emulates the human CNS by optimizing energy output while fulfilling the primary requirement of following a given trajectory. In Figure 5.12, there are two complementary sets of feedback loops (outer-loop and inner-loop), each with its own control law, which are shown to control a common plant, the actuator system. The PI controller is in the outer-loop of the control system, and since it is for the global task performance, it can be considered to be primary. The perturbation controller is in the inner-loop and its purpose is to ensure a stable performance when the robot executes constrained tasks. Hence, it is considered subordinate to the PI controller. Finally, the control command F may be expressed as:

$$F = F_{pi} + dF \quad (5.8)$$

where F_{pi} is the PI controller command and dF is the output of the perturbation controller.

When the robotic system deals with non-contact task (i.e., the environment is passive), the perturbation controller (i.e., the inner-loop) may be deactivated. Then, the system reduces to only include the outer-loop, a PI controller, which is a simple closed-loop positioning system. When robot and the active environment are in contact, both controllers, i.e., PI controller and perturbation controller, are active. Then, the position of robot is controlled along the command path while locally minimizing the energy consumption of the driving actuator will result in stabilizing characteristics during trajectory. We also realize that another advantage of our perturbation controller is that it

does not need a prior knowledge of the environment in order to provide a stabilizing influence.

Once the structure of the control system has been developed, the system parameters (i.e., the controller and observer gains) can be fine-tuned using common control techniques, (s-plane, optimization theory, simulation, etc.). Since our system is non-linear, we find that the best tool for evaluation and development is computer simulation. On the other hand, the ultimate goal of our work is to design a robotic system which has the capacity to successfully complete tasks in a stable mode under all circumstances (i.e., with all kinds of environments). A comprehensive simulation exercise of the design will confirm stability and allow us to verify the true performance to be expected from the system. If the results of the simulation are favorable, the system could be prototyped. We will only consider the simulation part which will be discussed in next chapter.

CHAPTER 6

SIMULATION

6.1 INTRODUCTION

In this chapter, we will investigate the performance of the robotic control system proposed in the previous chapter. Since the control scheme (a global PI controller combined with a novel local perturbation controller) is nonlinear, it is difficult to analyze the system behaviour using conventional methods. Instead, we will build a computer simulation model of the system in SIMULINK® to illustrate the application and utility of our control scheme. Also by use of simulation, the controller and observer parameters can easily be modified interactively to reach the combination that achieves the best performance.

In this introductory work, we mainly concern ourselves with the stability (particularly the contact stability) and near-optimal performance of our robotic control system, even though some other requirements, such as noise sensitivity, and robustness to unmodelled dynamics, are also important. In order to verify that the control scheme is general and versatile, we will consider two types of interaction. The first of these is a single robot that works against a passive environment. In this situation, we expect that the local perturbation controller will attempt to minimize the actuator energy output, while the PI controller attempts to follow the desired path with accuracy. But we expect the PI controller to dominate the response. The second type of interaction involves two robots working cooperatively to complete a common task so that each robot may be considered to be individually working against an active environment. In this situation, the perturbation controller of each robot will attempt to output no more energy than the

other robot. Hence, we expect the two robots to ultimately work in unison with equal energy output, rather than result in a back-and-forth struggle which is so typical of interacting robots.

As a result of simulation, any of the system state variables or a combination (e.g., power) can be plotted as a function of time or of another variable to illustrate the response. In the plots that follow, all forces are in Newtons, and displacements are in metres, velocities are in metres/second, powers are in Watts, and work / energies are in Joules.

6.2 SIMULATION PARAMETERS

Before the results of a computer simulation are obtained, the simulation parameters such as the desired position or path command, initial conditions, as well as the simulation time step must be specified. We will define and discuss these parameters individually in the following text.

Path Command

For a path-tracking robotic system, the selection of the command (i.e., the desired path) is arbitrary. However, we will define our desired path based on some common criteria, such as to be: a) general; b) practical; and c) consistent.

For the desired path to be general, we must not select some complex arbitrary motion. Instead, for our one-degree of freedom actuator model, it is appropriate to define an “A” to “B” type path, such as is shown in Figure 6.1. To add to the generality, we do not select point A as the origin; rather, it is better to choose point A as having the opposite sign (positionally) to point B. This will ensure that our robot must execute both a push and a pull motion as defined by the geometry of Figure 4.10. As shown in Figure

6.1, the selection of point A being $x_d = -2$ and point B being $x_d = 2$ is arbitrary, but presents a practical extension of the actuator defined in Figure 4.3.

Figure 6.1 The path command which is defined as x_d

On the other hand, the selection of the break point, t_1 , is also arbitrary, but we provide a sufficient lag from $t = 0$ to $t = t_1$. This ensures that the system will have ample time to approach point A prior to being subjected to the ramp command, in the event that the system's initial position is not coincident with point A. Further, this will allow us to adequately visualize the ability of the system to track the ramp without the influence of the transients. The rate of the rise between point A and B is chosen so that it is rapid enough to require a sufficiently fast control algorithm (this will be demonstrated in our simulations), but not so rapid as to be an un-realistic command for a mechanical system with finite mass, stiffness and damping. Moreover, the selection of this command path shall be consistent regardless of initial conditions, control schemes or gains.

Initial Condition

To study the response of the simulated system, it is necessary to specify a set of initial conditions of the system, represented by $[x_1(0), x_2(0), x_3(0)]^T$. That is, each state variable must be assigned a value representing its state when the simulation is to begin. In general, we can arbitrarily select any combination of initial values for our three states. However, we recognize that it is most practical to use the trivial states, i.e.,

$[x_1(0), x_2(0), x_3(0)] = [0, 0, 0]$ as our initial condition. Note that this initial condition satisfies the static equilibrium based on the manner in which the states were defined in Figure 4.10. On the other hand, the initial position of the end-effector of the actuator, x_2 , is zero which does not coincide with the point A of our command path (Figure 6.1). Therefore, this will enhance the generality of our simulation.

Time Step

The time step (T_s) is needed by the computer to run the simulation and should typically be chosen such that the computer sampling effect is negligible for control analysis. Usually, the frequency of the time step (i.e., T_s^{-1}) should be on the order of 20 times the system's closed-loop bandwidth. This ensures that the simulation will capture all of the important system dynamics. On the other hand, T_s should not be chosen too small or else the simulation becomes too computationally intensive with no benefit to the accuracy of the result. We will find that a time step of 0.003 sec. is appropriate for our simulation.

6.3 CONTROL SYSTEM PARAMETERS

For the PI controller, we need to define the proportional gain, K , and integral gain, K_i . For the perturbation controller, we need to define the feedback gain, G , and the perturbation's parameters, such as height of the perturbation (H_p), duration of the perturbation (W_p), and the perturbation period (T_p). On the other hand, for the two observers which were defined in Chapter 5, we need to define their feedback gains ($[L1, L2, L3]$). Because of the non-linear nature of our control system, especially for the perturbation controller, the selection of its parameters is primarily done empirically with the aid of simulation. That is, a set of practical gains for use in the control system shown in Figure 5.12 is selected such that the simulated response demonstrates satisfactory

performance. We will discuss the selection of each of these parameters separately in this section.

PI Controller: K and K_i

For a single actuator system, the PI controller will dominate the system performance. This was discussed in Chapter 5: the perturbation controller will tend towards zero work output, but the PI controller will insist that the command path be followed. Hence, in order to choose K and K_i , we may simply turn off the perturbation controller and select parameters that provide satisfactory performance. In this way, the PI controller gains can be chosen using conventional method, such as root locus techniques.

For a typical PI controller, we realize that a large proportional gain, K , will make the system fast but also jeopardize the stability. This is because the root locus plot in terms of K , as we recall from Figure 5.7, tends to branch into the right half of s-phase. The actuator system's dominant eigenvalues, from Chapter 4, are:

$$-0.6810 \pm 2.3699i$$

and we observe that they are already very close to the imaginary axis. Based on this, we will pick a modest proportional gain, say $K = 5$, since we must also add a destabilizing integral term.

The step response¹ of the proportional-only system (i.e., $K = 5$, $K_i = 0$) is shown in Figure 6.2. Comparing with Figure 4.11, the proportional feedback gives the system a fast response, smaller tracking error, but more oscillation. In order to eliminate the steady-state error without further increasing the proportional gain, K , we insert an integrator into the control law. Given that the steady-state error of the proportional-only system with $K = 5$, is approximately 94 percent, we need a substantial integral gain, K_i , to rapidly eliminate this steady-state error. However, we must be careful to not select parameters for the PI controller such that the performance is outstanding. This is because

¹ The computer simulation code, in the form of Matlab mfiles, are presented in Appendix B for all analyses in this section. SIMULINK diagrams are given in Appendix C.

Figure 6.2 Step response of the proportional controller with $K = 5$.

we need to leave some space for visualization of the influence of the perturbation controller when it is introduced. We may readily find that an integral gain of $K_i = 50$ results in a satisfactory response.

The step response of the complete PI controlled system, with $K = 5$ and $K_i = 50$, is illustrated in Figure 6.3. It shows that the closed-loop system is fast enough to track the step command: the response is smooth and the steady-state error is zero. If other values are chosen, such as K_i is reduced to 10, as shown in Figure 6.4(a), the response is too slow to reach the steady state. In Figure 6.4(b), K_i is increased to 100, which results in unstable oscillation.

Figure 6.5 shows the desired path, as defined in section 6.2, overlaid by the system output. From that, we can see the steady-state error is too small to be visible on this scale. Hence, we can conclude that the PI controller system tracks the command path in a satisfactory manner.

Observer: $[L1 \ L2 \ L3]$

Before defining the perturbation controller's parameters, we need to specify the observer gain matrix, L . This is because the function of the perturbation controller depends on the observers. As a rule of thumb, the observer poles usually might be chosen to be a factor of 2 to 6 faster than the controller poles. This is to ensure sufficiently fast decay of the estimation error. In other words, the observer can sense and react to the changes of the states and estimate them as quickly as possible to avoid propagation of the estimation error. As a consequence, the controller rather than the observer will dominate the system response.

As we mentioned before, since the observer has no physical limitation on its feedback gains, we can choose them to be as high as we desired. However, high gains do result in a high bandwidth which might pass more sensor noise to the actuator.

Figure 6.3 Step response of the proportional plus integral controller with $K = 5$ and $K_i = 50$.

Figure 6.4 Step response of other value set of K and K_i . (a) response for $K = 5$, $K_i = 10$. (b) response for $K = 5$, $K_i = 100$.

Figure 6.5 Typical closed-loop response to the path command, x_d , with $K = 5$ and $K_i = 50$.

Therefore, in a manner analogous to the controller design, the best observer design will incorporate a balance between a good transient response and a low enough bandwidth so that sensor noise does not significantly affect the actuator. Here, we may simply pick, through empirical methods, the observer gains to be:

$$L = [80,80,80] \quad (6.1)$$

Figure 6.6 shows that, regardless of the initial error, the estimated states rapidly converge to the system states.

Perturbation Controller: the Perturbation Generator and G

To complete the definition of the perturbation controller, we must first define a periodic train of force perturbations. This is most readily done by adapting a standard pulse-generator² signal, as shown in Figure 6.7 and imposing this signal on the system.

Figure 6.7 Description of the force pulse train.

We see from the figure that the pulse train is defined by the pulse height H_p , the pulse duration W_p , and the pulse period T_p .

² Although SIMULINK provides a pre-defined pulse generator, in the version used, it is found to have inconsistent behaviour and cause erratic side effects on the simulation. Therefore, we define our own pulse generator signal for our simulation.

Figure 6.6 Comparison of the actual system output (x_2 —solid line) with the estimated output state ($\hat{x}_2$ —dashed line) using observer gains, $L = [80, 80, 80]$. (a) no initial error. (b) initial error equals 0.2.

Our first consideration is the pulse period, T_p , which essentially defines the frequency of the perturbation. Recall that the perturbation train is intended to behave as a light, but rapid, tapping on the system. The frequency should be approximately one order of magnitude faster than the dominant system frequency so that the perturbation controller will be able to influence the system response. With this consideration, we find $T_p = 0.1$ sec. is appropriate since the actuator system exhibits a dominant time constant on the order of 1 sec..

Our second consideration is the pulse duration or width W_p . We intend that the perturbation signal add only minimal energy to the system, so we keep the pulse width small. This also allows adequate “recovery time” for the system prior to the subsequent perturbation. A duty cycle of approximately 1:3 (i.e., the pulse is “high” for about one third of the period) is appropriate for the pulse generator, so we select $W_p = 0.03$ sec..

Finally, the pulse height, H_p , must be defined for the pulse generator. We understand that the perturbation height influences the amount of energy given to the system, but this is also related to the feedback gain, G , of the perturbation controller. Hence, it is most practical to define a unit pulse height, i.e., $H_p = 1N$, so that the feedback gain magnitude, G , will be the characteristic number of the feedback loop.

In order to choose the feedback gain, G , we realize that this is best done through an empirical process for our non-linear system. A larger magnitude of the gain will result in more influence of the perturbation controller on the system response. Yet there is also the influence of the PI controller on the system which ensures that the system follows the command path. Ultimately, we find that a perturbation feedback gain of $G = -10$ is satisfactory for demonstrating the performance of the perturbation controller when using a single actuator.

When a *single* actuator works against a passive environment, the perturbation controller is a sort of redundancy in the system. Therefore, the perturbation controller's

gain, G , should be relatively small, and in fact, may possibly even be set to zero so that the system will be dominated by the PI controller only. The reason for this, as was discussed in section 5.3, is that the PI controller will command the system to follow the desired path, whereas the function of the perturbation controller always is to minimize the actuator's energy output. If the perturbation controller were to have a large influence on the system, it would tend to reduce the system output to zero instead of following the desired path.

Until now, we have only discussed the function of the perturbation controller without a real demonstration. In order to clearly illustrate, we will first deactivate the PI controller by making $K = 0$; $K_i = 0$ and set the energy integrator's (see Figure 5.8) initial value to unity, meaning that there is an initial controller force of one Newton. In this configuration, the only controller operating is the perturbation controller. Theoretically, the perturbation controller should command the system to reduce the energy output to zero, thereby turning the system "off". As shown in Figures 6.8(a) through (c), when G is increased, the perturbation controller command, dF , (i.e., the system input) is more rapidly reduced from its unity offset to zero and the system output, x_2 , goes to zero more quickly too, as we expected. This means that our perturbation controller is functioning properly as it naturally drives the system to a minimum energy output.

When we add the PI controller without an initial force offset on the perturbation controller and given a task to track the desired path which was defined in last section, we achieve the system response shown in Figure 6.9. We notice that when G is too large (e.g., $G = -100$), the magnitude of dF is comparable to the magnitude of F_{pi} . Hence, the PI controller does not dominate the system response and a large steady-state error results. This is not desirable and therefore we will do well to limit the size of G .

We find that the performance is satisfactory when G is chosen to be -10, since it is large enough to demonstrate that the perturbation controller works, but does not overly influence the system performance.

Figure 6.8(a) Response of the single actuator system with the perturbation controller only for $G = -1$. The upper figure is the position, x_2 , of the environmental load as defined in Figure 4.10. The lower figure is the perturbation controller feedback force, dF . It is seen that this force decreases from its initial offset value in order to minimize the energy output.

Figure 6.8(b) Response of the single actuator system with the perturbation controller only for $G = -10$. The upper figure is the position, x_2 , of the environmental load as defined in Figure 4.10. The lower figure is the perturbation controller feedback force, dF . It is seen that this force decreases more rapidly than for the lower gain, $G = -1$.

Figure 6.8(c) Response of the single actuator system with the perturbation controller only for $G = -100$. The upper figure is the position, x_2 , of the environmental load as defined in Figure 4.10. The lower figure is the perturbation controller feedback force, dF . It is seen that this force decreases more rapidly than for the lower gain, $G = -10$.

Figure 6.9 Single actuator system response to the path command for $G = -100$. The upper left figure shows the position, x_2 , of the environmental load tracking the command path, x_d . The upper right figure shows the perturbation controller feedback force, dF . The lower left figure shows the PI controller command, F_{pi} . The lower right figure shows the sum of dF and F_{pi} . Notice that dF and F_{pi} are of comparable magnitude so that the perturbation controller is not operating subordinate to the PI controller for this gain, resulting in a large steady state error.

Now that the choice of the control system parameters have been justified, let us further describe the system behaviour under the influence of our control scheme. As shown in Figure 6.10, for x_2 to respond from its initial condition of zero to the origin of the command path at $x_d = -2$, the PI controller must pull on the system, so that the command signal F_{pi} is negative. During this event, the perturbation controller will tell the system to save energy by not exerting a pull. Therefore, the output of the perturbation controller, dF , is positive in opposition to the negative force F_{pi} . After x_2 tracks positive, the PI controller's output force will necessarily be positive, and the perturbation controller's output start to decay to a negative value, in an attempt to minimize the energy output.

Summary

In actual design problems, especially for nonlinear systems, it is necessary to consider several different set of gains and carry out computer simulations to find the ones that yield the best overall system performance. This design procedure has been repeated for several values of K , K_i and G until a final set is chosen that reach an acceptable response. Table 6.1 summaries the parameters used in the development of our control system and for the demonstration of the system performance.

K	K_i	T_p (sec)	W_p (sec)	H_p (N)	G	L
5	50	0.1	0.03	1	-10	[80,80,80]

Table 6.1 The final selection of values for the control system parameters.

Note that until now, we have completed the control system design, and demonstrated the system behaviour with a single actuator. However, to further our development, two or more robots cooperating on a single environment should be considered to test the control system's behaviour under a constrained task environment.

Figure 6.10 Single actuator system response to the path command for $G = -10$. The upper left figure shows the position, x_2 , of the environmental load tracking the command path, x_d . The upper right figure shows the perturbation controller feedback force, dF . The lower left figure shows the PI controller command, F_{pi} . The lower right figure shows the sum of dF and F_{pi} . Notice that F_{pi} dominates dF so that the perturbation controller is operating subordinate to the PI controller for this gain, resulting in a satisfactory tracking of the command path.

The remainder of this chapter is devoted to the simulation of our control system for a cooperative manipulation.

6.4 COOPERATIVE SYSTEM

The objective of our novel robotic control system is to gain a satisfactory performance as well as the stability under all kinds of circumstances. In the previous section, we demonstrated how the single actuator system behaves under the command of our control system. In order to fully examine the application or feasibility of the proposed control scheme, we will demonstrate its use in situations involving the interaction with an active environment in this section.

In this work, we will demonstrate one example of a contact task to illustrate the application of the proposed control scheme. For the purpose of understanding the application of the scheme, the problem to be considered is simplified but general: two robots handling one object (environmental load) in a cooperative system. The term cooperative system is used to refer to groups of robots working together. In this example, the active environment is simply another robot. This is a very common case, particularly in unstructured environments such as space or aquatic domains. And, the control of multiple cooperative robot systems represents a significant step in complexity and computational requirements in comparison to that of a single robot. The complexity arises from the fact that the cooperative manipulation forms a closed kinematic chain which induces kinematic and dynamic constraints. Hence, complex problems such as collision avoidance, synchronization of the robots, etc., are required to be solved.

Once again, our simulation efforts will focus on the cooperative robot system to execute the selected task which was defined in section 6.2. And we consider a particular case of two robots configured in a parallel connection, both moving a common environmental mass along a desired path as illustrated by Figure 6.11. The main feature

of a parallel connection is that the bases of all the robots are fixed on the ground and the manipulators are in the same operational space and attached to a common object.

Figure 6.11 Block diagram of two parallel robots configured in a cooperative system (left). Mechanical model of the cooperative system (right).

In order to attain the desired action of the mass of environment, we realize that it is essential to properly coordinate two robots. It is usual to designate one of the two robots as “master”, the other one as “slave” [43], in order to reconcile the problems of having a closed kinematic chain. In our work, we do not have this subordination, as each actuator independently attempts to complete the task. This normally would result in a back-and-forth struggle between the two robots as they fight against the closed kinematic chain constraint. However, our novel work-minimizing control strategy is expected to allow the actuators to cooperate as they execute the task in unison, with neither actuator working to get ahead of the other. This emulates the features of the human system where it is found [32] that limbs appear to be controlled individually at first when executing a common task, but that they ultimately learn to cooperate.

In order for us to understand how our control scheme coordinates the actions of mutually interacting actuators, and especially to allow us to study the effects of the perturbation controller utilized during cooperation, first, we will consider the two robots to have identical structure and parameters. This means the system is powered by two identical actuator systems which are individually dictated to by their respective controllers. Such a configuration allows us easily to observe the influences of the perturbation controller.

The method of monitoring the co-activation effect of the cooperative system is via simulation³. In the following text, we refer to the two robots as “Actuator1” and “Actuator2” respectively. We will demonstrate the performance of the dual-actuator system by subjecting it to a number of basic tests as listed below:

- Test 1: “balanced path tracking”
- Test 2: “unbalanced static position holding”
- Test 3: “PI-only controlled path tracking”
- Test 4: “path tracking with an initial unbalance”
- Test 5: “non-identical robots in a cooperative system”
- Test 6: “non-identical perturbation control gains”
- Test 7: “output position calibration error”

Test 1: “balanced path tracking”

Figure 6.12 illustrates the situation where the two actuators are activated at the same time, $t = 0$, and both have no initial force or stored energy. Therefore, to make the load follow the desired path, x_d , each of the individual PI controller and perturbation controller commands should be same. That is, the two actuators are to function in the same manner and share the load equally.

³. The computer simulation code, in the form of Matlab mfiles, are presented in Appendix D for all analyses in this section. SIMULINK diagrams are given in Appendix E.

Figure 6.12(a) Dual actuator response for Test 1: balanced path tracking. The upper left figure shows the position, x_2 , of the environment load tracking the desired path, x_d . The upper right figure shows Actuator1's perturbation controller output $dF1$. The lower left figure shows Actuator1's PI controller output F_{pi1} . The lower right figure shows the total force $F1=dF1+F_{pi1}$ driving Actuator1. Note that $F1$ is identical to $F2$ on Figure 6.12(b), so that the robots work in unison. Also notice that there is a small steady-state error in the position x_2 .

Figure 6.12(b) Dual actuator response for Test 1: balanced path tracking. The upper left figure shows the total actuator force $F1+F2$. The upper right figure shows Actuator2's perturbation controller output $dF2$. The lower left figure shows Actuator2's PI controller output F_{pi2} . The lower right figure shows the total force $F2=dF2+F_{pi2}$ driving Actuator2. Note that $F2$ is identical to $F1$ on Figure 6.12(a).

Figure 6.13(a) Dual actuator system response for Test 2: unbalanced static position holding for $G = -10$. The upper left figure shows the position, x_2 , of the environment load and the desired static position, x_d , corresponding to an equilibrium force of 20N. The upper right figure shows Actuator1's perturbation controller output $dF1$. The lower left figure shows Actuator1's PI controller output F_{pi1} . The lower right figure shows the total force $F1 = dF1 + F_{pi1}$ driving Actuator1. Note that $dF1$ is initially offset by 15N meaning that Actuator1 is initially supporting 15N of the total load. Again also note that there is a small steady-state error in the position x_2 .

Figure 6.13(b) Dual actuator system response for Test 2: unbalanced static position holding for $G = -10$. The upper left figure shows the total actuator force $F1+F2$. The upper right figure shows Actuator2's perturbation controller output $dF2$. The lower left figure shows Actuator2's PI controller output F_{pi2} . The lower right figure shows the total force $F2=dF2+F_{pi2}$ driving Actuator2. Observe that $F1$ and $F2$ are approaching a steady state.

Test 2: “unbalanced static position holding”

Now consider the result when the two actuators begin the task with different initial force levels, such as Actuator1 having a 15N load, and Actuator2 having a 5N load. This is to say that the two actuators begin with an unbalanced fraction of the total force. Their goal is hold the load at the position of a 20N static equilibrium. As shown in Figure 6.13, where both perturbation controller's feedback gains are equal $G = -10$, both perturbation controllers' outputs $dF1$ and $dF2$ decrease, and the PI controllers' outputs F_{pi1} and F_{pi2} increase to compensate for the loss in dF . However, we see that the total force output of Actuator1 ($F1=dF1+F_{pi1}$) is reduced where that of Actuator2 increases. Also note that a steady-state error exists. As the magnitude of G is increased to, say $G = -50$ or -80 , as shown in Figure 4.14 and 4.15 respectively, the total output of each actuator more rapidly achieves its equilibrium level. This means that Actuator1 decreases its output energy while that of Actuator2 increases, and, at last, balance at identical force output levels. In this way, the two robots can cooperate very well, instead fighting with each other or maintaining an unbalanced duty of force. We do note that the position output decays from its desired position and a small steady-state error results. This is because the balanced equilibrium force achieved by each actuator is somewhat less than the ideal $20N/2 = 10N$. This is obviously a situation that may need to be addressed with a more sophisticated alpha system algorithm. We also find that when $G = -80$, the two actuators reach their balance quickly enough and the steady-state error is reasonable, so, we will choose this to be our nominal perturbation feedback gain when further dealing with the two actuator cooperative system.

Test 3: “PI-only controlled path tracking”

In order to demonstrate how our robotic system is benefited by the perturbation controller, we consider a test where we initially deactivate each perturbation controller. Therefore, only PI controllers command the system to track our standard path. As shown in Figure 6.16, Actuator1 begins with 100N force offset, and Actuator2's initial force is zero. Since the perturbation controllers are deactivated, their outputs ($dF1$ and $dF2$) are steady. From the figure, we can see that the difference of Actuator1 and Actuator2's

Figure 6.14(a) Dual actuator system response for Test 2: unbalanced static position holding for $G = -50$. The upper left figure shows the position, x_2 , of the environment load and the desired static position, x_d , corresponding to an equilibrium force of 20N. The upper right figure shows Actuator1's perturbation controller output $dF1$. The lower left figure shows Actuator1's PI controller output F_{pi1} . The lower right figure shows the total force $F1 = dF1 + F_{pi1}$ driving Actuator1. Note that $dF1$ is initially offset by 15N meaning that Actuator1 is supporting 15N of the total load.

Figure 6.14(b) Dual actuator system response for Test 2: unbalanced static position holding for $G = -50$. The upper left figure shows the total actuator force $F_1 + F_2$. The upper right figure shows Actuator2's perturbation controller output dF_2 . The lower left figure shows Actuator2's PI controller output F_{pi2} . The lower right figure shows the total force $F_2 = dF_2 + F_{pi2}$ driving Actuator2. Notice that F_1 and F_2 have achieved a steady state more rapidly than for $G = -10$.

Figure 6.15(a) Dual actuator system response for Test 2: unbalanced static position holding for $G = -80$. The upper left figure shows the position, x_2 , of the environment load and the desired static position, x_d , corresponding to an equilibrium force of 20N. The upper right figure shows Actuator1's perturbation controller output $dF1$. The lower left figure shows Actuator1's PI controller output F_{pi1} . The lower right figure shows the total force $F1 = dF1 + F_{pi1}$ driving Actuator1. Note that $dF1$ is initially offset by 15N meaning that Actuator1 is supporting 15N of the total load.

Figure 6.15(b) Dual actuator system response for Test 2: unbalanced static position holding for $G = -80$. The upper left figure shows the total actuator force $F1+F2$. The upper right figure shows Actuator2's perturbation controller output $dF2$. The lower left figure shows Actuator2's PI controller output F_{pi2} . The lower right figure shows the total force $F2=dF2+F_{pi2}$ driving Actuator2. Observe that $F1$ and $F2$ have very rapidly achieved a steady state, and thus we will use $G = -80$ as our baseline gain for further simulation.

Figure 6.16(a) Dual actuator system response for Test 3: PI-only controlled path tracking. The upper left figure shows the position, x_2 , of the environment load tracking the desired path, x_d . The upper right figure shows Actuator1's perturbation controller output $dF1$. The lower left figure shows Actuator1's PI controller output F_{pi1} . The lower right figure shows the total force $F1=dF1+F_{pi1}$ driving Actuator1. Note that $dF1$ has an initial value of 100N and this value holds steady throughout the task.

Figure 6.16(b) Dual actuator system response for Test 3: PI-only controlled path tracking. The upper left figure shows the total actuator force $F1+F2$. The upper right figure shows Actuator2's perturbation controller output $dF2$. The lower left figure shows Actuator2's PI controller output F_{pi2} . The lower right figure shows the total force $F2=dF2+F_{pi2}$ driving Actuator2. Note that $F2$ remains 100N lower than $F1$. The PI-only controller robots do not cooperate.

output is always 100N. This is because the PI controller only depends on the difference between the system's output, x_2 and the desired path, x_d . And this path error is identical for both actuators, therefore, each PI controller's output is identical. These two actuators have not cooperated as evidenced by the fact that their effort output does not balance. This situation is analogous to two persons pushing a stalled car where one person is contributing much less effort than the other.

Test 4: "path tracking with an initial unbalance"

Now let us consider the same test, this time with active perturbation controllers, as illustrated with Figure 6.17. Compared with Figure 6.16, we see that the force outputs are initially unbalanced but ultimately become balanced to reach the same effort. That means for Actuator1, it very rapidly reduces its effort output from its initial value by taking advantage of the presence of Actuator2. As the result, Actuator1's output force is decreasing and Actuator2's output force is increasing until they reach their average, demonstrating the utility of our local perturbation controller.

Also from Figure 6.17, we notice that there is a smooth distribution of force between the two actuators. This is an important point, especially for high speed maneuvers, as the force distribution will affect how well the actuators will be able to faithfully produce the desired forces and, eventually, desired path.

Test 5: "non-identical robots in a cooperative system"

To further our analysis, we should consider a system composed of two non-identical actuators operating on a common environment. We will consider two actuators having the same structure but different parameters. We will only demonstrate the different damping and different perturbation controller gains because these two should have the greatest influence on the energy cost.

Figure 6.17(a) Dual actuator system response for Test 4: path tracking with an initial unbalance. The upper left figure shows the position, x_2 , of the environment load tracking the desired path, x_d . The upper right figure shows Actuator1's perturbation controller output $dF1$. The lower left figure shows Actuator1's PI controller output F_{pi1} . The lower right figure shows the total force $F1=dF1+F_{pi1}$ driving Actuator1. Note that $dF1$ is initially set to 100N.

Figure 6.17(b) Dual actuator system response for Test 4: path tracking with an initial unbalance. The upper left figure shows the total actuator force $F1+F2$. The upper right figure shows Actuator2's perturbation controller output $dF2$. The lower left figure shows Actuator2's PI controller output F_{pi2} . The lower right figure shows the total force $F2=dF2+F_{pi2}$ driving Actuator2. Note that $F2$ has balanced with $F1$ in spite of the 100N initial offset of $dF1$. The robots have cooperated to share the load equally under the command of the perturbation controllers.

Figure 6.18(a) Dual actuator system response for Test 5: non-identical robots in a cooperative system with $B_{p2} = 60\text{Ns/m}$. The upper left figure shows the position, x_2 , of the environment load and the desired static position, x_d , corresponding to an equilibrium force of 20N. The upper right figure shows Actuator1's perturbation controller output $dF1$. The lower left figure shows Actuator1's PI controller output F_{pi1} . The lower right figure shows the total force $F1 = dF1 + F_{pi1}$ driving Actuator1. Note that $dF1$ is initially offset by 15N meaning that Actuator1 is supporting 15N of the total load.

Figure 6.18(b) Dual actuator system response for Test 5: non-identical robots in a cooperative system with $B_{r2} = 60\text{Ns/m}$. The upper left figure shows the total actuator force $F_1 + F_2$. The upper right figure shows Actuator2's perturbation controller output dF_2 . The lower left figure shows Actuator2's PI controller output F_{pi2} . The lower right figure shows the total force $F_2 = dF_2 + F_{pi2}$ driving Actuator2. Note that while the two robots have cooperated to distribute the 20N load, there remains a small unbalance between F_1 and F_2 indicating that Actuator2 is less efficient than Actuator1.

We will again consider the unbalanced static holding application, similar to Test 2, where a desired static equilibrium of 20N is the task, and Actuator1 has an initial force offset of 15N where Actuator2 has an initial force offset of 5N. As shown in Figure 6.18, we assign Actuator2's damping parameter to be $B_p = 60\text{Ns/m}$, which is much larger than that of Actuator1 (6Ns/m). We immediately see that cooperation is still achieved as in Test 2, even though our actuators are not identical. However, we do note that a small residual unbalance in steady-state force exists: $F_1 = 9.0\text{N}$, $F_2 = 9.3\text{N}$. We realize that the actuator with the bigger damper has the smaller force at equilibrium. This makes sense since a greater amount of power absorbed by the damper implies a less efficient actuator, therefore it attempts to provide a smaller force. As damping coefficients increase, the efficiency of the actuator decreases.

Test 6: “non-identical perturbation control gains”

Next, we will consider the dual actuator system, with each robot having a different perturbation controller feedback gain, G . We again configure the system to hold a static equilibrium of 20N as in Test 5 with the initial force duty of Actuator1 being 15N and Actuator2 being 5N. Comparing Figure 6.19(a) and (b) showing a smaller gain, G , for Actuator2 versus Actuator1, we see that Actuator1 will end up bearing a smaller portion of the load than will Actuator2. Essentially, since Actuator1 has a higher perturbation feedback gain, it is driven at a greater rate than Actuator2 to reduce its energy output. This is equivalent to Actuator1 having less capacity, or being “weaker” than Actuator2, and therefore, ultimately sharing less of the load.

Test 7: “output position calibration error”

As the ultimate test of our system, we will command the two actuators to work towards different goals. In this case, Actuator1 will be commanded to a step position of $x_2 = 2\text{m}$ whereas Actuator2 will be commanded to a step position of $x_2 = 1.9\text{m}$. Such a situation can be commonly encountered when there exists a calibration error in the position sensor of either or both robots. That is to say that the two robots might not agree on where $x_2 = 2\text{m}$ is because the calibrations of their positioning sensors. The usual

Figure 6.19(a) Dual actuator system response for Test 6: non-identical robots in a cooperative system with $G2 = -50$. The upper left figure shows the position, x_2 , of the environment load and the desired static position, x_d , corresponding to an equilibrium force of 20N. The upper right figure shows Actuator1's perturbation controller output $dF1$. The lower left figure shows Actuator1's PI controller output F_{pi1} . The lower right figure shows the total force $F1 = dF1 + F_{pi1}$ driving Actuator1. Note that $dF1$ is initially offset by 15N meaning that Actuator1 is supporting 15N of the total load.

Figure 6.19(b) Dual actuator system response for Test 6: non-identical robots in a cooperative system with $G2 = -50$. The upper left figure shows the total actuator force $F1+F2$. The upper right figure shows Actuator2's perturbation controller output $dF2$. The lower left figure shows Actuator2's PI controller output F_{pi2} . The lower right figure shows the total force $F2=dF2+F_{pi2}$ driving Actuator2. Note that while the load burden is distributed so that $F2$ is greater than $F1$ because Actuator1 is the "weaker" of the two robots such that its gain, G , is greater than that of Actuator2.

result is a struggle between the two robots, and consequently high internal forces and perhaps even actuator failure.

As shown in Figure 6.20, we first consider the perturbation controllers being deactivated and only the PI controllers in command. However, there exists a difference between the system output, x_2 , and the desired position, x_d . Notice that the equilibrium position is $x_2 = 1.95\text{m}$ which is the average of $x_2 = 2.0\text{m}$ or $x_2 = 1.9\text{m}$. Consequently, Actuator1's F_{pi1} will go to positive infinity whereas Actuator2's F_{pi2} will go to negative infinity. Such high interactive forces are very common problem found in current robot designs [38], and as suggested, actuator failure may very well be the result.

Now, when we add the perturbation controller to each actuator, as shown in Figure 6.21, even though the PI controller's output will still try to go to infinity, the perturbation controller will actually compensate for this. Eventually, each actuator's total output reaches a steady, finite state. Therefore, our perturbation controllers have allowed the robots to reach a mutually cooperative position and have prevented actuator wind-up and possible failure. We can say that the coordination function of the perturbation controller plays a vital role in constrained task performance. We do note however that there exists a force unbalance between the two actuators. We believe that this may be because Actuator1's steady-state error is relatively larger than Actuator2's, and therefore, its ultimate output is bigger than Actuator2. Nevertheless, this is still a more satisfactory result than the never-ending struggle that arises between the PI-only controlled actuators.

6.5 RESULT SUMMARY

From the simulation results of the cooperative system, contact stability as well as satisfactory performance is verified in order to show the advantage of our control scheme. Particularly, the results demonstrate that the extra command from the local perturbation

Figure 6.20(a) Dual actuator system response for Test 7: output position calibration error with the perturbation controllers deactivated. The upper left figure shows the position, x_2 , of the environment load. The upper right figure shows Actuator1's perturbation controller output $dF1$. The lower left figure shows Actuator1's PI controller output F_pi1 . The lower right figure shows the total force $F1 = dF1 + F_pi1$ driving Actuator1. Note that the equilibrium position is $x_2 = 1.95\text{m}$, the average of 1.9m and 2.0m. Also notice that $F1$ grows without bound as the integral controller continues to windup because x_2 does not reach 2.0m.

Figure 6.20(b) Dual actuator system response for Test 7: output position calibration error with the perturbation controllers deactivated. The upper left figure shows the total actuator force $F1+F2$. The upper right figure shows Actuator2's perturbation controller output $dF2$. The lower left figure shows Actuator2's PI controller output F_{pi2} . The lower right figure shows the total force $F2=dF2+F_{pi2}$ driving Actuator2. Note that $F2$ decreases without bound as the integral controller continues to windup because x_2 does not reach 1.9m.

Figure 6.21(a) Dual actuator system response for Test 7: output position calibration error with active perturbation controllers. The upper left figure shows the position, x_2 , of the environment load. The upper right figure shows Actuator1's perturbation controller output dF_1 . The lower left figure shows Actuator1's PI controller output F_{pi1} . The lower right figure shows the total force $F_1 = dF_1 + F_{pi1}$ driving Actuator1. Note that there is a small position error in x_2 , but the perturbation controller output dF_1 compensates for the integrator wind-up in F_{pi1} so that the total actuator force F_1 reaches a steady output.

Figure 6.21(b) Dual actuator system response for Test 7: output position calibration error with active perturbation controllers. The upper left figure shows the total actuator force F_1+F_2 . The upper right figure shows Actuator2's perturbation controller output dF_2 . The lower left figure shows Actuator2's PI controller output F_{pi2} . The lower right figure shows the total force $F_2=dF_2+F_{pi2}$ driving Actuator2. Note that dF_2 compensates the integrator wind-up in F_{pi2} so that the total actuator force F_2 reaches a constant value.

controller plays an important role during constraint task performance. That is, it can easily coordinate the cooperative system by minimizing individual energy output, in order to stabilize the system. Additionally, the perturbation control law ensures that the control system design is not dependent on a prior knowledge of the environment which provides an important criterion to lead an alternative optimal algorithm to coordinate the robots.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

7.1 SUMMARY

This thesis has explored the design of a robotic system from the perspective of emulating biological neuro-muscular control characteristics in the inanimate robotic controller and actuator. This work was motivated by the common observation that the human manipulator system demonstrates a natural ability to smoothly and efficiently perform complex constrained tasks even within the context of an unknown environment. Conversely, robotic manipulators could well be expected to experience unstable performance when dealing with similar tasks. The objective is to attempt to incorporate certain features of the human neuro-muscular system into the design of a robotic system to improve the overall system.

In order to reach the objective, our work begins by investigating in some detail, the human neuro-muscular system. The biological structure of the muscle actuator is explored so that its function, in terms of phenomenological behaviours is reasonably well understood. Separate from the muscle actuator is the neural control system. The key feature of that is identified is the existence of the alpha system for global task control and gamma system for local stability. This motivates us to further investigate the topic of contact stability. In particular, we generate the notion that an optimal energy control system may be useful for stabilizing system that are redundantly constrained such as cooperative dual actuator systems.

Our robotic system design begins by developing a mechanical model of the muscle-like actuator. This model is based on the biological muscle structure, but is general enough to be applicable to a real mechanical system. We then couple the actuator to a general spring-mass-damper representation of the environmental load.

The design of the control system is accomplished by emulating the alpha system as a global path tracking PI controller. Subordinate to this is the emulation of the gamma system as our novel perturbation controller. The perturbation controller accomplishes the function of minimizing actuator energy output by constantly measuring the incremental work done by a train of force pulses.

At last, several numerical simulations are worked out in effort to demonstrate the performance and stability of the proposed control algorithms and to further encourage our intuition. We first test the validity of our total system by configuring a single actuator command by our control scheme. We then test the ability to maintain the stability of a coordinate task by configuring a redundant dual actuator system.

7.2 CONCLUSIONS

Several important conclusions may be drawn from this work; and they are summarized here.

- The adult human neuro-muscular system is observed to have desirable features in terms of performance and stability. It is hypothesized that these features can be realized in robotic system by emulating the key structures and strategies of the neuro-muscular system.

- The alpha system within the neural control system is recognized to have primary responsibility for the global task completion, whereas the subordinate gamma system plays a key role in maintaining local stability.
- A control system design has been conceived by implementing the alpha-gamma structure into the control strategy. The alpha system is represented by a PI controller whereas the gamma system function is emulated by a novel perturbation controller. The minimal work output strategy of the perturbation controller effectively reproduces the phenomenological function of the gamma system.
- The designed robotic system is effectively tested by computer simulation. These simulations verify the key desired features of satisfactory performance and stability under constrained tasks. In particular, the design is proven to reconcile certain well identified difficulties with conventional cooperative systems, such as integrator wind-up.

7.3 RECOMMENDATIONS

The scope of this work is broad enough that certain simplifications in our analysis and design had to be made in order to keep the work focused. Further progress in robotic systems development could be benefited by exploring, in more depth, those topics that were simplified for our purpose. In particular, our results have shown promise and therefore continued work is merited towards better understanding, by experiment, the human neuro-muscular control system, especially the gamma system. Independent of this, we may do well to experiment with more sophisticated actuator models that, perhaps, include some of the non-linear features of the muscle structure.

For the immediate system design, some further work is also merited. For example, our use of the two observers has required a prior knowledge of the

environmental model. However, we have observed that the human system does not seem to require this sort of knowledge in order to function well. Therefore, we may need a more sophisticated means of developing the information required to implement the perturbation controller's system image. Additionally, we have identified in our simulations, a steady-state error caused by the presence of the perturbation controller. Hence we recognize that a PI controller is not a fully sufficient emulation of the alpha system, and we may do well to put further effort in this direction.

After all, even though the simulation results indicate that the path following and local energy minimization combination is very promising, additional experiments could be performed, such as altering the force pulse shape, introducing time delays and specifying more complex tasks.

BIBLIOGRAPHY

- [1] Adams, J.A.; Human Factors Engineering, Macmillan Publishing Company, New York, 1989.
- [2] An, C.H., and Hollerbach, J.M.; *Dynamic Stability Issues in Force Control of Manipulators*, in Proceedings of the IEEE International Conference on Robotics and Automation, 1987: p. 890-896.
- [3] Beaman, J.J., and Lin, C.H.; *Application of Nonlinear Quadratic Stochastic Control to a Position Servomechanism*, Journal of Dynamic Systems, Measurement, and Control, December 1990: p. 675-679.
- [4] Bhat, S.P., and Miu, D.K.; *Precise Point-to-Point Positioning Control of Flexible Structures*, Journal of Dynamic Systems, Measurement, and Control, December 1990: p. 667-674.
- [5] Brooks, V.B.; The Neural Basis of Motor Control, Oxford University Press, Oxford, New York, 1986.
- [6] Chao, E.Y., and An, K.N.; *Graphical Interpretation of the Solution to the Redundant Problem in Biomechanics*, Journal of Biomechanical Engineering, August 1978: p. 159-167.
- [7] Chapel, J.D., and Su, R.; *Coupled Stability Characteristics of Nearly Passive Robots*, in Proceedings of the IEEE International Conference on Robotics and Automation, 1992: p. 1342-1347.
- [8] Chapman, A.E., and Harrower, P.T.; *Linear Approximations of Muscle Mechanics in Isometric Contraction*, Biological Cybernetics, 1977: p.1-7.
- [9] Colgate, J.E.; *The Control of Dynamically Interacting Systems*, Ph.D., M.I.T. Department of Mechanical Engineering, August, 1988.
- [10] Colgate, J.E.; *Coordinate Transformations and Logical Operations for Minimizing Conservativeness in Coupled Stability Criteria*, Journal of Dynamic Systems, Measurement, and Control, December 1994: p. 643-649.
- [11] Colgate, J.E., and Hogan, N.; *Robust Control of Dynamically Interacting System*, International Journal of Control, 1988: p. 65-88.

- [12] Dubowsky, S., and DesForges, D.T.; *The Application of Model-Referenced Adaptive Control to Robotic Manipulators*, Journal of Dynamic Systems, Measurement, and Control, September 1979: p. 193-200.
- [13] Eppinger, S.D., and Serring, W.P.; *On Dynamic Models of Robot Force Control*, in Proceedings of the IEEE International Conference on Robotics and Automation, 1986: p. 29-34.
- [14] Eppinger, S.D., and Serring, W.P.; *Understanding Bandwidth Limitations in Robot Force Control*, in Proceedings of the IEEE International Conference on Robotics and Automation, 1987: p. 904-909.
- [15] Franklin, G.F., Powell J.D., and Naeini, A.E.; Feedback Control of Dynamic Systems, Addison-Wesley Publishing Company, 1986.
- [16] Friedland, B.; Control System Design, McGraw-Hill Book Company, 1986.
- [17] Fung, Y.C.; Biomechanics: Mechanical Properties of Living Tissues, Springer-Verlag, 1993.
- [18] Hemami, H., and Wyman, B.F.; *Indirect Control of the Forces of Constraint in Dynamic Systems*, Journal of Dynamic Systems, Measurement, and Control, December 1979: p. 355-360.
- [19] Hill, A.V.; *The Heat of Shortening and the Dynamic Constants of Muscle*, in Proceedings of the Royal Society, 1938: p.136-195.
- [20] Hogan, N.; *Impedance Control: An Approach to Manipulation*, Journal of Dynamic Systems, Measurement, and Control, March 1985: p. 1-24.
- [21] Hogan, N.; *Adaptive Control of Mechanical Impedance by Coactivation of Antagonist Muscles*, IEEE Transactions on Automatic Control, August 1984: p. 681-690.
- [22] Iqbal, K., Dogruel, M., and Hemami, H.; *Stability of Linearized Robotic and Musculoskeletal Systems with Feedback Delays*, in Proceedings of the 32nd Conference on Decision and Control, December 1993: p. 1623-1624.
- [23] Kazerooni, H., Sheridan, T.B., and Houpt, P.K.; *Robust Compliant Motion for Manipulators, Part I: The Fundamental Concepts of Compliant Motion*, IEEE Journal of Robotics and Automation, June 1986: p. 83-92.
- [24] Kazerooni, H.; *Robust, Non-Linear Impedance Control for Robot Manipulators*, in Proceedings of the IEEE International Conference on Robotics and Automation, 1987: p. 741-750.
- [25] Kazerooni, H., Bouklas, K.G., and Guo, J.; *Theory and Experiments on the Compliance Control of Redundant Robot Manipulators*, Journal of Dynamic Systems, Measurement, and Control, December 1990: p. 653-660.

- [26] Kazerooni, H., and Guo, J.; *Human Extenders*, Journal of Dynamic Systems, Measurement, and Control, June 1993: p. 281-290.
- [27] Klein, C.A., and Huang, C.H.; *Review of Pseudoinverse Control of Use with Kinetically Redundant Manipulators*, IEEE Transactions on Systems, Man, and Cybernetics, March/April 1983: p. 245-250.
- [28] Lee, S.R., and Srinivasan, K.; *Self-Tuning Control Application to Closed-Loop Servohydraulic Material Testing*, Journal of Dynamic Systems, Measurement, and Control, December 1990: p. 680-689.
- [29] Loeb, G.E.; *The Control and Response of Mammalian Muscle Spindles During Normally Executed Motor Tasks*, Exercise Sports Sci. Rev., 12: p. 157-204.
- [30] Luenberger, D.G.; *An Introduction to observers*, IEEE Transactions on Automatic Control, December 1971: p.596-602.
- [31] Mansfield, J.M.; *Functional Assessment of Amputee Performance*, Ph.D., M.I.T. Department of Mechanical Engineering.
- [32] Marray, R.M, et al.; *Control Primitives for Robot Systems*, IEEE Transactions on Systems, Man, and Cybernetics, January/February 1992: p. 183-193.
- [33] Mason, M.T.; *Compliance and Force Control for Computer controlled Manipulators*, IEEE Transactions on Systems, Man, and Cybernetics, June 1981: p.418-432.
- [34] McMahon, T.A.; Muscles, Reflexes, and Locomotion, Princeton University Press, Princeton, New Jersey, 1984.
- [35] Nahon, M., and Angeles, J.; *Minimization of Power Losses in Cooperating Manipulators*, Journal of Dynamic Systems, Measurement, and Control, June 1992: p. 213-219.
- [36] Orin, D.E., and Oh, S.Y.; *Control of Force Distribution in Robotic Mechanisms Containing Closed Kinematic Chains*, Journal of Dynamic Systems, Measurement, and Control, June 1981: p. 134-141.
- [37] Pandy, M.G., Anderson, F.C., and Hull, D.G.; *A Parameter Optimization Approach for the Optimal Control of Large-Scale Musculoskeletal Systems*, Journal of Dynamic Systems, Measurement, and Control, November 1992: p. 450-460.
- [38] Paul, R.P.; *Problems and Research Issues Associated with the Hybrid Control of Force Displacement*, in IEEE International Conference on Robotics and Automation, 1987: p. 1966-1971.

- [39] Raibert, M.H., and Craig, J.J.; *Hybrid Position/Force Control of Manipulators*, Journal of Dynamic Systems, Measurement, and Control, June 1981: p. 126-133.
- [40] Schmidt, R.A.; *Motor Control and Learning*, Human Kinetics Publishers, Champaign, Illinois, 1982.
- [41] Sharon, A., Hogan, N., and Hardt, D.E.; *Controller Design in the Physical Domain*, IEEE Journal of Robotics and Automation, 1989: p. 552-559.
- [42] Soechting, J.F., et al.; *Evaluation of Neuromuscular Parameters Describing Human Reflex Motion*, Journal of Dynamic Systems, Measurement, and Control, December 1971: p.221-226.
- [43] Uchiyama, M., and Dauchez, P.; *A Symmetric Hybrid Position / Force Control Scheme for the Coordination of Two Robots*, in Proceedings of the IEEE International Conference on Robotics and Automation, 1988: p. 350-356.
- [44] Whitney, D.E.; *Force Feedback Control of Manipulator Fine Motion*, Journal of Dynamic Systems, Measurement, and Control, June 1977: p.91-97.
- [45] Whitney, D.E.; *Historical Perspective and State of the Art in Robot Force Control*, The International Journal of Robotics Research, Spring 1987: p.3-14.
- [46] Wu, C., Young, K., and Houk, J.C.; *A Neuromuscular-Like Model for Robotic Compliance Control*, in Proceedings of the IEEE International Conference on Robotics and Automation, 1990: p. 1885-1890.
- [47] Wu, C., Young, K., and Hwang, K.; *Analysis of Voluntary Movements for Robotic Control*, in Proceedings of the IEEE International Conference on Robotics and Automation, 1991: p. 326-331.
- [48] Yang, B., and Mote, C.D.; *On Time Delay in Noncollocated Control of Flexible Mechanical Systems*, Journal of Dynamic Systems, Measurement, and Control, September 1992: p.409-415.
- [49] Yen, J.Y., Hallamasek, K., and Horowitz, R.; *Track-Following Controller Design for a Compound Disk Drive Actuator*, Journal of Dynamic Systems, Measurement, and Control, September 1990: p. 391-402.
- [50] Yoshikawa, T.; *Analysis and Control of Robot Manipulators with Redundancy*, Robotics Research, 1983: p. 735-747.

APPENDIX A

MATLAB MFILES FOR CHARACTERIZING MODEL 1 THROUGH 4

.

Appendix A

% define the parameters of the system (plant)

```
Ks=30;  
Kp=10;  
Bp=6;  
M=1;  
Me=10;  
Be=10;  
Ke=50;
```

% define the time step

```
t_step=0.05;  
t_top=10;  
t=[0:t_step:t_top];
```

% Model 1 (actuator plus environment)

```
A1=[-Ks/Bp Ks/Bp 0;0 0 1;Ks/Me -(Ks+Ke+Kp)/Me -Be/Me];  
B1=[1/Bp;0;0];  
C1=[0 1 0];  
D1=[0];
```

% Model 2 (actuator plus environment)

```
A2=[-(Ks+Kp)/Bp Ks/Bp 0;0 0 1;Ks/Me -(Ks+Ke)/Me -Be/Me];  
B2=[1/Bp;0;0];  
C2=[0 1 0];  
D2=[0];
```

% Model 3 (actuator plus environment)

```
A3=[0 0 1 0;0 0 0 1;-Ks/M Ks/M -Bp/M 0;Ks/Me -(Ks+Ke+Kp)/Me 0 -Be/Me];  
B3=[0;0;1/M;0];  
C3=[0 1 0 0];  
D3=[0];
```

% Model 4 (actuator plus environment)

```
A4=[0 0 1 0;0 0 0 1;-(Ks+Kp)/M Ks/M -Bp/M 0;Ks/Me -(Ks+Ke)/Me 0 -Be/Me];  
B4=[0;0;1/M;0];  
C4=[0 1 0 0];  
D4=[0];
```

% eigenvalues

```
EI_1=eig(A1);  
EI_2=eig(A2);  
EI_3=eig(A3);  
EI_4=eig(A4);
```

% check controllability and observability

```
c1=det(ctrb(A1,B1));  
c2=det(ctrb(A2,B2));  
c3=det(ctrb(A3,B3));  
c4=det(ctrb(A4,B4));
```

```
o1=det(observ(A1,C1));  
o2=det(observ(A2,C2));  
o3=det(observ(A3,C3));  
o4=det(observ(A4,C4));  
  
% step response  
[y1,x1]=step(A1,B1,C1,D1,1,t);  
[y2,x2]=step(A2,B2,C2,D2,1,t);  
[y3,x3]=step(A3,B3,C3,D3,1,t);  
[y4,x4]=step(A4,B4,C4,D4,1,t);
```

APPENDIX B

MATLAB MFILES DEFINING THE SYSTEM AND SIMULATION PARAMETERS FOR SINGLE ACTUATOR SYSTEM

.

Appendix B

Single Actuator System

```
% define the parameters of the system (plant)
Ks=30;
Kp=10;
Bp=6;
Me=10;
Be=10;
Ke=50;

% define the time step
t_step=0.003;
t_top=45;
t=[0:t_step:t_top];
G1=1/t_step;

% Model 2 (actuator plus environment)
A=[-(Ks+Kp)/Bp Ks/Bp 0;0 0 1;Ks/Me -(Ks+Ke)/Me -Be/Me];
B=[1/Bp;0;0];
C=[0 1 0];
D=[0];

% desired command
for i=1:(t_top/t_step+1),
    if (t(i))<10,
        uu(i)=-2;
    elseif (t(i)<30),
        uu(i)=0.2*t(i)-4;
    else
        uu(i)=2;
    end
end
Xd=uu';

% pulse generator
period_p=0.1;
thigh_p=0.03;
height_p=1;
nperiod_p=period_p/t_step;
nhigh_p=thigh_p/t_step-2;
nlow_p=nperiod_p-nhigh_p-1;
for i=1:(t_top/period_p),
    offset_p=(i-1)*nperiod_p;
    for j=1:nlow_p,
        pusle(j+offset_p)=0;
    end
    for j=1:(nhigh_p+1),
        pulse(j+offset_p+nlow_p)=height_p;
    end
end
```

```

end
pulse(t_top/t_step+1)=0;
p=pulse';
G2=1/height_p;

% sample generator
period_s=0.1;
height_s=1;
nperiod_s=period_s/t_step;
nlow_s=nperiod_s-1;
for i=1:(t_top/period_s),
    offset_s=(i-1)*nperiod_s;
    for j=1:nlow_s,
        sample(j+offset_s)=0;
    end
    sample(offset_s+nlow_s+1)=height_s;
end
sample(t_top/t_step+1)=0;
s=sample;

% proportional-plus-integral controller gains
K=5;
Ki=50;

% perturbatoion controller gain
G=-10;

% observer gains
L1=70;
L2=70;
L3=70;
L=[L1;L2;L3];

% initial condition
x_0=[0;0;0];

```

APPENDIX C

SIMULATION DIAGRAMS FOR SINGLE ACTUATOR SYSTEM

.

Figure C.1 Simulation block diagram of a single actuator system used in section 6.3.

Figure C.2 Expanded simulation diagram of the block in Figure C.1 labelled “System”. The “System” represents the mechanical model of the actuator and environment described in section 4.5.

Figure C.3 Expanded simulation diagram of the block in Figure C.1 labelled “PI_Controller”. The “PI_Controller” represents the alpha system described in section 5.3.

Figure C.4 Expanded simulation diagram of the block in Figure C.1 labelled “Observer1” which was described in section 5.4.

Figure C.5 Expanded simulation diagram of the block in Figure C.1 labelled “Observer2” which was described in section 5.4.

APPENDIX D

MATLAB MFILES DEFINING THE SYSTEM AND SIMULATION PARAMETERS FOR DUAL ACTUATOR SYSTEM

.

Appendix D

Dual Actuator System

% define the parameters of the system (plant)

```
Ks=30;  
Kp=10;  
Bp=6;  
Me=10;  
Be=10;  
Ke=50;
```

% define the time step

```
t_step=0.003;  
t_top=60;  
t=[0:t_step:t_top];  
G1=1/t_step;
```

% Model 2 (actuator plus environment)

```
A=[-(Ks+Kp)/Bp Ks/Bp 0 0;0 0 1 0;Ks/Me -(2*Ks+Ke)/Me -Be/Me Ks/Me;0 Ks/Bp 0 -(Ks+Kp)/Bp];  
B=[1/Bp 0;0 0;0 0;0 1/Bp];  
C=[0 1 0 0];  
D=[0];
```

% desired command

```
for i=1:(t_top/t_step+1),  
    if (t(i))<10,  
        uu(i)=-2;  
    elseif (t(i)<30),  
        uu(i)=0.2*t(i)-4;  
    else  
        uu(i)=2;  
    end  
end  
Xd=uu';
```

% pulse generator

```
period_p=0.1;  
thigh_p=0.03;  
height_p=1;  
nperiod_p=period_p/t_step;  
nhigh_p=thigh_p/t_step-2;  
nlow_p=nperiod_p-nhigh_p-1;  
for i=1:(t_top/period_p),  
    offset_p=(i-1)*nperiod_p;  
    for j=1:nlow_p,  
        pusle(j+offset_p)=0;  
    end  
    for j=1:(nhigh_p+1),  
        pulse(j+offset_p+nlow_p)=height_p;  
    end
```

```

        end
    end
    pulse(t_top/t_step+1)=0;
    p=pulse';
    G2=1/height_p;

    % sample generator
    period_s=0.1;
    height_s=1;
    nperiod_s=period_s/t_step;
    nlow_s=nperiod_s-1;
    for i=1:(t_top/period_s),
        offset_s=(i-1)*nperiod_s;
        for j=1:nlow_s,
            sample(j+offset_s)=0;
        end
        sample(offset_s+nlow_s+1)=height_s;
    end
    sample(t_top/t_step+1)=0;
    s=sample;

    % proportional-plus-integral controller gains
    K=5;
    Ki=50;

    % perturbatoion controller gain
    G=-80;

    % observer gains
    L1=70;
    L2=70;
    L3=70;
    L4=70;
    L=[L1;L2;L3;L4];

    % initial condition
    x_0=[0;0;0;0];

```

APPENDIX E

SIMULATION DIAGRAMS FOR DUAL ACTUATOR SYSTEM

.

Figure E.1 Simulation block diagram of two actuator system used in section 6.4.

Figure E.2 Expanded simulation diagram of the block in Figure E.1 labelled "System". The "System" represents the dynamics of the mechanical model representing a pair of actuators operating on a common environment.

Figure E.3 Expanded simulation diagram of the block in Figure E.1 labelled "Controller1". The "Controller1" represents the PI and perturbation control system commanding the first of the two actuators.

Figure E.4 Expanded simulation diagram of the block in Figure E.3 labelled “PI_Controller”.

Figure E.5 Expanded simulation diagram of the block in Figure E.3 labelled "Observer1".

Figure E.6 Expanded simulation diagram of the block in Figure E.3 labelled "Observer2".

Figure E.7 Expanded simulation diagram of the block in Figure E.1 labelled “Controller2”. The “Controller2” represents the PI and perturbation control system commanding the second of the two actuators.

Figure E.8 Expanded simulation diagram of the block in Figure E.7 labelled “PI_Controller”.

Figure E.9 Expanded simulation diagram of the block in Figure E.7 labelled "Observer1".

Figure E.10 Expanded simulation diagram of the block in Figure E.7 labelled “Observer2”.

IMAGE EVALUATION TEST TARGET (QA-3)

APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved