Soft Computing Based Approaches for the Robust Control of Cooperative Manipulator Systems

by

Wail Gueaieb

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2001

0-612-65242-4

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

In this thesis, we tackle the problem of robust simultaneous position and force control of cooperative robotic (CR) structures in the presence of *ununmodel known time-varying structured* and *unstructured* uncertainties. Most non-adaptive control schemes for the coordinated control of multiple manipulator systems, usually assume a full knowledge of the system's dynamics. This is an unrealistic assumption in many cases since these complex systems are usually subject to the ubiquitous presence of uncertainties. If not dealt with appropriately, these uncertainties may have a dramatic effect on the controller's performance and may even induce instability. Although recent research work carried out in the area of conventional adaptive control has led to a significant improvement in the tracking performance of, both, the payload's desired position/orientation and the internal force desired values, in the face of parametric uncertainties, the majority of it has ignored the effect of unstructured uncertainties on the controller's performance and its stability. Modeling imperfection of complex systems, such as closed-chain robotic manipulators, is inevitable. This makes the development of a robust control approach for the increasingly complex cooperative manipulator systems a necessary step to keep up with the increasingly demanding design requirements of such systems.

In this thesis, we develop novel approaches based on soft computing tools to tackle this complex, yet important control problem. A special type of adaptive fuzzy controllers is first designed to learn the system's overall dynamics without a prior knowledge of it. The controller is then improved even further to provide for a more efficient behavior particularly with respect to computational complexity. Both soft computing based controllers are shown to have excellent tracking abilities of the payload's desired position/orientation while meeting the internal force desired values. The controllers are also shown to be highly robust in the face of a substantial amount of parametric and modeling uncertainties with varying intensity levels. It is also formally proven that, under a few reasonable assumptions, the position and the internal force tracking errors always converge to zero. The results obtained in this work have been very encouraging and would certainly open new opportunities for tackling robot control of complex structures in general. This being said, we believe some more improvements could be done to make the approaches even more powerful and readily implementable in real world environments.

Acknowledgements

I would like to thank Dr. Fakhri Karray for his invaluable help, time, persistent encouragement, advice, ideas, and suggestions throughout this study. Not only has he been a role model, he has also been more than a friend. Without his continuous support and supervision this work would not have been possible. I also would like to extend my gratitude to the rest of my Ph.D. committee members: Dr. O. Basir, Dr. C. W. de Silva, Dr. M. Kamel, Dr. D. Wang, and Dr. A. K. C. Wong, for their constructive comments on the thesis.

I also would like to thank all my friends for their true friendship and continuous support during the critical times of my Ph.D.

Lastly, I am forever grateful to my parents, wife, brothers, and sister, for their unwavering support throughout my student life.

To

Mazin, Hayfa,

my parents, and my grandparents

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Thesis Motivations

1.1.1 Historical Background

Industrial Robots have had major contributions to the tremendous industrial and economical developments that have been witnessed in the last two decades. From a manufacturing perspective, they have been playing a leading role in increasing the productivity, reducing the production costs, and improving the products' quality. Besides manufacturing, robotic systems have been widely used in a drastically increasing number of applications. In fact, one of the major driving forces behind the development of robotic systems, is their potential for replacing humans in hazardous working environments, such as the cleanup of toxic waste, nuclear power plant decommissioning, mining, space exploration, search and rescue missions, security, and surveillance. They are also used to replace humans in performing repetitive type of tasks, such as automated manufacturing, industrial maintenance, spray painting, and spot and arc welding [4,43,48,90].

1.1.2 Why Cooperative Robots?

Although the use of a single robot system is efficiently feasible in many of the existing industrial applications, the need for multi-cooperating robots[1] is becoming increasingly crucial and even necessary at times, particularly in modern industrial frameworks. This is due to several reasons:

- cooperative robots are capable of performing tasks that are either difficult or impossible to be accomplished by a single-arm robot. Such tasks can be manipulating heavy, large, and/or flexible objects, just to name a few. In such cases, grasping the object would make its manipulation more secure.
- the complexity of the working environment, or mission, may require more than one robot of complementary capabilities that may be too expensive to design in a single robot.
- a considerable amount of time may be saved by operating multiple robots to work simultaneously on different aspects of the task in order to successfully accomplish the objective.

[1] As far as we are concerned in this thesis, the term multi, or cooperative, robot systems, refers to cooperative robot manipulators acting on a single object to achieve a common prespecified target goal. We also refer to such systems as multiple and coordinated manipulator systems, in some parts of the thesis.

- multiple cooperating robots may also increase the productivity in a manufacturing environment. This explains the intensive use of multi-robotic systems in assembly operations. for instance.
- designing cooperative robots may. in some cases. be easier or cheaper to perform some tasks than it would be if a single robot were used.

Given this. the use of cooperative robotics is becoming increasingly essential to improve the versatility and potential applications of robotics.

1.1.3 Cooperative Robotics in Industry

Although cooperative robots are still not as intensively used as single robots. they are gradually taking their place in the industry thanks to their high potentials. In a number of industrial applications. a large glass sheet or a large heavy planar object. has to be smoothly transferred from an initial position to a final position. Obviously. a single robot arm cannot succeed in such a mission as the glass sheet may break if handled from one single end. Similarly. and in the case of a large heavy object. it might be just too heavy for a one single arm to accomplish the task. A very convenient way to transfer the object is to use two. or even more. cooperative robotic arms [121]. as shown in Figure 1.1(a).

One of the most successful applications of coordinated manipulator systems is one that was realized by a Japanese firm in mid 1990s. The firm developed a dual robot arm to remotely maintain (and construct. at a later phase) high voltage power cables as illustrated in figure 1.1(b). The system consists of a twin-armed power manipulator that is mounted on a lift truck and is remotely controlled by a

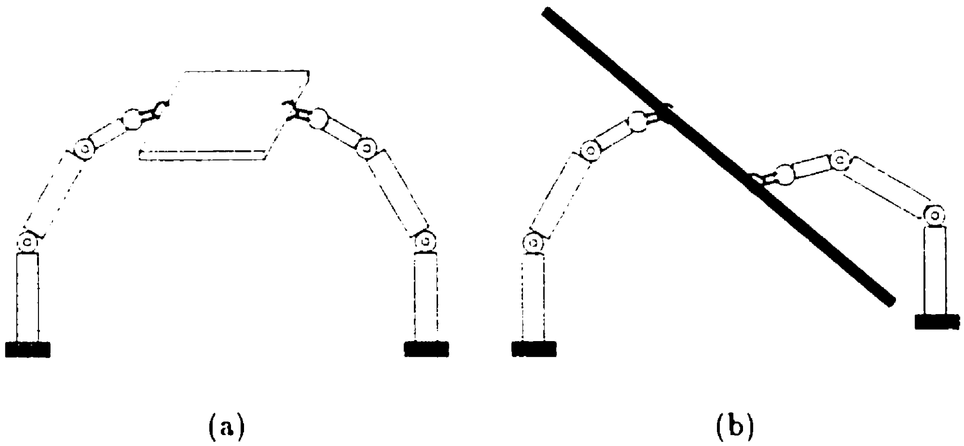


Figure 1.1: (a) Two Cooperative Robot Arms Handling a Flat Object. (b) Two Cooperative Robot Arms Repairing a Power Cable.

technician from either a cabin at the back of the truck, or from a bucket to which the two arms are connected. In some cases, the system is even equipped with a third arm capable of lifting and installing heavy objects. This technique provides

- labor saving: three line men are reduced to one, hence reducing the working time. This is convenient for the customers as it leads to a less expensive service;
- improvement of the working environment: the number of accidents caused by the electric shocks is reduced, proving a safer working environment for the workers;
- simplification of operations: workers do not need to have special skills as most of the operations are done remotely with the assistance of a computer. So the

work that cannot be done manually can be performed safely and accurately;

- automation of frequent tasks: tasks that are repeated frequently are programmed to be automatically performed by the computer, taking the burden of constantly repeating them off the technician.

The system design is so flexible that it can be customized to be applicable to telecommunication, transportation, fire fighting, construction, and many other non-organized working environments.

Cooperative robotic systems have also been used in aerospace applications. Dominating the domain of aerospace technology has been the driving force for the ever increasing interest in cooperative robots in the aerospace domain. A highly advanced dextrous extra vehicular space telerobot with four cooperative manipulators, called the *Ranger Telerobotic Flight Experiment*, is designed as the first telerobot of its kind in the world [90]. Using its ability to freely fly in space, the Ranger spacecraft performs on-orbit serving tasks that were previously accomplished by tethered astronauts only. A schematic figure of the Ranger spacecraft telerobot is shown in figure 1.2.

In addition to the Ranger spacecraft, the Canadian space agency has been studying the possibility of extending Canadarm to a dextrous multiple manipulator system to better serve the international space station Alpha. Such an extension would provide Canadarm with a higher flexibility and maneuvering ability, and a wider range of applications it can perform in space. Moreover, it would enable it to execute several tasks in a shorter time than what it would traditionally take. This is very important especially in space missions in which time is a crucial factor.

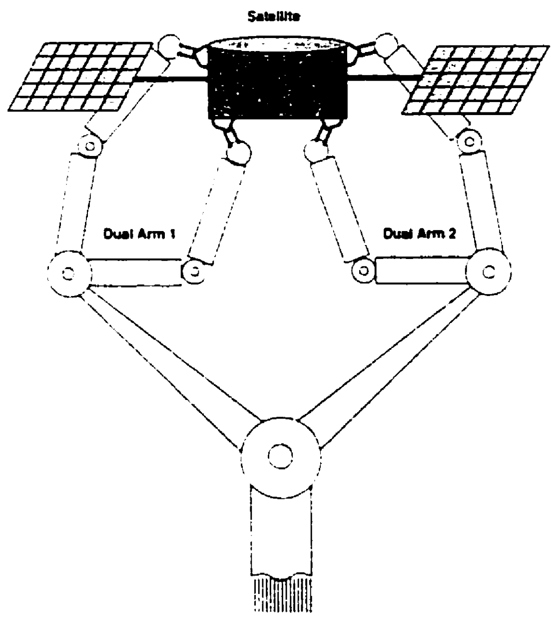


Figure 1.2: Dextrous Extra Vehicular Space Telerobot.

1.2 Main Issues

When multiple manipulators grasp a common object, they form a closed kinematic chain mechanism. This implicitly imposes a set of kinematic and dynamic constraints on the position and velocity of the manipulators as they have to remain in contact with the object while in motion. As a result, the decrease in the degrees of freedom of the whole system leads to the generation of internal forces. Such forces, which have to be controlled appropriately, stem from the direct interaction between the end-effectors and the object. Hence, controlling multiple manipulators interacting with an object, or a given environment, is usually a much more complex problem than that of a single robot control. The motion of the manipulators has to be kinematically and dynamically coordinated. The kinematic coordination means that all the end-effectors involved have to move synchronously to track a certain prespecified desired position and orientation of the manipulated object without losing contact with it. In this case, the dynamic coordination means that the end-effectors have to move in a certain manner as to control the internal forces induced in the system.

The simultaneous position and force control has been usually accomplished by either one of two well known control techniques: impedance control [45,55] or hybrid position/force control [14,92,94,141]. Impedance control usually aims at satisfying a certain desired relationship between the position error and the force acting at the end-effector at which a mechanical impedance is sought to be achieved. The main drawback of impedance control is the assumption that the dynamics of the contact environment can be accurately expressed by a linear equation. In most cases, such a linear formulation is not only very difficult to attain but also requires a full knowledge of the contact environment's dynamics, which is usually not available. The (traditional) hybrid position/force control is based on the idea of splitting the task space into two orthogonal subspaces. One for the direction(s) in which only position control is considered, and one for the other direction(s) in which only force is controlled. The orthogonality between the two subspaces has been assumed to exist in the vast majority of the works. However, in a number of applications this might not be the case and hence such a technique cannot be adopted [20,124].

In real life applications, it is very difficult to design reliable multi-robot control systems using the conventional[2] modeling and control tools. This is due to several factors, including highly dynamic complexities and the often ill-defined workspace environments of such systems. Although coordinating principles for multiple systems have been developed since the early 1980's, most of the work has only addressed the conceptual interpretation and only a few dealt with real-world applications. In particular, the bulk of the research has focused on the control of mobile robots and weakly coupled cooperative manipulators. On the other hand, relatively less work has been done in the area of the control of strongly coupled manipulators where two or more robot arms cooperate to move a certain object along a predefined path. When using cooperative robots of different complex configurations, the dynamics of the robots become completely different and result in major issues in terms of control and motion coordination for real-time applications as different conventional controllers have to be built for robots of different configurations. The coordination of the robots (or robot arms) in a multi-robot system is extremely difficult due to several reasons:

- The lack of *precise* system models.
- The lack of *precise* dynamic parameters.
- The necessity to *simultaneously* control the payload's position as well as the internal forces, and
- The lack of the tools for an efficient system design, analysis, and computation

[2]Throughout this thesis, the terms conventional tools or standard tools refer to those tools that are purely based on precise mathematical models not involving any kind of soft computing approaches. We also refer to such tools as standard tools in some parts of the thesis.

of an optimal solution in real time.

Soft computing techniques based on tools of fuzzy logic, connectionist modeling, and genetic algorithms, have become among the state of the art techniques in the domain of artificial intelligence and expert systems design. They have attracted much interest in the last few years due to their high potential in dealing with large structured and parametric uncertainties of a given system. Appropriate combinations of these tools can make them very powerful to tackle *ill-defined* systems; i.e., systems whose dynamics or working environments are fully/partially unknown or poorly understood. In fact, several combinations of these techniques have been successfully integrated in a wide range of applications: single-robot systems [65,117], mining excavation [101,102], nuclear plants control [96], and medical analyses [109], just to name a few. Integrating soft computing tools with conventional control methods, gave birth to an ever growing research area that has become known in the research community as *intelligent control* [3].

1.3 Thesis Contributions

The focus of this work is on *intelligent control of cooperative robotic systems*. The main thrust is to propose an efficient and advanced control system that would enable the tracking of a predefined desired position and orientation of the payload, while controlling the internal forces of the closed-chain system and make them converge to their predefined desired values. The controller proposed has to satisfy several key requirements:

- compensating for parametric uncertainties: as some of the system parameters might be unknown or slowly varying in time, the controller has to be able to track the payload's desired position/orientation trajectory in the face of such type of uncertainties.

- compensating for modeling uncertainties: in almost every complex control system, there exist modeling errors. The more complex is the system, the larger is the impact of modeling uncertainties on the controller's performance. Hence, it is very important to consider this issue at the design stage to make the controller robust in the face of such types of uncertainties.

- controlling the internal forces: the controller has to simultaneously track the desired trajectory of the manipulated object as well as track the predefined desired internal forces generated between the manipulator end-effectors and the payload, and not just assure their boundedness as in some research works [6, 125, 135]. In our case, this is a challenging constraint as we are mainly interested in the case where the payload, the end-effectors, and the internal forces, have the same lines of action. Thus, the widely used conventional hybrid position-force control technique cannot be applied in this case as it requires orthogonal position and force subspaces.

- portability: the controller has to be generic enough to be efficiently useful. In other words, unlike conventional control techniques, the proposed control strategy has to be independent of a specific robot dynamics so that it can control a wide variety of cooperative manipulator systems with as little customization as possible.

- speed of execution: the controller has to be computationally efficient to keep up with the bandwidth of the closed loop control system.

Inspite of the rapidly growing applications of soft computing techniques. not much work has been done in applying them to cooperative robotic systems. Among the main goals of the current work is to come up with viable alternatives based on soft computing tools. which are more powerful and more efficient than their conventional counterparts in tackling the control problem of cooperative manipulator systems. To accomplish this. two soft computing based adaptive controllers are proposed. in this thesis. They are implemented for a multi-arm cooperative robotic system manipulating a common object. The main contributions of this work are summarized as follows:

- The design and the implementation of two novel control schemes using a decentralized approach and state of the art soft computing based controllers that satisfy collectively the enumerated constraints.

- The derivation of an innovative rule-reduction technique that drastically reduces the number of fuzzy rules used by an adaptive fuzzy controller to control coordinated multiple robots. The proposed rule-reduction algorithm makes the practical use of the proposed adaptive fuzzy controller very possible since it takes off a major part of the computational burden load from such types of controllers. This is very important as it allows the proposed controllers to operate in real time particularly when using the recently developed microcontrollers.

- The design and the implementation of a hierarchical structure that enables knowledge-based controllers to operate in parallel with a conventional adaptive controller while reducing the computational complexity of the overall control system. Hierarchical control strategies involving intelligent controllers have been proposed for single arm robotic systems [15], but to the best of our knowledge, no such strategy has been designed for multiple robot systems similar to the one we are proposing in this work.

- The derivation of formal proofs validating the stability of the proposed controllers. This in turn permits for the payload's position and the internal forces errors to converge to zero even in the presence of parametric and modeling uncertainties including external disturbances.

1.4 Thesis Outline

The thesis is composed of eight chapters. The introductory chapter has presented a summary of the motivations and the contributions of this work. It has also provided a brief overview of some of the main issues related to the field of multi-robot control.

Chapter 2 presents a general overview on the different modules involved in cooperative robotic systems. The chapter also provides some of the conceptual and technical difficulties that are associated with each one of them. A literature review of the different techniques proposed to tackle the different problems associated to each module and their practical limitations are also covered in this chapter. The chapter provides the reader with an idea on the level of difficulty for designing cooperative manipulator systems. Although several modules are described in the chapter, it is important to bring the reader's attention to the fact that only the motion coordination control and the disturbance rejection modules are addressed in details within the scope of this thesis.

Chapter 3 defines the problem of multiple manipulators control in a more formal way. It also provides a comparison study between centralized and decentralized control strategies, which justifies why a decentralized control approach is adopted to solve the control problem in hand. As such, a decentralized dynamic model is derived for an arbitrary number of cooperative manipulators. Based on this model, the kinematic and dynamic configurations of the manipulators used for the simulations throughout this thesis, are described.

Two main classes of conventional control algorithms that have been used in the literature of the control of cooperative robots are discussed in Chapter 4. The two classes covered in this chapter are of non-adaptive and of adaptive nature. Their advantages and practical limitations when applied to multi-robot systems are also highlighted. A practical evaluation of the tracking performance of these two control strategies is carried out through computer simulations using the robot configuration presented in Chapter 3.

Chapter 5 provides the background material on fuzzy logic based controllers and their adaptive counterparts. It also provides the reader with a general overview on the different classes of adaptive fuzzy controllers. The main type used in this work is the first-type direct adaptive fuzzy controllers. A general design strategy of such a controller is also provided. The chapter also highlights the advantages of adaptive fuzzy controllers over conventional adaptive controllers and neural controllers.

The main contributions of this research work are presented in Chapters 6 and 7. In Chapter 6, we present our first soft computing based controller that satisfies the control objectives enumerated in Section 1.3. The chapter also provides a stability analysis of the proposed controller based on a Lyapunov's stability approach. Simulation results are presented to show the tracking ability and the robustness of the controller in the face of different intensity levels of parametric and modeling uncertainties including external disturbances. The tracking performance and the robustness of the controller is compared to its conventional adaptive counterpart.

A more enhanced controller is proposed in Chapter 7. Its innovative structure takes advantage of the mathematical structure of the manipulators dynamics to drastically reduce the number of fuzzy rules used by the controller described in Chapter 6. This rule reduction technique leads to an even higher computational efficiency as compared to the controller described in Chapter 6. To reduce the computational complexity even further, a hierarchical framework is designed to make the adaptive fuzzy controller operate at a supervisory level whereas a conventional adaptive controller operate within the closed loop of the control system. This hierarchical control structure enables the adaptive fuzzy controller to operate at a lower bandwidth than that of the conventional one located at the lower level, and hence consumes less of the system's computational resources than the controller presented in Chapter 6. The stability of this hybrid controller in the face of parametric and modeling uncertainties is also proven using a Lyapunov's stability approach. Simulation results are also illustrated to show the tracking performance of the hybrid controller and compare it with the conventional one.

Chapter 8 presents a summary of this work along with some concluding remarks and suggestions for further studies pertaining to this important yet complex problem.

Chapter 2

Overview on Cooperative Robotic Systems

2.1 Introduction

Multiple cooperating robot (CR) arms sharing the same workspace have been often classified into two major categories: interacting and non-interacting systems. When cooperating arms have no direct physical interaction, the system is said to be *weakly coupled*. In such a case, the motion coordination problem becomes that of determining a collision-free path [21,44,73,120]. The present work pertains mainly to systems with strong direct physical interacting manipulators. Such systems are referred to as *strongly coupled* systems [122]. Controlling the internal forces has been a key issue for coordinating the manipulators of this type.

When two or more robot arms grasp a common object, with the ground, they form a closed kinematic chain mechanism. Such systems are referred to as *closed-*

chain robotic systems. This implicitly imposes a set of kinematic and dynamic constraints on the position and velocity of the manipulators as they have to remain in contact with the object while in motion. As a result, the degrees of freedom of the whole system decrease leading to the generation of internal forces. Such internal forces, which have to be controlled appropriately, stem from the direct interaction between the end-effectors and the object. Hence, controlling multiple manipulators interacting with an object, or a given environment, is usually a much more complex problem than that of a single robot control. The motion of the manipulators has to be kinematically and dynamically coordinated. The kinematic coordination means that all the end-effectors involved have to move synchronously to track a certain prespecified desired position and orientation of the manipulated object without losing contact with it. In this work, the dynamic coordination translates to having the end-effectors move in a certain manner so as to control the internal forces induced in the system.

In dealing with the overall operation of a typical cooperative robotic structure, several modules may have to operate in a synchronized and well coordinated manner. These modules, which might be structured in the same way as in figure 2.1, may include a vision component, a gripper component, and a motion component, to name a few. We restrict our focus in this thesis to the motion component which involves position/force control aspects. Other components such as the vision and gripper modules have been tackled elsewhere and do not show major differences with the well established research work for one single manipulator systems.

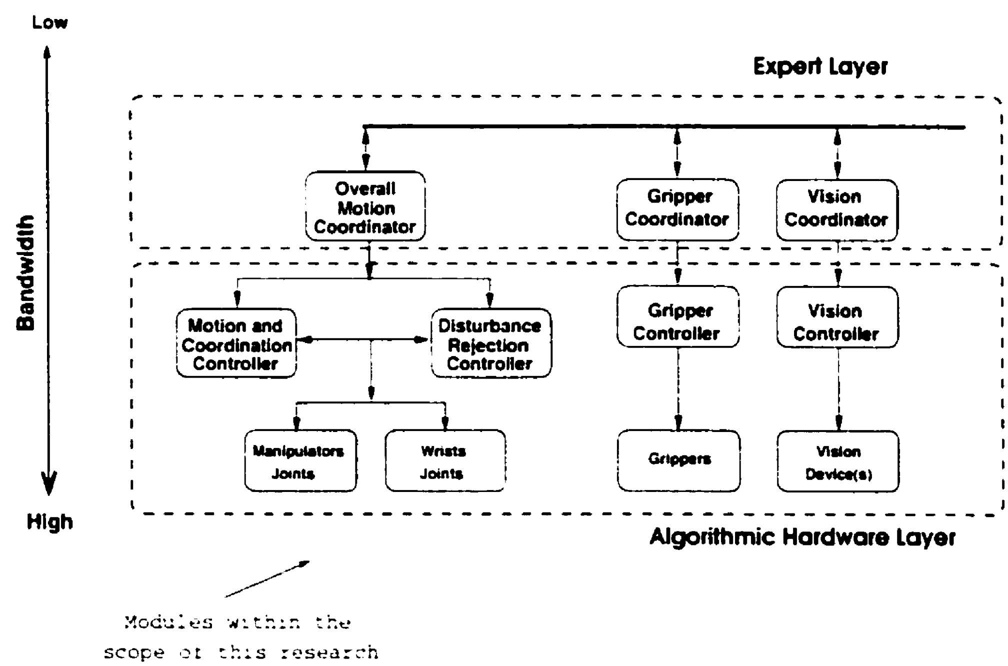


Figure 2.1: Modules Tackled Within the Scope of this Research.

2.2 Kinematics of CR

About two decades ago, Fujii and Kurono [30] presented a work about two cooperating robots. Their work was relatively simplistic and it consisted of studying the position control of the two robot end-effectors without taking the coupling effect between them into account. Hemami [41], then came up with what he called the *master-slave* method. The method involved controlling the kinematics of a certain arm, called the master arm, whereas the kinematics of the other (slave) arm is determined relatively to that of the master arm. The basic idea in solving for the kinematics problem in the master-slave configuration is to determine a transformation matrix H that transforms the position and orientation of an object, defined in a reference coordinate system (usually taken as the base frame of the master robot), into the equivalent coordinates in the slave robot's base frame. The position and orientation of the object in the slave robot's base frame, T_s, is then determined by

$$T_s = H\,T_m, \tag{2.1}$$

where T_m is the matrix defining the object's position and orientation in the reference coordinate system.

When designing a cooperative robotic system, the problem of minimizing the relative error between the robots' end-effectors has to be addressed. Luh and Zhang [74] derived a technique which uses a set of equations describing the coupling between two cooperative robots. Zheng and Luh [151] solved the kinematics for the joint torques of two cooperating arms in the context of master-slave dual-arm coordination. Suh and Shin [106] extended the work along this direction by developing the mathematical model for the coordination of dual arms carrying a rigid object. The model suggested, forces one robot to grasp the object rigidly from one end while allowing the second robot to change its contact point with the object from the other hand. The kinematics constraint relationships between two cooperative robots were also studied by Yan and Koivo [138]. Their approach was based on the joint velocities coupling constraint equations. It basically consists of deriving Jacobian matrices for a multiple manipulator system, which map the global joint space velocity into the task space gravity velocity of the object.

2.3 Dynamics of CR

Dynamics of closed chain cooperative multi-robot systems were first introduced by Hemami et al. [42]. They investigated dynamic systems which are linearized about a working point, and in which forces constraints are used in the linear feedback. Tarn et al. [111] proposed a linear transformation method that transforms the non-linear dynamic equations of the system to a linear one with decoupled matrix-form equations. Another approach to solve the dynamics of cooperative manipulators, was also proposed and involved both external and internal forces [82]. This is done by solving an internal force optimization problem. A more generalized technique was invented by Chiacchio et al. [10]. This technique uses dynamic manipulability ellipsoids which define and analyze multi-arm system configurations. For a two-arm closed chain system, the joint variables and the generalized coordinates can be represented, respectively, as

$$\begin{aligned} \theta &= [\theta_1, \theta_2, \ldots, \theta_n]^T, \quad \text{and} \\ q &= [q_1, q_2, \ldots, q_m]^T = [\theta_1, \theta_2, \ldots, \theta_m]^T, \end{aligned}$$

where n and m are the total number of joints and the degree of freedom of the chain, respectively. Note here that $m \leq n$. The Lagrangian of the system is expressed as

$$L(\theta, \dot{\theta}) = K(\theta, \dot{\theta}) - V(\theta),$$

where K and V are the system's kinetic and potential energies, respectively. The generalized torque/force vector $\tau = [\tau_1, \tau_2, \ldots, \tau_m]^T$ is then given by

$$\tau_i = \frac{d}{dt}\frac{\partial L(\theta, \dot{\theta})}{\partial \dot{q}_i} - \frac{\partial L(\theta, \dot{\theta})}{\partial q_i}, \qquad i = 1, 2, \ldots, m. \tag{2.2}$$

Using the functional relation $\theta = \Theta(q)$ and equation (2.2), the dynamic model of the two-arm closed chain can be written as

$$D(q)\,\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau, \tag{2.3}$$

where $D(q)$ is a symmetric positive definite matrix representing the manipulators inertias, $C(q,\dot{q})$ is a matrix containing the centrifugal and Coriolis terms of the robots, and $G(q)$ is a matrix representing the gravitational terms. The dynamics equation (2.3) is also valid for a single-manipulator robot except that in that case $\theta = q$ and $m = n$. It is important to point out that in real-world applications, equation (2.3) is very complex and it is very difficult to come up with an accurate estimation for it. This is mainly due to the ubiquitous presence of highly nonlinear terms in them and also due to the ill-defined working environment of the robots.

2.4 Motion Coordination Control

It was stated in the 80's that an efficient design for multi-robot systems does not only require an efficient control methodology but also the design of a very accurate control model that encapsulates the dynamics of the robots involved. The control strategy used is very important as it controls the joint torques of the robots, and hence has a major contribution to the speed and the accuracy of the overall system. Thus, the main issue is how to solve this control problem.

Due to the highly nonlinear dynamic equations and couplings that govern the motion of the robots, this problem has been quite difficult to solve. Following the emergence of the first cooperating manipulator models in [30], the control problem of such systems has drawn the attention of several researchers and different approaches have been proposed. Some are based on the linearization of the system model [52, 133]. Others use general decoupling techniques, leading to an arbitrary pole-placement approach [25–27]. These techniques were among the first to provide a direct nonlinear treatment of the equations of motion. They led to a design method based on the complete decoupling of all the variables of motion [29]. Such a design needs a solid knowledge of the model parameters, which is extremely hard to come by in many real world applications. Moreover, such techniques require a large computational resources and ignore the interacting forces. These difficulties have made these techniques inapplicable for industrial real-time applications. To alleviate these problems, the master-slave architecture was introduced [1, 74]. The general approach behind it, is to specify the desired path of the master manipulator first. The desired paths of the slave manipulators are then expressed with respect to the path of the master manipulator. Its motion coordination strategy helped reducing the complexity of the problem, but still many dynamical related difficulties remained unresolved.

Different control approaches were tested on the master-slave cooperating arms architecture. Alford *et al.* [1] used a two-robot coordination control computer to control the relative position error between the two end-effectors. Koivo [56] used a MIMO (Multiple Input Multiple Output) discrete-time autoregressive stochastic model with external inputs. The parameters estimation was computed recursively and on-line. The controllers used were mainly adaptive in nature. Mayorga and Wong [79] used a conceptual representation for each of the redundant robotic manipulators to formulate an inverse kinematics problem under an inexact context. The scheme also used a linear system of equations for each robot. a motion planning vector. and an original procedure for the proper perturbation of the pseudo-inverse matrix. to plan the motion of a multi-robot system. The proposed scheme was shown to. simultaneously. coordinate the robots motion on-line. and prevent singularities in a sensor-based environment. These properties qualify the procedure to be suitable for a large class of robotic applications. such as autonomous and telerobotic systems. Tarn et al. [111] proposed a linear transformation method that transforms the nonlinear dynamic equations of the system to a linear one with decoupled matrix-form equations. A nonlinear feedback controller was proposed in [144]. and a controller based on a PD plus gravity compensation was discussed in [131]. The main drawback of feedback-linearization-based control schemes along with most non-adaptive control schemes for the coordinated control of multiple manipulator systems. is that they usually assume a full knowledge of the system's dynamics [75. 76. 142. 143]. This is an unrealistic assumption in most cases since these complex systems are usually subject to the ubiquitous presence of uncertainties.

To deal with the uncertainty problem in controlling robotic models and specifically cooperative-robotic systems. several adaptive control schemes were proposed [47. 70. 84. 104. 105. 125. 126. 139]. These control algorithms approximate the system's dynamics using a continuous online estimation of a set of the plant's physical parameters through well-defined adaptation laws. For it to provide a satisfactory performance. a typical adaptive control algorithm assumes that the dynamic model is perfectly known and free of significant external (unmodeled) disturbances. In other words, the controller is only robust to parametric, or structured (also called modeled) uncertainties and to minor unstructured uncertainties. In addition, the unknown physical parameters must have a constant or slowly varying nominal values. Moreover, an explicit linear parameterization of the uncertain dynamics parameters has to exist, and even if it does, it might not be trivial especially with complex dynamic systems. Although the latter condition is guaranteed for every robotic dynamic equation, it might not be the case for many other systems. It is worth mentioning that all the aforementioned conditions are necessary but are insufficient for a satisfactory performance and stability of a conventional adaptive controller. Many adaptive control techniques suggested for multi-robotic manipulator systems, can only guarantee a bounded internal force error in the best case. Although, this error depends on the controller being used, in most cases it has a nonzero value which might be relatively large [125, 135]. It is also important to point out that the majority of the work on conventional adaptive controllers ignores the effect of unstructured uncertainties and external disturbances on the controller's performance and its stability. Modeling imperfection of complex systems, such as closed-chain robotic manipulators, is inevitable in most cases. This makes the development of a control approach for the increasingly complex cooperative manipulator systems, which is robust in the face of modeled and unmodeled uncertainties, a necessary step to keep up with the increasingly demanding design requirements of such systems.

With the recent developments made in the area of connectionist modeling, several researchers have attempted using them to achieve this goal. The neural networks that have been used in this regard can be classified into two classes. The first class consists of those that are trained off-line [22, 35, 65]. Such neural networks lack any online built-in adaptation capability to handle any changes in the system's dynamics. The second class of neural networks consists of those that possess an online learning and adaptation capability [7, 12, 16, 31, 66, 85, 135]. These neural networks have been found to be quite efficient in a number of applications. However, their efficiency heavily depends on their structures and adaptation laws which are usually obtained in a trial-and-error manner. This is a complex procedure and its complexity increases with that of the system to be controlled. This is mainly due to the difficulty in finding a meaningful physical interpretation to the parameters used by the neural network. Besides, even when they show satisfactory performances like in [31, 135], their behavior is still not well understood or interpreted. The performance of these neural network based adaptive techniques when applied to other types of problems is not guaranteed and cannot be easily quantified. In addition, the number of parameters which a neural network depend on is usually high and is proportional to the complexity of its architecture. This dramatically enlarges the parameters' tuning space and makes it even more difficult for the controller to achieve its global goal. It is worth pointing out that in many cases, like in [7, 12, 16, 66], a formal mathematical proof of the control system's stability could not be accomplished.

Fuzzy logic systems have been credited in various applications as powerful tools capable of providing robust controllers for mathematically ill-defined systems that may be subjected to structured and unstructured uncertainties [8, 15, 46, 64, 91, 105, 107, 115, 140]. However, and until recently, most practical applications of fuzzy logic control have been limited to relatively small classes of problems. This is mainly due to the lack of an efficient and systematic online adaptation mechanism that would adapt the controller at varying working conditions of the system. Fuzzy controllers used to depend heavily on the expertise of the designer who has to run the system through several off-line trial-and-error tune-up cycles before finally integrating the controller into the system. In this case, if there are any more changes in the system's dynamics or its working conditions, the whole tune-up process has to be restarted. Another major shortcoming fuzzy logic controllers used to suffer from is the lack of formal synthesis techniques that guarantee the basic requirements for fuzzy controllers' global stability. These limitations have stood as an open problem for several years until the pioneering work of Wang presented in [127, 128]. The universal approximation theorem has been another main driving force behind the increasing popularity of fuzzy logic controllers as it shows that fuzzy systems are theoretically capable of uniformly approximating any continuous real function to any degree of accuracy. This has led to the recent advances in the area of adaptive fuzzy control [8, 46, 107, 115, 140].

2.5 Force/Load Distribution

Researchers have used a wide variety of optimization techniques, based on different types of objective functions and optimization constraints, to solve the force/load distribution problem. Linear programming (LP) techniques were the first and simplest methods to be used. To overcome the problem of the large number of variables—with respect to the number of constraints—the compact dual LP technique was developed [86]. The duality principle of LP reduces the problem size by exchanging some variables with constraints. Inspite of its computational time efficiency and the ease of its real-time implementation, the compact dual LP method still suffers from the LP limitations. For instance, it cannot cope with quadratic quantities, such as: force norm and torque effort. The possible discontinuities in the solution produced, also represent an even further handicap for the LP-based techniques.

These difficiencies were solved by using nonlinear programming (NLP) based techniques. Nakamura et al. [82] minimized the force norm (objective function) considering the friction cone as the inequality constraints. This method is inefficient as it does not reduce much the problem size, and hence it requires high computational resources. An alternative for this scheme, is to minimize the energy dissipated by the cooperating robots subject to joint torque constraints, and the internal force without any joint torque constraints [150]. A slightly modified approach is to minimize the joint torque effort while constraining the maximum allowable joint torque of the robots [11]. Nahon et al. [81] introduced what is so called quadratic programming (QP) technique to minimize a quadratic criterion subject to linear inequality constraints. The method was used to solve a multi-finger force distribution problem. Due to the large number of constraints in a multi-robot system, the QP method requires a substantial computational resources, making it inefficient for such applications. This inefficiency could be alleviated using the duality principle's strong potential for reducing the problem size. Therefore, a hybrid strategical technique of QP and compact duality proved to be highly performant with quadratic constraints [61].

2.6 Object Manipulation

Three main types of tasks have been addressed in the literature of strongly-coupled multi-robot systems. The most cited one is when the cooperating multiple arms grasp a common rigid object to move it from an initial position/orientation to a desired one [69, 121]. The end-effectors are rigidly attached to the object. Tracking the desired position and orientation of the payload and the desired internal forces between the manipulators and the payload is among the main control issues in this type of tasks. The second type of tasks is when the cooperating arms grasp and manipulate objects having movable arms, such as a pair of pliers [17]. The third type of tasks involve grasping and manipulating large, possibly heavy, objects that do not have a suitable geometric shape to be grasped by an ordinary single end-effector. An instance of such objects can be a cardboard with the size of the manipulators. Designing large powerful end-effectors does not solve the problem because they would require large manipulators to support them in return. As a solution to this, the concept of enveloping grasp and manipulation is introduced [88, 89, 147]. Here, in addition to the end-effectors, the robots arms also contribute to the embracement of large objects. This technique even allows manipulating multiple objects at once. The idea behind it, mainly, consists of pushing the object from two opposite sides using open palm end-effectors [89, 145, 146]. The object may subsequently either roll or slide along the contact surface [147]. Such controllers possess high potentials for real-life applications. In addition to their capability in handling large industrial and waste disposal objects, such as cardboard boxes and barrels, they can also be used to handle industrial and space objects, such as crates and satellites.

2.7 Collision Avoidance

When several robots operate in a common workspace, at an assembly station for instance, they may become obstacles to each others causing possible harmful collisions. Collision avoidance is thus a crucial problem that has to be addressed when designing multi-robotic systems, especially when they operate in real-time. Simple collision avoidance methods were based on the fact that only one robot arm can be active at once. The other arms have to remain locked until the active arm suspends its motion [21,120]. Not only are these implementations restricted to a pre-specified working space, but also are very far from being optimal in terms of the time spent by the robot arms waiting out of the collision space. To overcome this problem, more advanced approaches were developed [44,73]. The techniques suggested were based on an extensive search for a collision free path. Due to the exhaustive nature of these methods, they were considered prohibitively time consuming using the computing facilities available at the time. Therefore, they were inapplicable to industrial real-time applications.

Freund et al. [28] and Cheung [9] presented a systematic collision avoidance design. This design requires that the hierarchical structure of the system takes into account the exact overall dynamics. The collision avoidance strategies were based on the trajectories which serve for collision detection and avoidance at the same time. Other approaches such as the use of a sphere model, straight line trajectory planning, and collision maps and time scheduling, were also investigated [63].

2.8 Disturbance Rejection

The ability of tolerating external disturbances and unmodeled uncertainties is a very important and desirable characteristic of a given control system. Although the disturbance attenuation problem has not been studied as extensively as position control or force control, for instance, several researchers suggested diverse techniques to approach it.

A disturbance observer that estimates and compensates for the external disturbances and parameter variations in robotic systems has been proposed [39, 57–59, 83]. The disturbance observer is known to provide a satisfactory performance if certain criteria are met. However, it tolerates neither modeling uncertainties nor time-varying disturbances.

Another disturbance rejection technique that has been used in position control of robots pertains on using a PI (proportional integral based) estimator to assess the robot joint acceleration, the centrifugal, centripetal, Coriolis, gravitational, and the friction forces acting on each of the robot joints [51, 97]. This type of estimators draws its popularity from its simplicity and low computational cost. Also, it does not require a very precise knowledge of the robot dynamics as it is based on an online continuous approximation of it. PI estimators showed to provide a satisfactory disturbance rejection and a fast tracking capability if proper

PI gains were used. However, determining suitable gains is a very model-dependent task. It is important to point out that the performance of PI estimators is heavily dependent on these gains. If the gains are not carefully chosen, not only does this degrade the performance of the controller but it might also incur instability of the system. Besides, the PI estimation technique has been mainly applied to position control and its behavior is not very well established in the presence of a force control module, for instance.

Adaptive control algorithms have also been used to attenuate external disturbances in robotic systems. An adaptive control with nonlinear parameter friction function was suggested in [2]. However, the proposed controller does not tolerate time-varying disturbances and does not guarantee a satisfactory transient response. To retain the advantages of, both, adaptive and robust control, Tao [110] proposed a robust adaptive controller that tolerates time-varying parameter disturbances. The proposed controller guarantees an asymptotic tracking error convergence to zero only in the case when the robot's dynamical parameters are not varying and the external disturbances are vanishing. Its transient performance can only be improved up to a certain extent as it depends on the disturbance bounds. To overcome these shortcomings, a robust adaptive controller with a satisfactory transient performance and the capability of high disturbance attenuation was designed in [113]. This controller does not take into account modeling uncertainties, though. In spite of the fact that all of the aforementioned adaptive disturbance rejection techniques assume bounded disturbances, none of them assume unknown bounds of these disturbances. Tomei [114] proposed a robust adaptive tracking controller that guarantees an asymptotic convergence of the position tracking error to zero in the face of bounded time-varying disturbances and modeling uncertainties with unknown bounds. The main drawback of this technique, though, is that it requires that these external disturbances and modeling uncertainties belong to L_2 for the tracking error to asymptotically converge to zero. Such a constraint is very difficult to satisfy in a real-life situation as one can hardly guarantee that the norm of the external disturbances vector and that of the modeling uncertainties will actually tend to zero as time tends to infinity.

Several researchers have attempted to tackle the disturbance rejection problem by using tools of computational intelligence. Neural network based controllers have been designed for this purpose [7, 12, 16, 31, 66, 85, 135]. The suggested neural controllers are provided with online learning and adaptation capabilities to estimate the unknown overall dynamics of the manipulators a part of which are the external disturbances. An innovative neural-based approach has been recently proposed by Wang and Lin [130]. Although neural controllers, in general, have been found to be quite efficient in a number of applications, their efficiency heavily depends on the structures of the neural networks used and on the adaptation laws which are usually determined by a trial-and-error process. This is a complex procedure and its complexity increases with that of the system being controlled. It is also worth pointing out that in many cases, like in [7, 12, 16, 66], a formal mathematical proof of the control system's stability could not be accomplished.

Several adaptive fuzzy controllers have also been proposed to attenuate the effect of the external disturbances and modeling uncertainties. Chang and Chen [8] designed a robust adaptive fuzzy controller for this purpose. However, the proposed controller is only applicable to a certain limited number of classes of holonomic and non-holonomic systems. Other adaptive fuzzy controllers have also been proposed to tackle this problem [107, 115, 140]. Although these controllers have generally shown a satisfactory position tracking performance in the face of external disturbances, they have not been tested when a force control module is involved. In addition, the numbers of fuzzy rules they use are relative quite large which necessitates the use of a special customized type of hardware to keep up with their high computational complexity. This makes these controllers quite expensive and their application to real-life robots very uncommon.

2.9 Conclusion

In this chapter, we overviewed some of the major issues pertaining to the kinematics, dynamics, control, and disturbance rejection of cooperative robotics. Although several modules with different roles and objectives have to operate simultaneously to achieve the overall goal of any robotic system, the work presented in this thesis tackles specifically the motion and disturbance rejection control module of cooperative robotic systems in the face of parametric and modeling uncertainties. The current work is also concerned, up to a certain extent, with the hierarchical aspect of the control of such systems. We provided some of the most recent research work carried out in this area. The next chapter deals with the kinematics and the dynamics modeling aspects of cooperative robotics. This background material is necessary to tackle the remaining chapters of the thesis.

Chapter 3

Modeling Aspects of CR

3.1 Problem Statement

The simultaneous position and force control has usually been accomplished by two well known control techniques: impedance control [45,55] and (traditional) hybrid position/force control [14,92,94,141]. Impedance control usually aims at satisfying a certain desired relationship between the position error and the force acting at the end-effector at which a mechanical impedance is sought to be achieved. The main drawback of impedance control is the assumption that the dynamics of the contact environment can be accurately expressed by a linear equation. In most cases, such a linear formulation is not only very difficult to attain but also requires a full knowledge of the contact environment's dynamics, which is usually not available. The (traditional) hybrid position/force control is based on the idea of splitting the task space into two orthogonal subspaces. One for the direction(s) in which only position control is considered, and one for the other direction(s) in which only force is controlled. The orthogonality between the two subspaces has been assumed to exist in the vast majority of the works. However, in a number of applications this might not be the case and hence such a technique cannot be used [20, 124].

Consider m cooperative manipulators holding a common object as shown in figure 3.1. The objectives of the robots is to *simultaneously*

(i) move that object so that its center of mass tracks a predefined trajectory (position and orientation), and

(ii) control the internal forces between the object and the end-effectors so that they converge to their predetermined desired values.

These objectives have to be achieved in the presence of *unknown time-varying* external disturbances and in the presence of, both, *structured* and *unstructured* uncertainties. To facilitate the formulation of the closed-chain system's dynamics, the following assumptions are considered.

Assumption 3.1.1 *The object is rigidly grasped by the end-effectors. In other words, there is no relative motion between the end-effectors and the object in order not to increase the system's degrees of freedom.*

Assumption 3.1.2 *The kinematics of each manipulator is perfectly known.*

3.2 Centralized Versus Decentralized Schemes

Cooperative manipulators control methods are generally classified into two major categories: centralized and decentralized control schemes. In the last decade, research in the area of cooperative manipulators control was mainly dominated by

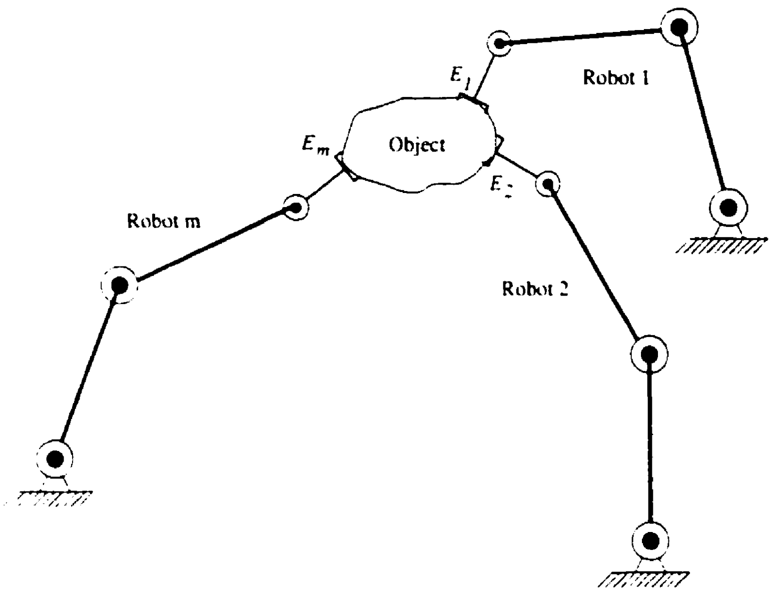


Figure 3.1: Multiple Cooperative Robots Manipulating a Common Object.

the design of centralized type of controllers. In a centralized control scheme, a coordinator is usually assigned the task of coordinating the robot manipulators. In most cases, this coordinator is either a customized piece of hardware, such as a micro processor or simply a computer system, which is responsible for computing the inputs of all the manipulators involved. This computation is mainly based on the manipulators kinematics and dynamics. As a result, using a centralized control scheme for dealing with highly nonlinear complex systems with large number of degrees of freedom, such as the case for cooperative robotic systems, may be a very much time consuming and cumbersome process. This becomes an even more tedious problem if the overall system's dynamics has uncertainties or time-varying parameters. More complexity could be added if there exist more than a few possible desired trajectories or if the desired trajectory may not be determined in advance. In such cases, the control signal cannot be computed and stored off-line but it has to be continuously computed at every hardware clock cycle in real-time throughout the whole trajectory. This may be a very computationally expensive process for large-scale complex cooperative manipulator systems.

Unlike centralized controllers, decentralized control schemes are based on the fact that each manipulator performs its own computation locally based on its own state space, dynamics, and kinematics. Additional boundary conditions and other constraints are added to assure appropriate coordination of the robot arms. Although such type of control strategy may result in some time overhead due to the possible exchange of information among the manipulators, it nevertheless drastically reduces the computational burden that centralized controllers are known for. But while the distribution of the computational tasks, control, and data, forms a 'natural' way of breaking down the overall control task in a decentralized controller, the modeling aspect is far from being trivial. As a matter of fact, decentralized control of coordinated manipulator systems is considered as a new field of research with little major results compared to the centralized control case [47,69,71,72]. A decentralized model of a general cooperative manipulator system is derived in this chapter. In other words, each robot is modeled independently of the others and its model is based solely on its own kinematics, dynamics, and state space. This is described in the following sections.

3.3 Kinematics

At any instant of time, the location of the manipulated object can be defined by the vector $x = [x_{(p)}^T, x_{(o)}^T]^T \in \mathbb{R}^{k_0}$, where $x_{(p)} \in \mathbb{R}^p$ and $x_{(o)} \in \mathbb{R}^o$ are vectors defining the position and orientation of the object, respectively. The superscript integers p and o belong to the set $\{0, 1, 2, 3\}$, where $\mathbb{R}^0$ denotes the empty vector, and k_0 ($k_0 = p + o$) is the object's total degrees of freedom. In what follows, all positions and orientation coordinates of the object and the end-effectors are expressed relative to a common reference frame unless otherwise stated. Using the manipulators forward kinematics equations, x can be rewritten as

$$x = \phi_1(q_1) = \phi_2(q_2) = \cdots = \phi_m(q_m), \tag{3.1}$$

where $q_i \in \mathbb{R}^{k_i}$ and k_i are the respective joint coordinates and the degrees of freedom of robot i, for $i = 1, \ldots, m$. The study conducted in this thesis is valid for redundant as well as non-redundant manipulators. In other words, it is valid for $k_i \geq k_0$, $i = 1, \ldots, m$. Differentiating equation (3.1) with respect to time yields

$$\dot{x} = \dot{\phi}_i(q_i) = J_{\phi_i}(q_i)\,\dot{q}_i, \quad i = 1, \ldots, m, \tag{3.2}$$

where $J_{\phi_i}(q_i) \in \mathbb{R}^{k_0 \times k_i}$ is the Jacobian matrix from the object's center of mass to q_i. Differentiating equation (3.2) with respect to time results in

$$\ddot{x} = \dot{J}_{\phi_i}(q_i)\,\dot{q}_i + J_{\phi_i}(q_i)\,\ddot{q}_i, \quad i = 1, \ldots, m. \tag{3.3}$$

Equations (3.1), (3.2) and (3.3) are known as the kinematics equations of a cooperative robotic system.

3.4 Dynamics

In a closed-chain robotic system, the dynamics of the ith manipulator in the joint space are given by

$$M_i(q_i)\ddot{q}_i + Q_i(q_i, \dot{q}_i)\dot{q}_i + W_i(q_i) - J_{e_i}^T(q_i)F_{e_i} - \tau_{d_i} = \tau_i, \tag{3.4}$$

where $\tau_i \in \mathbb{R}^{k_i}$ denotes the joint torque/force applied by the actuators on the ith manipulator. $M_i(q_i) \in \mathbb{R}^{k_i \times k_i}$ is the positive definite inertial matrix. $Q_i(q_i, \dot{q}_i) \in \mathbb{R}^{k_i \times k_i}$ is the Coriolis and centrifugal matrix. $W_i(q_i) \in \mathbb{R}^{k_i}$ represents the vector of gravitational forces. $J_{e_i}(q_i) \in \mathbb{R}^{k_0 \times k_i}$ is the manipulator Jacobian matrix from the end-effector E_i to q_i. $F_{e_i} \in \mathbb{R}^{k_0}$ is the force exerted by the object on the end-effector E_i, and $\tau_{d_i} \in \mathbb{R}^{k_i}$ represents the unstructured disturbance vector due to unmodeled friction forces and other unknown and varying factors. The dynamics of the object in the task space are given by

$$M_0(x)\ddot{x} + Q_0(x, \dot{x})\dot{x} + W_0(x) = F_0, \tag{3.5}$$

where $M_0(x) \in \mathbb{R}^{k_0 \times k_0}$ is the object's symmetric positive definite inertial matrix. $Q_0(x, \dot{x}) \in \mathbb{R}^{k_0 \times k_0}$ denotes the object's Coriolis and centrifugal matrix. $W_0(x) \in \mathbb{R}^{k_0}$ represents the gravitational force acting on the object, and $F_0 \in \mathbb{R}^{k_0}$ is the resulting force of the m manipulators acting on the object's center of mass. The force F_0 can be expressed as

$$F_0 = -\sum_{i=1}^{m} F_{ce_i},$$

where $F_{ce_i} \in \mathbb{R}^{k_0}$ is the force "indirectly" applied by the object's center of mass on the ith manipulator, and is known as the interaction force. The forces F_{ce_i} and F_{e_i}

are related by

$$F_{ce_i} = J_{ce_i}^T(x)\, F_{e_i}, \tag{3.6}$$

where $J_{ce_i}(x) \in \mathbb{R}^{k_0 \times k_0}$ is the Jacobian matrix from the end-effector E_i to the object's center of mass. Notice that the Jacobian matrices $J_{\phi_i}(q_i)$, $J_{e_i}(q_i)$, and $J_{ce_i}(x)$, are related by

$$J_{e_i}(q_i) = J_{ce_i}(x)\, J_{\phi_i}(q_i), \quad i = 1, \dots, m.$$

The force F_{ce_i} can be regarded as the sum of an internal force f_i and an external force δ_i,

$$F_{ce_i} = f_i + \delta_i. \tag{3.7}$$

From the property of internal forces, it is known that

$$\sum_{i=1}^{m} f_i = 0. \tag{3.8}$$

The internal forces cancel each other out and only external forces contribute to the motion of the object. Hence, equation (3.5) can now be rewritten as

$$M_0(x)\ddot{x} + Q_0(x,\dot{x})\dot{x} + W_0(x) = -\sum_{i=1}^{m} \delta_i. \tag{3.9}$$

The external force δ_i can be expressed as

$$\delta_i = -c_i(t)\big(M_0(x)\ddot{x} + Q_0(x,\dot{x})\dot{x} + W_0(x)\big), \tag{3.10}$$

where $t \geq 0$ denotes the time variable, and $c_i(t) \in \mathbb{R}^{k_0 \times k_0}$ is a positive definite diagonal matrix representing the load distribution of the object onto the ith manipulator. An important physical property of load distribution matrices is that they sum up to the identity matrix $I_{k_0} \in \mathbb{R}^{k_0 \times k_0}$. That is,

$$\sum_{i=1}^{m} c_i(t) = I_{k_0}, \quad \text{for } t \geq 0.$$

Substituting equation (3.9) into (3.10) results in

$$\delta_i = c_i(t) \sum_{j=1}^{m} \delta_j. \tag{3.11}$$

Substituting equation (3.11) into (3.7) yields

$$f_i = F_{ce_i} - c_i(t) \sum_{j=1}^{m} \delta_j.$$

Using equations (3.4), (3.6), (3.7), (3.11), and the kinematics equations (3.1), (3.2), and (3.3), the dynamics equation of the ith manipulator can be rewritten as

$$\tau_i = D_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + G_i(q_i) - J_{\phi_i}^T(q_i) f_i - \tau_{d_i}, \tag{3.12}$$

where

$$\begin{aligned} D_i(q_i) &= M_i(q_i) + c_i(t)\, M_0(q_i) \\ C_i(q_i, \dot{q}_i) &= Q_i(q_i, \dot{q}_i) + c_i(t)\left(\dot{M}_0(q_i) + Q_0(q_i, \dot{q}_i)\right) \\ G_i(q_i) &= W_i(q_i) + c_i(t)\, W_0(q_i) \end{aligned}$$

and

$$\begin{aligned} M_0(q_i) &= J_{\phi_i}^T(q_i)\, M_0(x)\, J_{\phi_i}(q_i) \\ Q_0(q_i, \dot{q}_i) &= J_{\phi_i}^T(q_i)\, Q_0(x, \dot{x})\, J_{\phi_i}(q_i) \\ W_0(q_i) &= J_{\phi_i}^T(q_i)\, W_0(x). \end{aligned}$$

The matrix $\{2C_i(q_i, \dot{q}_i) - (\dot{D}_i(q_i) - \dot{c}_i(t)\, Q_0(q_i, \dot{q}_i))\}$ is a skew symmetric matrix, which means that

$$\forall\, r \in \mathbb{R}^{k_i},\; r^T \left\{2C_i(q_i, \dot{q}_i) - \left(\dot{D}_i(q_i) - \dot{c}_i(t)\, Q_0(q_i, \dot{q}_i)\right)\right\} r = 0. \tag{3.13}$$

Assumption 3.4.1 *The matrix $(\dot{c}_i(t) Q_0(q_i, \dot{q}_i))$ is uniformly continuous and bounded. Thus, there exists a positive constant η_i such that*

$$\|\frac{1}{2}\, \dot{c}_i(t)\, Q_0(q_i, \dot{q}_i)\| \leq \eta_i, \quad \forall\, t \geq 0. \tag{3.14}$$

Since the matrices $D_i(q_i)$, $C_i(q_i, \dot{q}_i)$, and $G_i(q_i)$, are linear in terms of the manipulators physical parameters, the first terms of the dynamics equation (3.12) can be rewritten as

$$D_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + G_i(q_i) = Y_i(q_i, \dot{q}_i, \ddot{q}_i, c_i(t))\, \varphi_i, \quad i = 1, \ldots, m. \tag{3.15}$$

where $\varphi_i \in \mathbb{R}^{k_{\varphi_i}}$ is a vector of the physical parameters (mass, moments of inertia, friction coefficients, etc.) of the ith manipulator and the object, and k_{φ_i} is a positive integer denoting the number of such parameters. The matrix $Y_i(q_i, \dot{q}_i, \ddot{q}_i, c_i(t)) \in \mathbb{R}^{k_i \times k_{\varphi_i}}$ is known as the regression, or regressor matrix, and is independent of the physical parameters represented by φ_i.

3.5 The Model

All the simulations run throughout the thesis are carried out on two 3-DOF identical manipulators. Both manipulators cooperate to handle an object whose physical parameters are given in table 3.1. It is important to bring the reader's attention to the fact that the object's stiffness in table 3.1 is assumed to be known for the sole purpose of computing the contact and the internal forces during the simulation. In a real-life robotic system, such forces are determined through means of force sensors mounted at the manipulators wrists. The manipulators physical parameters are provided in table 3.2. The robotic model used in this paper is similar to the one used in [125]. The bases of the two manipulators are located at $(X_{1(base)}, Y_{1(base)}) = (0, 0)$ and $(X_{2(base)}, Y_{2(base)}) = (1, 0)$, respectively. In the simulations, the manipulated object moves following a straight line in the horizontal plane following a desired trajectory

$$\begin{pmatrix} x_d \\ y_d \\ \psi_d \end{pmatrix} = \begin{pmatrix} 0.4 - 0.2\, e^{(-t/6.6)} + 0.5\, e^{(-t/6.8)} \\ \frac{0.4}{\sqrt{3}} - 0.2 - \frac{0.2}{\sqrt{3}}\, e^{(-t/6.6)} + \frac{0.5}{\sqrt{3}}\, e^{(-t/6.8)} \\ \pi/3 \end{pmatrix}.$$

where ψ_d denotes the object's desired orientation in rad, $(x_d, y_d)^T$ is the desired position of its center of mass in meters (m), and t denotes the time variable in seconds (s). The object initially starts at $(0.6, (0.6/\sqrt{3}) - 0.2, \pi/3 - 0.1)^T$, that is $(0.1\ \text{m}, 0.1\ \text{m}, 0.1\ \text{rad.})^T$ off its desired initial position. The load distribution matrix is represented by $e_i(t)$:

$$e_i(t) = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}.$$

The desired internal forces are $f_{d_1} = -f_{d_2} = -10$ N acting on the line connecting the two end-effectors. It is worth mentioning that it is important to consider desired internal forces that are proportional to the dynamical parameters of the payload and the manipulator wrists. In our case, we are modeling these dynamical parameters as the stiffness k_e. In most practical cases, it is very uncommon to have systems where a high stiffness is coupled with low desired internal forces. If this does indeed occur, it may easily lead to instability in the control system since a small error in the end-effectors positions results in a very large error in the internal forces. The unstructured uncertainty and external disturbance term τ_{d_i}'s are provided by

$$\tau_{d_1} = \alpha\,(F_r(\dot{q}_1) + \rho(t) + \lambda), \text{ and} \tag{3.16}$$

$$\tau_{d_2} = -\alpha\,(F_r(\dot{q}_2) + \rho(t) + \lambda), \tag{3.17}$$

where α is a parameter in $\mathbb{R}$, and

$$F_r(\dot{q}_i) = \begin{bmatrix} 0.15 & 0 & 0 \\ 0 & 0.06 & 0 \\ 0 & 0 & 0.015 \end{bmatrix} \dot{q}_i,\ i = 1,2;\ \rho(t) = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.05 \end{bmatrix} \sin(10t);\ \lambda = \begin{bmatrix} 0.2 \\ 0.1 \\ 0.1 \end{bmatrix}.$$

The term $F_r(\dot{q}_i)$ corresponds to the modeling uncertainty associated to the friction at the joints, whereas the terms $\rho(t)$ and λ, correspond to time-varying and constant external disturbances, respectively.

Table 3.1: Values of the Object's Physical Parameters.

Object parameter	Value
m - mass	11.2 kg
d - length	0.24 m
I_c - moment of inertia	$\frac{1}{12}\, m\, d^2$
k_e - stiffness	300 N/m

The following are the matrices used in the manipulators dynamics and kinemat-

Table 3.2: Values of the manipulators Physical Parameters.

Manipulator's parameter	Value
I_j, $j = 1,2,3$ - rotational inertia of link j	$I_1 = 1.1$ kg·m², $I_2 = 0.3$ kg·m², $I_3 = 0.01$ kg·m²
m_j^*, $j = 1,2,3$ - mass of link j	$m_1^* = 30$ kg, $m_2^* = 15$ kg, $m_3^* = 5$ kg
$I_{col.}$ - rotational inertia of the inner transmission column for joint 2	$I_{col.} = 0.57$ kg·m²
l_1 and l_2 - lengths of links 1 and 2	$l_1 = 0.425$ m, $l_2 = 0.375$ m
r_2 - distance of the center of mass of link 2 from the axis of joint 2	$r_2 = 0.165$ m
VV_j, $j = 1,2,3$ - viscous friction coefficients	$VV_j = 1.2$
VS_j, $j = 1,2,3$ - Coulomb friction coefficients	$VS_j = 0.01$

ics.

$$M_i(q_i) = \begin{bmatrix} a & b\cos(q_{i2} - q_{i1}) & 0 \\ b\cos(q_{i2} - q_{i1}) & c & I_3 \\ 0 & I_3 & I_3 \end{bmatrix}; \quad W_i(q_i) = \begin{bmatrix} VS_1 \operatorname{sign}(\dot{q}_{i1}) \\ VS_2 \operatorname{sign}(\dot{q}_{i2}) \\ VS_3 \operatorname{sign}(\dot{q}_{i3}) \end{bmatrix}$$

$$Q_i(q_i, \dot{q}_i) = \begin{bmatrix} VV_1 & -b\sin(q_{i2} - q_{i1})\,\dot{q}_{i2} & 0 \\ b\sin(q_{i2} - q_{i1})\,\dot{q}_{i1} & VV_2 & 0 \\ 0 & 0 & VV_3 \end{bmatrix}$$

where $i = 1, 2$, and

$$a = I_1 + m_2^* l_1^2 + m_3^* l_1^2; \; b = m_2^* l_1 r_2 + m_3^* l_1 l_2; \; c = I_{col.} + I_2 + I_3 + m_2^* r_2^2 + m_3^* l_2^2.$$

The position and the orientation of end-effector E_i are given by the following forward kinematics equations

$$\begin{bmatrix} x_{ie} \\ y_{ie} \\ \psi_{ie} \end{bmatrix} = \begin{bmatrix} l_1 \cos q_{i1} + l_2 \cos q_{i2} + X_{i(base)} \\ l_1 \sin q_{i1} + l_2 \sin q_{i2} + Y_{i(base)} \\ q_{i1} + q_{i2} \end{bmatrix}.$$

The inverse kinematics is given by

$$\begin{aligned}
q_{11} &= \arctan\left(\frac{y_{1e}}{x_{1e}}\right) + \arccos\left(\frac{x_{1e}^2 + y_{1e}^2 + l_1^2 - l_2^2}{2l_1\sqrt{x_{1e}^2 + y_{1e}^2}}\right) \\
q_{12} &= q_{11} - \arccos\left(\frac{x_{1e}^2 + y_{1e}^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \\
q_{13} &= \psi_{1e} - q_{12} \\
q_{21} &= \pi - \arctan\left(\frac{y_{2e}}{X_{2(base)} - x_{2e}}\right) \\
&\quad - \arccos\left(\frac{(X_{2(base)} - x_{2e})^2 + y_{2e}^2 + l_1^2 - l_2^2}{2l_1\sqrt{(X_{2(base)} - x_{2e})^2 + y_{2e}^2}}\right) \\
q_{22} &= q_{21} + \arccos\left(\frac{(X_{2(base)} - x_{2e})^2 + y_{2e}^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \\
q_{23} &= \psi_{2e} - q_{22}
\end{aligned}$$

In fact, none of the controllers used in this thesis make use of the robots inverse kinematics as all the manipulators involved are controlled through computed-torque type of techniques. Inverse kinematics is only used here to determine the initial joint coordinates of the manipulators from the initial position of the object. In a practical situation, the joint coordinates are read directly from the encoders mounted at each joint.

Chapter 4

Standard Control Approaches and Their Limitations

4.1 Introduction

Nonlinear control techniques have dominated, for a number of years, the literature on the control strategies as applied to cooperative robots. This is mainly motivated by the highly nonlinear and complex structure of robotic manipulators dynamics. The control techniques of cooperative robotic manipulators proposed in the last several years have generally belonged to either one of two categories: non-adaptive and adaptive approaches. The rest of this chapter is devoted to the detailed discussion of these two classes of control strategies from a robotics perspective. Their features and limitations are also highlighted. This is important in assessing their abilities and suggesting remedies.

4.2 Non-Adaptive Strategies

There are three classes of conventional nonlinear control techniques discussed in the literature: *gain-scheduling*. *feedback linearization*. and *robust control*. Gain-scheduling is based on the idea of formulating a linear time-invariant approximation to the plant dynamics and then design a linear controller for each pre-determined linearized plant corresponding to a set of operating points in the system operation space [98]. The compensator parameters corresponding to the plants operating at the operating points lying between the pre-defined ones are scheduled through an interpolation procedure. Inspite of its simple concept and successful performance in a number of applications. gain-scheduling suffers from several crucial shortcomings. A major drawback is the lack of strong theoretical procedures to guarantee stability when it is applied to nonlinear systems. In addition. gain-scheduling. usually, results in a high computational burden due to the necessity for computing the parameters of a large number of linear controllers. This problem becomes even more serious for highly nonlinear complex systems. These issues are certainly among the main reasons why gain-scheduling is not commonly used in the control of complex robotic structures.

Unlike gain-scheduling. feedback linearization and robust control have been the subject of intensive research in the field of robotic control over several years now. The following is a general overview of these two techniques in the context of robotic control.

4.2.1 Feedback Linearization of CR in Task Space

The main idea of the feedback linearization technique is to transform an original nonlinear model of a certain system into an equivalent linear model of a simpler form by 'canceling' the nonlinearities in the original model and imposing a desired linear relation instead. This is achieved by introducing a nonlinear control law that linearizes the nonlinear system's model after performing an appropriate change in the state space coordinates.

Although several feedback linearization methods have been proposed in the literature for the control of cooperative manipulators, the vast majority of them adopt a centralized control scheme [6, 39, 95, 111, 112]. As we discussed in Section 3.2, decentralized control schemes are generally advantageous to centralized schemes, especially when dealing with large-scale complex systems. Hence, and to be consistent with the control techniques that are discussed in later chapters, a decentralized feedback linearization algorithm is adopted here. In this section, we discuss the extension of the feedback linearization to cooperative robotic systems rather than to single robotic manipulators. We also formulate the necessary control laws in the task space rather than in the joint space as it has been common in a number of studies.

The dynamics of a robot i interacting with other robots to manipulate a common object is given by

$$\tau_i = D_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + G_i(q_i) - J_{\phi_i}^T(q_i)f_i - \tau_{d_i}, \tag{4.1}$$

where $\tau_i \in \mathbb{R}^{k_i}$ denotes the joint torque/force applied by the actuators on the ith manipulator. $D_i(q_i) \in \mathbb{R}^{k_i \times k_i}$ is the positive definite inertial matrix. $C_i(q_i, \dot{q}_i) \in \mathbb{R}^{k_i \times k_i}$ is the Coriolis and centrifugal matrix. $G_i(q_i) \in \mathbb{R}^{k_i}$ represents the vector of gravitational forces. $J_{\phi_i}(q_i) \in \mathbb{R}^{k_0 \times k_i}$ is the Jacobian matrix from the object's center of mass to q_i. $f_i \in \mathbb{R}^{k_0}$ is the internal force dynamically coupling the object with the end-effector of robot i, and $\tau_{d_i} \in \mathbb{R}^{k_i}$ represents the unstructured disturbance vector due to unmodeled friction forces and other unknown and varying factors. Note that equation (4.1) has already been derived in Section 3.4 and is the same as equation (3.12). Recall from Chapter 3 that k_i and k_0 denote the degrees of freedom of the ith manipulator and of the payload, respectively. In such a case, the feedback control law τ_i which is to be fed as the input to the ith manipulator can be set as

$$\tau_i = D_i(q_i)v_i + C_i(q_i, \dot{q}_i)\dot{q}_i + G_i(q_i) - J_{\phi_i}^T(q_i)f_i - \tau_{d_i}, \tag{4.2}$$

with v_i as a new external input to the control system of manipulator i. It is easy to verify that computed torque (4.2) linearizes and decouples the dynamics equation (4.1), and leads to

$$\ddot{q}_i = v_i.$$

Since one of the main control goals is to track a certain pre-specified trajectory of the payload, it would be more convenient to rewrite the feedback control law (4.2) in the task space rather than in the joint space. The kinematic equation (3.3) implies that $\ddot{q}_i$ can be written as

$$\ddot{q}_i = J_{\phi_i}^{+}(q_i)\left(\ddot{x} - \dot{J}_{\phi_i}(q_i)\dot{q}_i\right). \tag{4.3}$$

where $\ddot{x} \in \mathbb{R}^{k_0}$ is the payload's acceleration and $J^{+}_{\phi_i}(q_i)$ is the pseudo-inverse of $J_{\phi_i}(q_i)$. That is,

$$J^{+}_{\phi_i}(q_i) = J^{T}_{\phi_i}(q_i)\ \left(J_{\phi_i}(q_i)J^{T}_{\phi_i}(q_i)\right)^{-1}.$$

Now, substituting equation (4.3) into the dynamics equation (4.1) yields

$$\tau_i = D_i(q_i)J^{+}_{\phi_i}(q_i)\ddot{x} + [C_i(q_i,\dot{q}_i) - D_i(q_i)J^{+}_{\phi_i}(q_i)\dot{J}_{\phi_i}(q_i)]\dot{q}_i + G_i(q_i) - J^{T}_{\phi_i}(q_i)f_i - \tau_{d_i}. \quad (4.4)$$

By observing equation (4.4), it is easy to notice that by choosing a feedback control law τ_i of the form

$$\tau_i = D_i(q_i)J^{+}_{\phi_i}(q_i)v_i + [C_i(q_i,\dot{q}_i) - D_i(q_i)J^{+}_{\phi_i}(q_i)\dot{J}_{\phi_i}(q_i)]\dot{q}_i + G_i(q_i) - J^{T}_{\phi_i}(q_i)f_i - \tau_{d_i} \quad (4.5)$$

the dynamics equation (4.4) becomes fully linearized and decoupled and leads to

$$\ddot{x} = v_i. \quad (4.6)$$

It is clear from equation (4.6) that for m manipulators, all the external inputs v_i, $i = 1,\dots,m$, are identical. Let v be one common external control input for all m manipulators. That is,

$$v = v_i = \ddot{x} = r - K_p x - K_d\dot{x}, \quad \text{for } i = 1,\dots,m, \quad (4.7)$$

where $r \in \mathbb{R}^{k_0}$ is a reference input characterized by the payload's desired trajectory, and $K_p \in \mathbb{R}^{k_0\times k_0}$ and $K_d \in \mathbb{R}^{k_0\times k_0}$ are diagonal matrices representing proportional and derivative gains, respectively. Having decided on the characteristics of the payload's trajectory, i.e., x_d, $\dot{x}_d$, and $\ddot{x}_d$, the reference input r can then be chosen as

$$r = \ddot{x}_d + K_d\dot{x}_d + K_p x_d. \tag{4.8}$$

Substituting equations (4.6) and (4.8) into equation (4.7) leads to the following differential equation describing the load trajectory's tracking error

$$\ddot{\tilde{x}} + K_d\dot{\tilde{x}} + K_p\tilde{x} = 0.$$

with $\tilde{x} = x - x_d$ being the manipulated object trajectory's tracking error. Again, K_p and K_d are ideally governed by the payload's natural frequency ω_j in the payload's jth degree of freedom as

$$K_p = \mathrm{diag}(\omega_1^2, \ldots, \omega_{k_0}^2) \quad \text{and} \quad K_d = \mathrm{diag}(2\omega_1, \ldots, 2\omega_{k_0}).$$

The decentralized feedback linearization control scheme expressed in the task space is schematically depicted in figure 4.1.

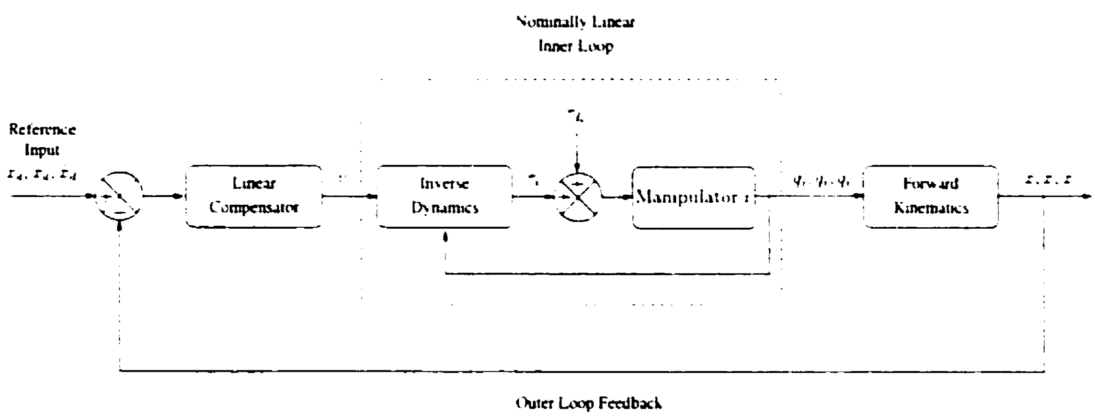


Figure 4.1: Decentralized Feedback Linearization Control of Manipulator i.

4.2.2 Robustness Issues in Feedback Linearization

As it can be seen from equations (4.4) and (4.5), the feedback linearization method is tightly dependent on the dynamic model of the robotic system and on the perfect knowledge of its parameters. These conditions are not easy to satisfy for all robotic systems due to the ubiquitous presence of parametric and modeling uncertainties in their structures, especially the dynamically complex ones. In addition, the inevitable external disturbances and the noisy measurements provided by the different types of sensors are other sources of unstructured uncertainties that may be very difficult to model. As a result, the nonlinear terms of the manipulators dynamics equations are not perfectly cancelled by the feedback control law. Hence, the presumably linear inner loop (figure 4.1) is practically still nonlinear and the joints of the manipulators are actually not decoupled as it was assumed in Section 4.2.1. In other words, the dynamics uncertainties might lead to the violation of two essential constraints on which the feedback linearization technique is based: the linearity of the inner control loop, and the decoupling of the manipulators' joints.

Let us rewrite the dynamics equation of manipulator i (equation (4.4)) as

$$\tau_i = H_i(q_i)\ddot{x} + h_i(q_i, \dot{q}_i), \tag{4.9}$$

with

$$H_i(q_i) = D_i(q_i)J^+_{\Phi_i}(q_i)$$
$$h_i(q_i, \dot{q}_i) = [C_i(q_i, \dot{q}_i) - D_i(q_i)J^+_{\Phi_i}(q_i)\dot{J}_{\Phi_i}(q_i)]\dot{q}_i + G_i(q_i) - J^T_{\Phi_i}(q_i)f_i - \tau_{d_i}.$$

In a real-life robotic system, only approximations of $H_i(q_i)$ and $h_i(q_i, \dot{q}_i)$ might be available. Let $\hat{H}_i(q_i)$ and $\hat{h}_i(q_i, \dot{q}_i)$ be the estimated values of $H_i(q_i)$ and $h_i(q_i, \dot{q}_i)$, respectively. Therefore, the feedback control law (4.5) may only be expressed in terms of those approximation values as

$$u_i = \hat{H}_i(q_i)v + \hat{h}_i(q_i, \dot{q}_i). \tag{4.10}$$

Equating equations (4.9) and (4.10) yields

$$\ddot{x} = H_i^{-1}(q_i)\hat{H}_i(q_i)v + H_i^{-1}(q_i)\tilde{h}_i(q_i, \dot{q}_i) = v + Ev + H_i^{-1}(q_i)\tilde{h}_i(q_i, \dot{q}_i),$$

where

$$\tilde{h}_i(q_i, \dot{q}_i) = \hat{h}_i(q_i, \dot{q}_i) - h_i(q_i, \dot{q}_i) \; ; \quad E = H_i^{-1}(q_i)\hat{H}_i(q_i) - I,$$

and I is an identity matrix with a proper dimension. Setting v as defined by equations (4.7) and (4.8) leads to

$$\ddot{\tilde{x}} + K_d\dot{\tilde{x}} + K_p\tilde{x} = \eta, \tag{4.11}$$

where η is known as the *uncertainty term*, and is given by

$$\eta = Ev + H_i^{-1}(q_i)\tilde{h}_i(q_i, \dot{q}_i) = E(\ddot{x}_d - K_d\dot{\tilde{x}} - K_p\tilde{x}) + H_i^{-1}(q_i)\tilde{h}_i(q_i, \dot{q}_i). \tag{4.12}$$

It is clear from equation (4.11) that the nonzero nonlinear uncertainty term η, not only has a negative effect on the tracking performance of the control system, but it may also induce instability. It might seem, from equation (4.11), that increasing the control gains, K_p and K_d, would enhance the control system's stability, however this is not true as equation (4.12) clearly shows that increasing those gains leads to the increase of the uncertainty term η as well.

To overcome the negative effects of the uncertainty in the feedback linearization algorithm, several robust control techniques have been proposed in the literature. One of these techniques, for instance, is to introduce an integration gain in the linear controller [15, 39, 80, 134]. One of the major problems with this technique is that the value of the integration gain is very critical and is, again, dependent on the amount of uncertainty in the system. A small value of this gain may not be sufficient enough to overcome the uncertainty effect on the tracking ability of the control system, whereas a large value may easily drive the system to instability. Another robust tracking approach that has been also used along with the feedback linearization of robotic manipulators is to use a variable structure controller instead of a the linear PID controller, such as in [6]. However, the robustness of such a controller in the face of different types of uncertainties still has not been formally proven.

Another major shortcoming of the feedback linearization technique when applied to cooperative robots is that it has a very poor control on the internal forces. As a matter of fact, none of the control laws discussed in Section 4.2.1 involves the internal forces. Adopting such a control strategy implicitly implies that the internal forces are assumed to be indirectly controlled through the position control hoping that a bounded position error implies a bounded internal force error [6]. This strategy is obviously inefficient as a small position tracking error may be translated into a high internal force error which may cause a serious damage to the manipulated object and to the manipulators wrists. An attempt to alleviate this problem has been proposed in [112]. The suggested technique is based on a feedback linearization that uses a weighted sum of the position and the internal forces instead of just the payload's position. As one may expect, this means that the controller has to sacrifice position tracking precision to that of the internal forces. Besides, the performance of the designed controller is very dependent to the type of task that is to be accomplished by the multi-robot system. In addition such a performance has not been studied in the presence of any kind of uncertainties.

4.2.3 Results and Practical Limitations

In order to assess the performance of the feedback linearization control algorithm when applied to cooperative manipulators along with a PD controller as a robustifying controller at the outer control loop, two experiments are carried out on the model described in Section 3.5. Since it is a common practice to use PD controllers with step inputs, the desired trajectory of the payload's center of mass is divided into 10 equally-spaced step-segments. The desired trajectory of the payload's orientation is not fragmented though as the payload is supposed to maintain a constant orientation throughout the entire trajectory. The PD controllers used for the two manipulators are identical with

$$K_p = \mathrm{diag}(\omega_1^2, \omega_2^2, \omega_3^2) \; ; \quad K_d = \mathrm{diag}(2\omega_1, 2\omega_2, 2\omega_3) \; ; \quad \omega_j = 3.6, \quad \text{for } j = 1, \ldots, 3.$$

In the first experiment, the manipulators dynamics are assumed to be perfectly known. In other words, the uncertainty term, η is assumed to be zero. The tracking performance of the resulting controller and the computed torques of the joints of the two manipulators are shown in figures 4.2 and 4.3, respectively. The tracking errors of the position of the payload's center of mass converge to zero after each excitation cycle. The tracking performance of the payload's orientation also decays to zero. However, it is clear that even in the case of a perfect knowledge of the system's dynamics, the controller becomes increasingly inappropriate for tracking the internal forces.

In the second experiment, modeling uncertainties are introduced. The matrices $\hat{D}_i(q_i)$, $\hat{C}_i(q_i, \dot{q}_i)$, and $\hat{G}_i(q_i)$, which are the estimates of $D_i(q_i)$, $C_i(q_i, \dot{q}_i)$, and $G_i(q_i)$, respectively, are assumed to have 101% of their nominal values. In other words,

$$\hat{D}_i(q_i) = 1.01\, D_i(q_i)\ ,\quad \hat{C}_i(q_i, \dot{q}_i) = 1.01\, C_i(q_i, \dot{q}_i)\ ,\quad \hat{G}_i(q_i) = 1.01\, G_i(q_i).$$

The tracking performance of the controller and the computed torques of the joints of the two manipulators are shown in figures 4.4 and 4.5, respectively. Figure 4.4 shows that the controller's tracking errors slowly increase after each excitation cycle until it becomes unstable towards the end of the trajectory.

It can be noticed from figures 4.2 and 4.4 that in the case of a stable controller the position tracking errors jump to certain peaks at the beginning of each excitation cycle as the step reference input of the object's center of mass is just an approximation of the smooth desired trajectory. A closer approximation of the desired trajectory requires a large number of step-excitations with shorter excitation periods. This means that higher gains are to be used to bring the control response to its steady-state value before the beginning of the next excitation cycle as it is the case in [112], for instance. However, it is important to point out that the use of high gains in control might be risky in real-life situations as they tend to magnify the existing parametric and modeling uncertainties as well as noise coming from the different sensors used by the robots, such as force sensors[1] for instance. This may increase the internal force tracking error even further.

The obtained results confirm that feedback linearization is certainly inadequate for the control of cooperative manipulators. Not only does it lack the potential to track, both, the payload's position/orientation and the internal forces, simultaneously, but it also has a severe sensitivity to dynamics uncertainties.

[1] It is commonly known that the force readings provided by the force sensors are extremely noisy.

Actual Vs. Desired Tracking Errors

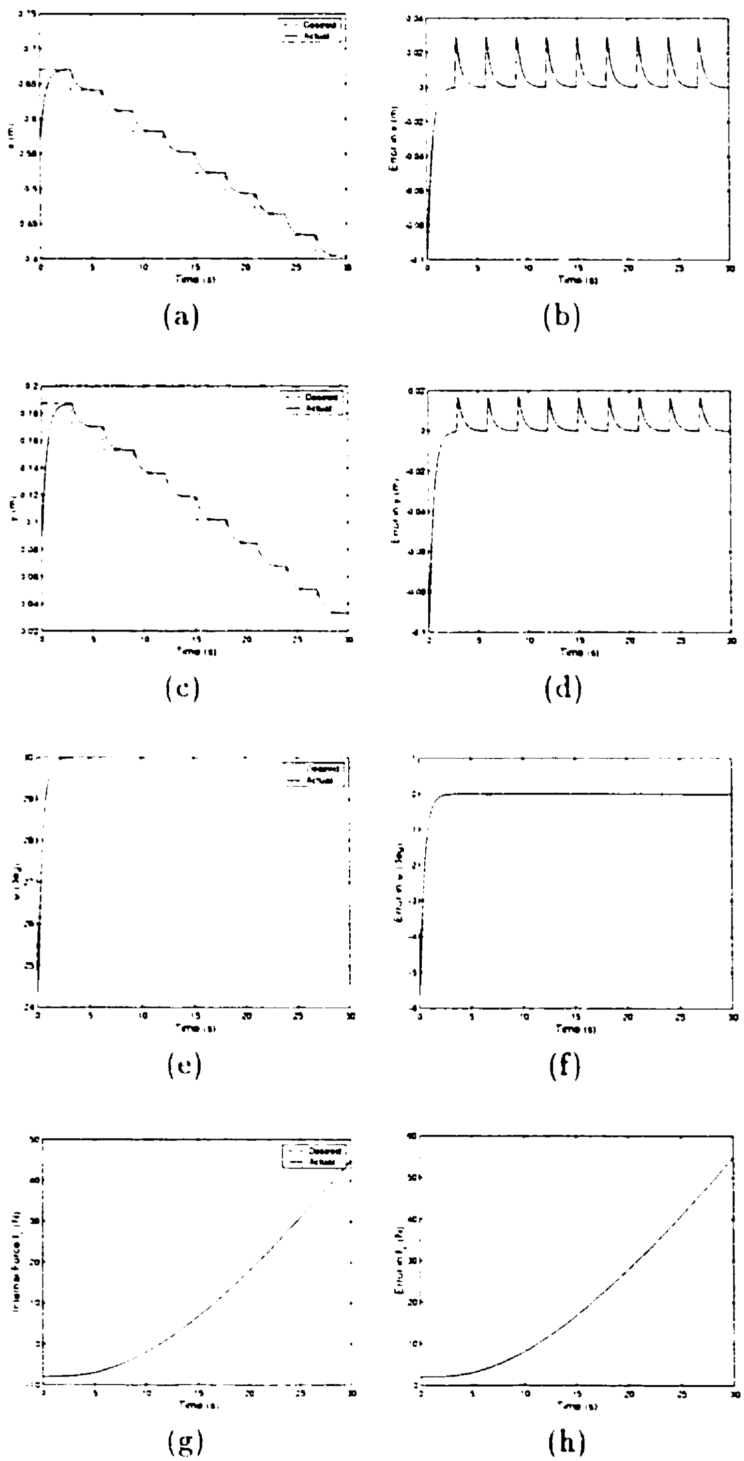


Figure 4.2: Tracking Performance of the Feedback Linearization-Based Controller Assuming Perfect Knowledge of the Robots Dynamics (No Uncertainties.)

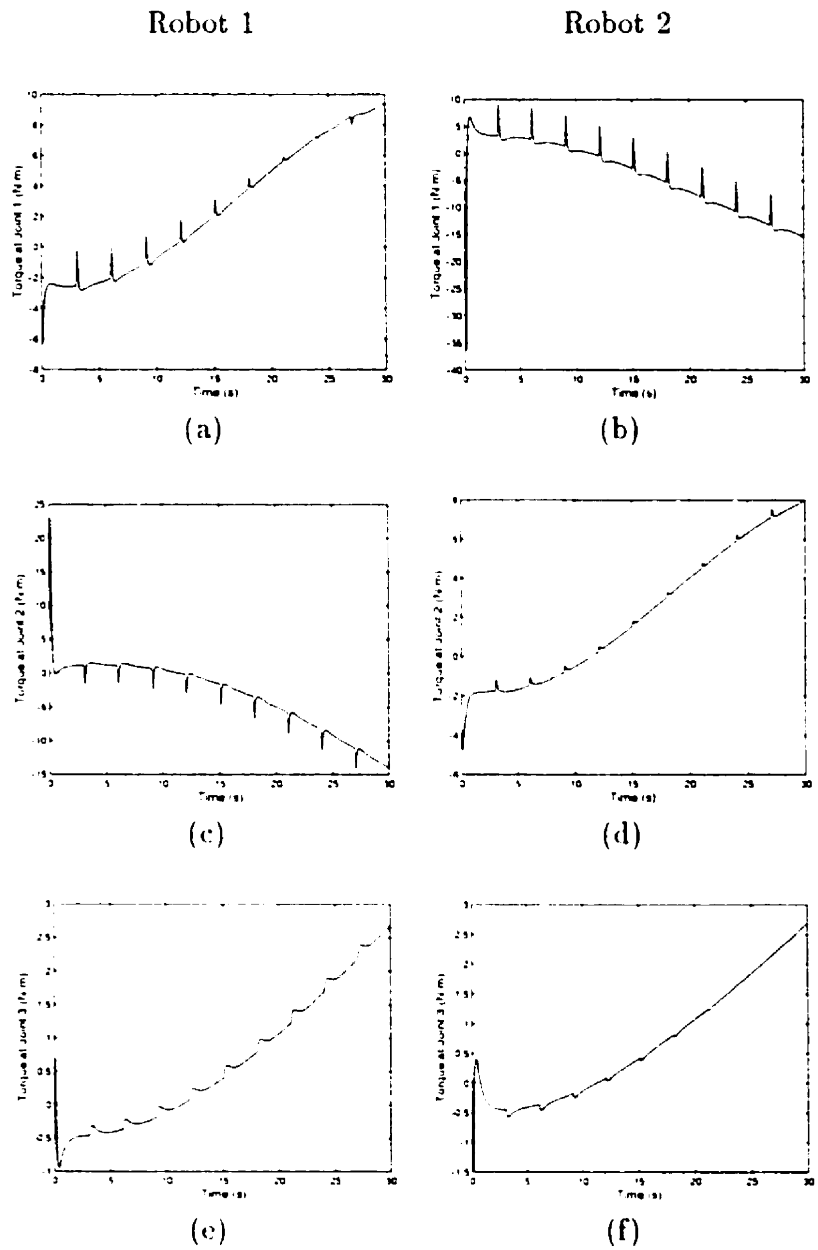


Figure 4.3: Computed Torques of the Feedback Linearization-Based Controller Assuming Perfect Knowledge of the Robots Dynamics (No Uncertainties.)

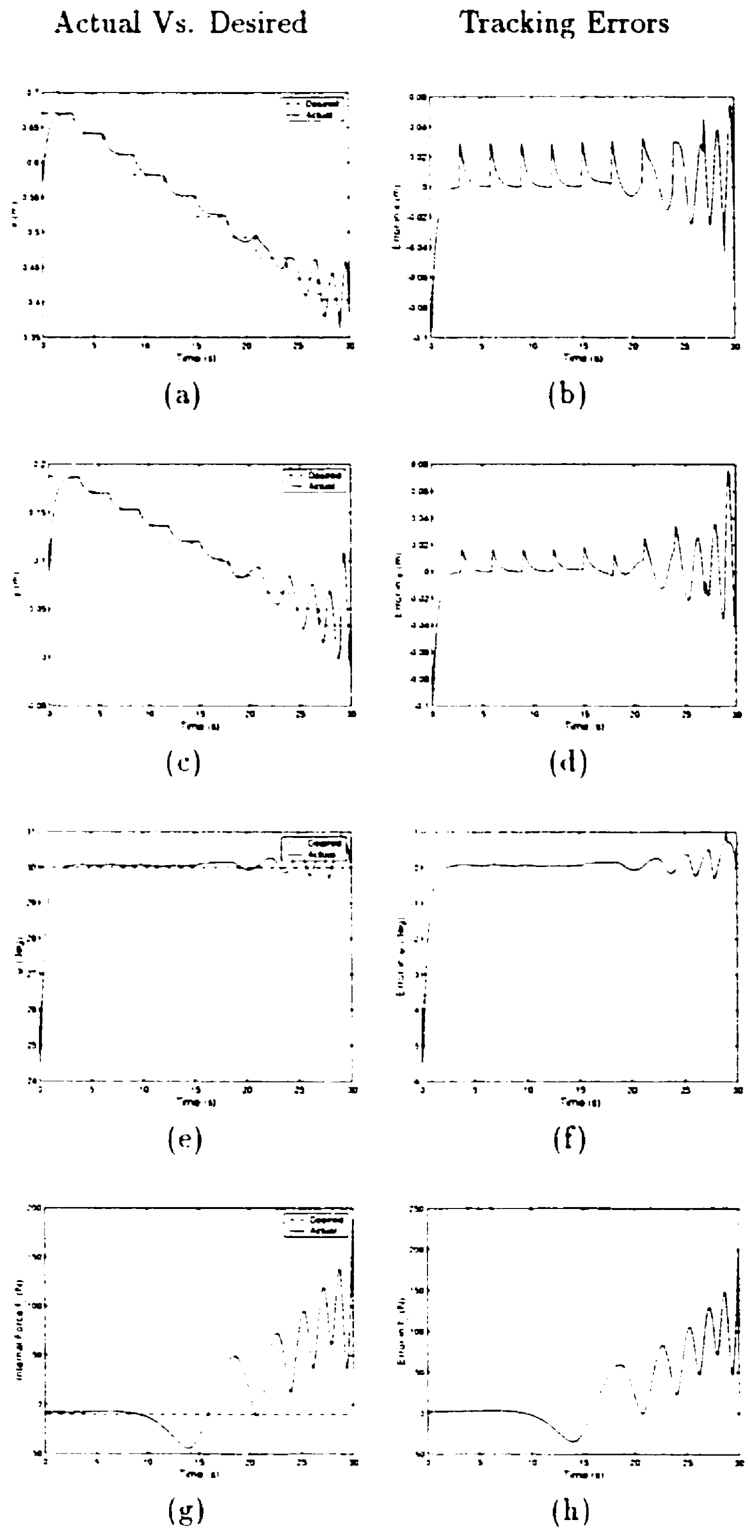


Figure 4.4: Tracking Performance of the Feedback Linearization-Based Controller in the Presence of Uncertainties.

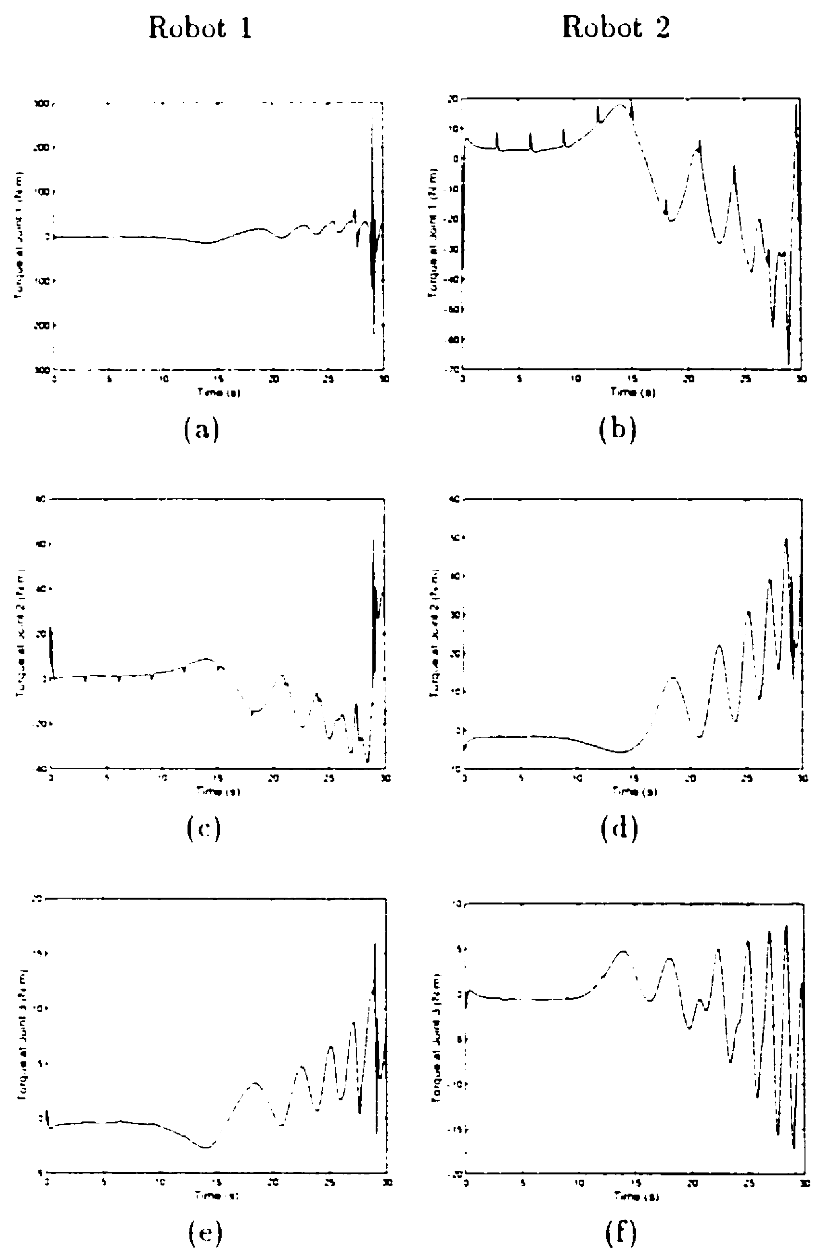


Figure 4.5: Computed Torques of the Feedback Linearization-Based Controller in the Presence of Uncertainties.

4.3 Adaptive Approach

In general, most non-adaptive control techniques assume perfect, or almost perfect knowledge of the system to control [75,76,142,143]. Such an assumption is obviously unrealistic in a large number of cases especially in those where the plant's dynamics is quite complex and where the parameters are time-varying. In addition, modeling uncertainties may have a dramatic effect on such control techniques as they may even induce instability in some cases. To overcome the problem of parameter uncertainties (also called modeled or structured uncertainties), adaptive control was introduced. The main idea of an adaptive controller is to maintain a certain consistent performance of the control system in the face of unknown time-varying plant parameters. This goal is generally achieved through a continuous online adaptation algorithm that tries to reduce the modeled uncertainties by continuously estimating the parameters of the plant's dynamics. Adaptive control systems are usually classified according to two main criteria:

1. The first criterion is based on whether the controller's parameters are tuned directly or indirectly.

2. The second criterion is based on whether the control system is provided with a reference model or not.

Based on this, several adaptive controllers have been suggested in the literature with varying degrees of implementation success. For an excellent review on these techniques, one may consult [5,32].

4.3.1 Adaptive Controllers Design

Unlike non-adaptive control design, constructing an adaptive controller can be quite complicated and involved. In adaptive control, the plant parameters are unknown and an adaptation law has to be derived to update the controller parameters. Although there is no systematic way for designing a generic adaptive controller as it depends on the type of control system in hand, adaptive controllers design generally require three key steps:

- deciding on a control law involving the system's adjustable parameters,
- deriving an adaptation law for tuning those parameters, and
- analyzing the stability and the convergence properties of the obtained control system.

When, for instance, model-reference adaptive control [5, 32] (MRAC) systems similar to the one shown in figure 4.6 are used, the three steps are usually accomplished using an appropriate Lyapunov function and by extracting the necessary conditions for insuring the system's stability. We continue using this type of adaptive controller for the remaining part of the chapter.

4.3.2 Adaptive Control of CR

Adaptive control of cooperative manipulators still stands as one of the most challenging problems in the area of robotics research due to the high complexity of the dynamics of the closed-chain robotic systems. Over the years, several researchers have used different approaches to tackle this problem [47, 70, 84, 105, 125, 126, 139].

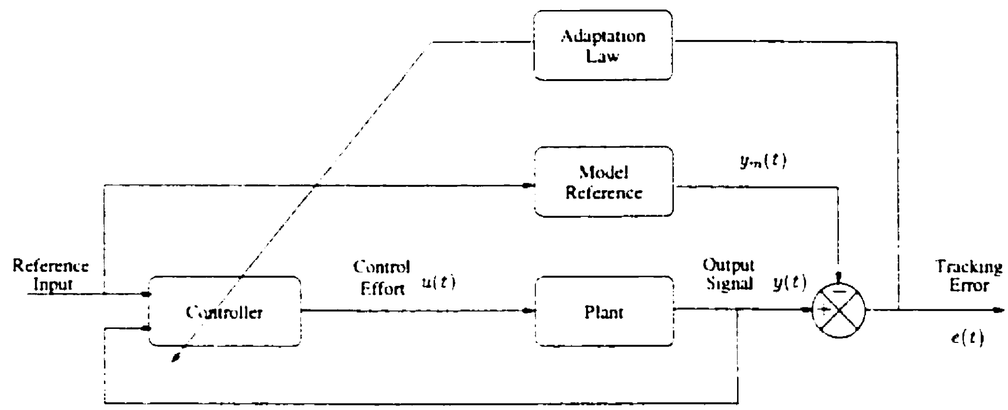


Figure 4.6: Model-Reference Adaptive Control System.

Most of the hybrid position/force controllers discussed in the literature divide the robots working space into two orthogonal subspaces in which position and force are independently controlled as they happen to be in the null space of each other. Although the subspaces orthogonality exists in a number of applications, it is not the case for strongly coupled multiple manipulator systems, and as such, the developed techniques become inapplicable. A promising technique has been recently proposed in [70]. It consists of a decentralized adaptive controller which is relatively inexpensive in terms of computational resources requirements and does not require the orthogonality condition of the end-effectors position and force subspaces. This is because each robot operates independently of the others saving the communication time overhead. The controller also provides stable adaptive tracking of the internal forces and the payload's position with fast convergence rates and without the need for decoupling the force and the position variables. This conventional adaptive controller (CAC) will be utilized later in this thesis for the design of two types of intelligent controllers and also as a benchmark to compare its performance to that of the intelligent controllers. What follows is a brief description of the CAC based on the work described in [70].

The reference joint velocity (also called nominal reference) of robot i, $i = 1, \ldots, m$, is defined by

$$\dot{q}_{r_i} = J^+_{\phi_i}(q_i)\,(\dot{x}_d - \gamma_i\,\tilde{x}), \tag{4.13}$$

where

$$J^+_{\phi_i}(q_i) = J^T_{\phi_i}(q_i)\,\left(J_{\phi_i}(q_i)\,J^T_{\phi_i}(q_i)\right)^{-1}.$$

γ_i is a positive constant, x_d is the desired position of the object, and $\tilde{x}$ is the corresponding position error.

$$\tilde{x} = x - x_d.$$

The pseudo-inverse of the Jacobian matrix $J_{\phi_i}(q_i)$ only exists if robot i is not at a singularity position. In other words, $J^+_{\phi_i}(q_i)$ exists if rank $[J_{\phi_i}(q_i)] = k_0$.

Assumption 4.3.1 *At every instant of time,*

$$rank[J_{\phi_i}(q_i)] = k_0, \quad i = 1, \ldots, m. \tag{4.14}$$

Notice how the nominal reference $\dot{q}_{r_i}$ depends solely on the information that is locally available at manipulator i and does not require any information from the other manipulators. Differentiating equation (4.13) with respect to time yields

$$\ddot{q}_{r_i} = J^+_{\phi_i}(q_i)\,(\ddot{x}_d - \gamma_i\,\dot{\tilde{x}}) + \dot{J}^+_{\phi_i}(q_i)\,(\dot{x}_d - \gamma_i\,\tilde{x}). \tag{4.15}$$

A residual error to the reference joint velocity $\dot{q}_{r_i}$ is then defined as

$$s_i = \dot{q}_i - \dot{q}_{r_i} = J^+_{\phi_i}(q_i)\,(\dot{\tilde{x}} + \gamma_i\,\tilde{x}). \tag{4.16}$$

So, the control law of robot i can be developed as

$$\tau_i = \hat{D}_i(q_i)\ddot{q}_{r_i} + \hat{C}_i(q_i,\dot{q}_i)\dot{q}_{r_i} + \hat{G}_i(q_i) - K_{s_i}s_i - J^T_{\phi_i}(q_i)(K_i\tilde{x} + f_{d_i}) - \hat{\tau}_{d_i}, \tag{4.17}$$

where $\hat{D}_i(q_i)$, $\hat{C}_i(q_i,\dot{q}_i)$, $\hat{G}_i(q_i)$, and $\hat{\tau}_{d_i}$, are the estimates of $D_i(q_i)$, $C_i(q_i,\dot{q}_i)$, $G_i(q_i)$, and τ_{d_i}, respectively; K_{s_i} and K_i are positive definite gain matrices, and f_{d_i} is the desired internal force exerted on the ith manipulator. Note that

$$J_{\phi_i}(q_i)s_i = \dot{\tilde{x}} + \gamma_i\tilde{x}. \tag{4.18}$$

Based on equation (3.15), the controller equation (4.17) can be reformulated as follows

$$\tau_i = Y_i(q_i,\dot{q}_i,\dot{q}_{r_i},\ddot{q}_{r_i},c_i(t))\,\hat{\varphi}_i - K_{s_i}s_i - J^T_{\phi_i}(q_i)(K_i\tilde{x} + f_{d_i}) - \hat{\tau}_{d_i}, \tag{4.19}$$

where

$$Y_i(q_i,\dot{q}_i,\dot{q}_{r_i},\ddot{q}_{r_i},c_i(t))\,\hat{\varphi}_i = \hat{D}_i(q_i)\ddot{q}_{r_i} + \hat{C}_i(q_i,\dot{q}_i)\dot{q}_{r_i} + \hat{G}_i(q_i), \tag{4.20}$$

and $\hat{\varphi}_i$ is the time-varying estimate of φ_i. The estimation vector $\hat{\varphi}_i$ is regularly updated at a given frequency which can either be greater than or equal to the actuators frequency. The adaptation is controlled by the following adaptive law

$$\dot{\hat{\varphi}}_i = -\Gamma_i^{-1}\,Y_i^T(q_i,\dot{q}_i,\dot{q}_{r_i},\ddot{q}_{r_i},c_i(t))s_i,$$

where $\Gamma_i \in \mathbb{R}^{k_{\varphi_i}\times k_{\varphi_i}}$ is a positive definite gain matrix. The block diagram of the CAC is shown in figure 4.7.

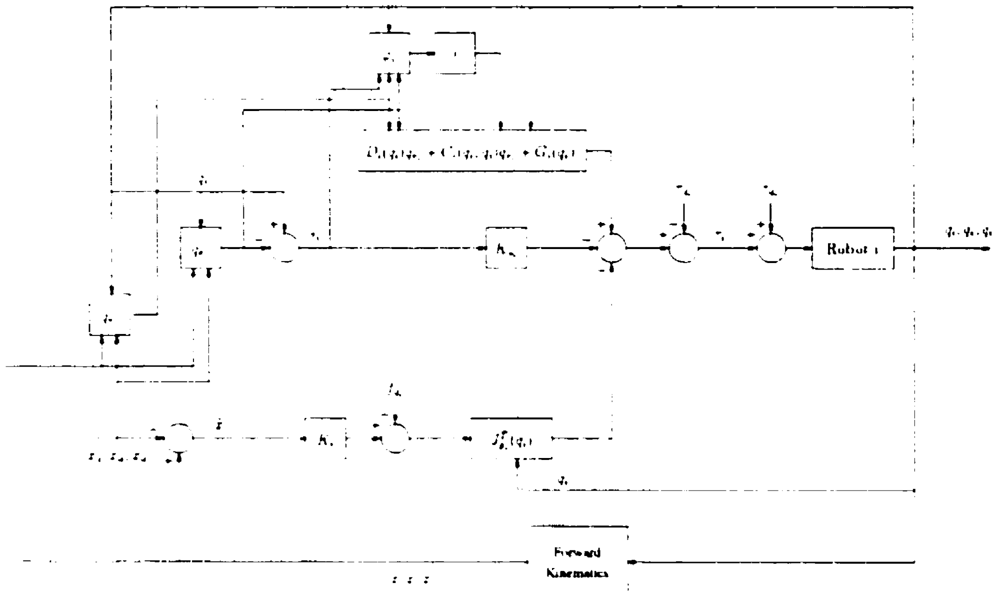


Figure 4.7: Block Diagram of the CAC.

It can be clearly seen from the control law (4.19) that the control of each manipulator depends on information that is local to that particular manipulator. No information exchange is required to or from the other robots. Each robot maintains its own physical parameters estimation and computes the object's kinematics through its own kinematics equations (3.1), (3.2), and (3.3). In addition, the controller uses a feedforward of the desired internal force f_{d_i} avoiding any force feedback that requires communication with the other manipulators.

Theorem 4.3.1 *If* $\tilde{\tau}_{d_i} \overset{\text{def}}{=} \hat{\tau}_{d_i} - \tau_{d_i} = 0$, *and the gain* K_{s_i} *is properly chosen, for* $i = 1, \ldots, m$, *then the control law (4.19) guarantees an exponentially asymptotic convergence of* $\tilde{x}$ *to zero, and only brings the internal force error* $\tilde{f}_i$ *to an exponential convergence to within the vicinity of zero.*

More details on theorem 4.3.1. as well as its proof. can be found in [70]. Although the proof does not take into account unstructured uncertainties and external disturbances. its extension to cover this case under the control law defined in (4.19) should be straight forward. It is very important to notice that a key condition for the satisfaction of theorem 4.3.1 is the existence of no external disturbance and modeling uncertainties estimation error (i.e., $\tilde{\tau}_{d_i} = 0$). In a practical working environment. this condition is almost impossible to satisfy for a complex dynamical system such as a closed-chain manipulator system. In such systems. external disturbances and modeling uncertainties may have a significant impact on the controller's performance and even on its stability. This shows the importance of taking into consideration the existence of these uncertainties in the controller design process. an aspect which is tackled in the next chapters.

4.3.3 Results and Discussion

In order to evaluate the performance of the CAC. two experiments are carried out on the cooperative manipulator system described in Section 3.5. In each one of them. the object mass is considered as the only parametric uncertainty of the CAC. This implicitly implies that the object's inertia represents another parametric uncertainty (see table 3.1.) The initial value of $\hat{\varphi}_i$. $i = 1.2$. is chosen to be zero.

The first experiment is meant to test the validity of the CAC in the case where the system's dynamics has no unstructured uncertainties or external disturbances (i.e., α is set to zero in equations (3.16) and (3.17).) In other words. the model is assumed to be subject to parametric uncertainties only. The controller's tracking performance and the resulting computed torques of the two manipulators are shown in figures 4.8 and 4.9, respectively. The CAC shows satisfactory performance in this case as all the tracking errors converge to zero after a few adaptation cycles. It is worth pointing out that in this case, both tracking errors of the payload's position/orientation and of the internal forces converge to zero. In addition, it is important to mention that by comparing figures 4.9 and 4.3 we notice that the CAC achieved this satisfactory tracking with less control efforts than those generated by the feedback linearization controller and this is true for all the six joints. As such, the CAC has already shown its superiority to the feedback linearization controller in this context.

A second experiment is carried out to study the tracking behavior of the CAC in the face of parametric and modeling uncertainties. For this purpose, an unmodeled external disturbance term is added to the manipulators dynamics in addition to the payload's mass uncertainty of the first experiment. This is achieved by setting α equal to 1 in equations (3.16) and (3.17). The controller's tracking performance and the resulting computed torques are provided in figures 4.10 and 4.11, respectively. The results clearly show how modeling uncertainties lead to a severe degradation in the controller's tracking ability as none of the tracking errors converge to zero this time. In fact, this kind of behavior is expected from conventional adaptive controllers in general as they are originally designed to compensate for parametric uncertainties only. To tackle this issue, a new class of controllers has to be designed.

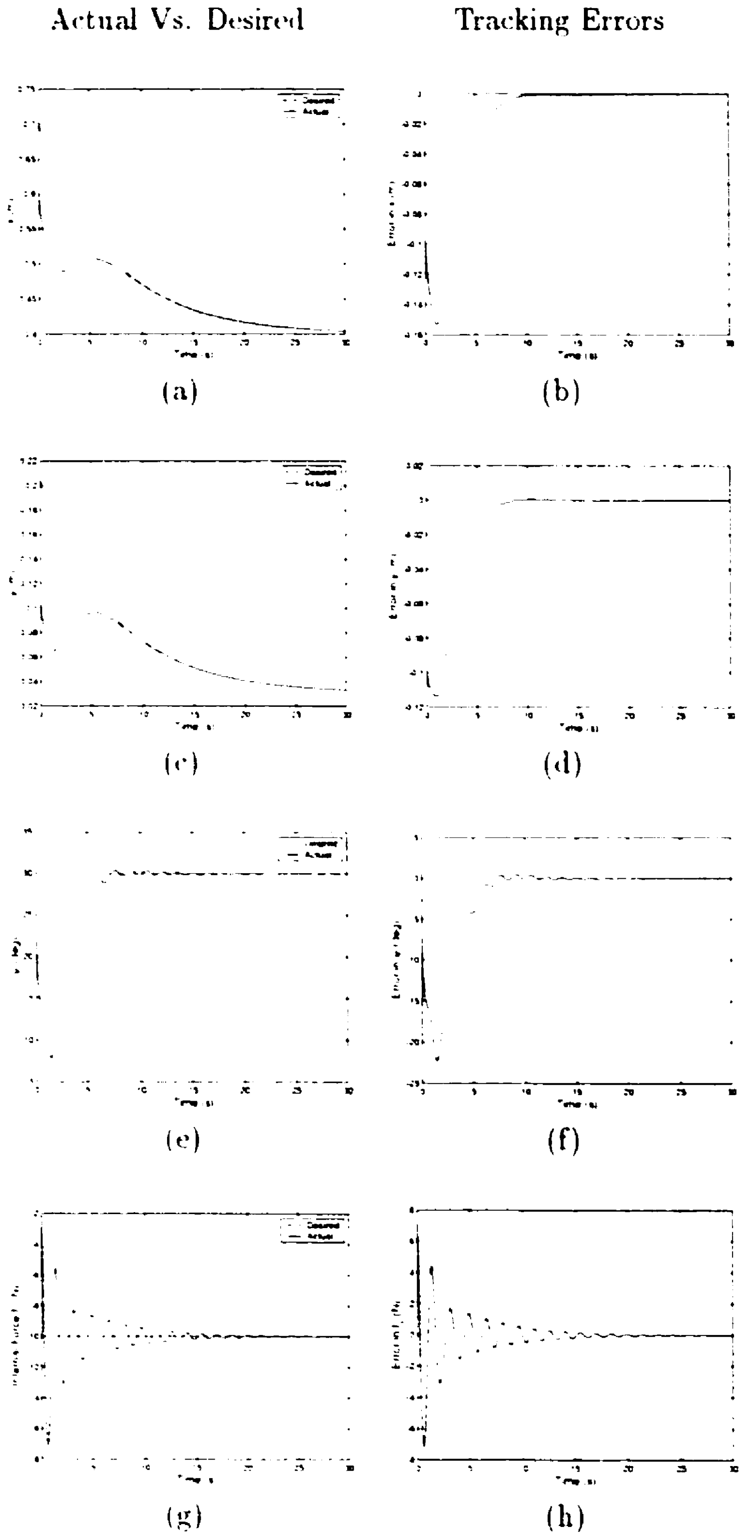


Figure 4.8: Tracking Performance of the CAC in the Presence of Parametric Uncertainties Only ($\alpha = 0$.)

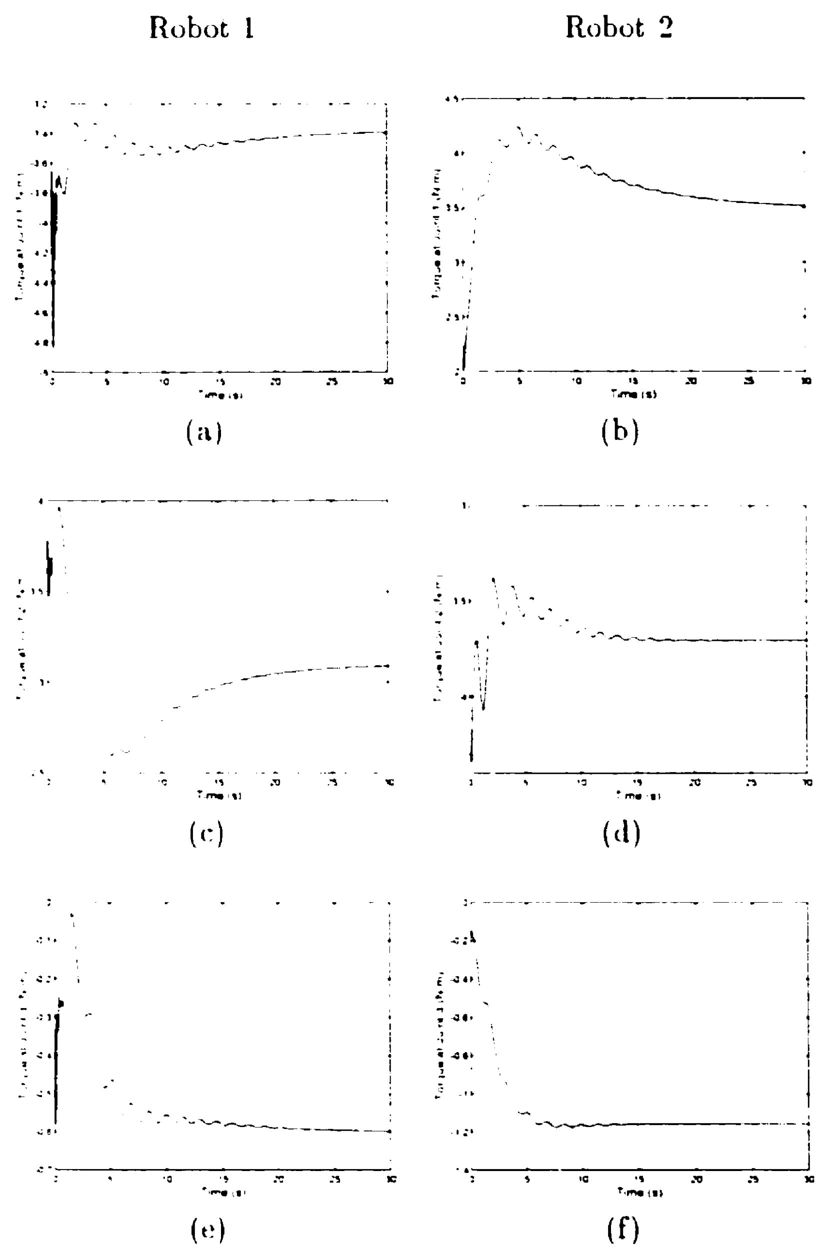


Figure 4.9: Computed Torques Using the CAC in the Presence of Parametric Uncertainties Only ($\alpha = 0$.)

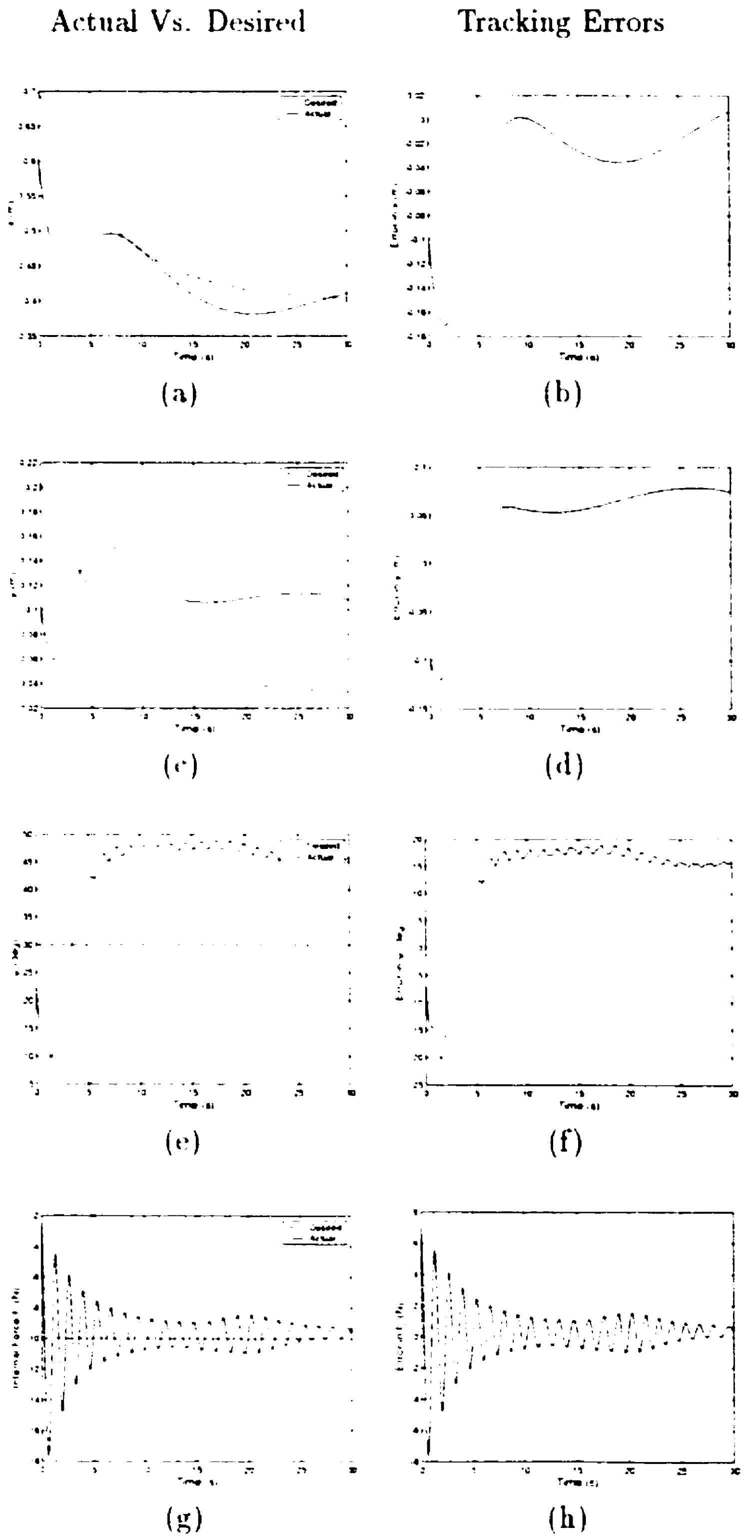


Figure 4.10: Tracking Performance of the CAC in the Presence of Parametric and Modeling Uncertainties ($\alpha = 1.$)

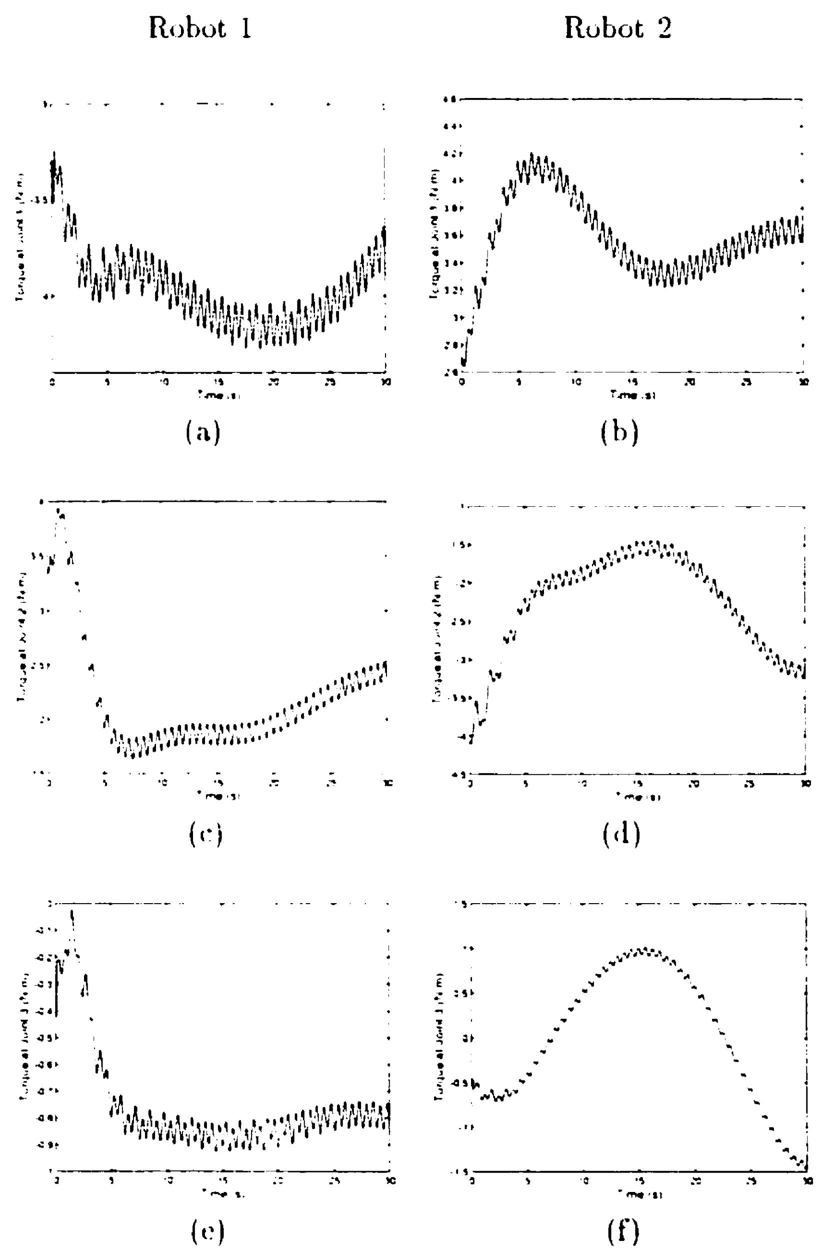


Figure 4.11: Computed Torques Using the CAC in the Presence of Parametric and Modeling Uncertainties ($\alpha = 1.$)

4.4 Conclusion

Most conventional non-adaptive control schemes for the coordinated control of multiple manipulator systems, usually assume a full knowledge of the system's dynamics [75,76,142,143]. This is an unrealistic assumption in most cases since these complex systems are usually subject to the ubiquitous presence of uncertainties. As it is shown in this chapter, these uncertainties may have a dramatic effect on the controller's performance and may even induce instability. To deal with such uncertainties, several adaptive control schemes were proposed [47,70,84,104,105,125,126,139]. These adaptive control algorithms approximate the system's dynamics using a continuous online estimation of a set of the plant's physical parameters through well-defined adaptation laws. For it to provide a satisfactory performance, a typical adaptive control algorithm assumes that the dynamic model is perfectly known and free of significant external (unmodeled) disturbances. In other words, the controller is only robust to parametric, or structured (also called modeled) uncertainties and to minor unstructured uncertainties. It is also important to point out that the majority of the work on conventional adaptive controllers ignores the effect of unstructured uncertainties and external disturbances on the controller's performance and its stability. Modeling imperfection of complex systems, such as closed-chain robotic manipulators, is inevitable in most cases. This is illustrated in this chapter by the degrading tracking performance of the CAC when modeling uncertainties are introduced. This makes the development of a robust control approach for the increasingly complex cooperative manipulator systems a necessary step to keep up with the increasingly demanding design requirements of such systems.

Chapter 5

Soft Computing Based Controllers

5.1 Introduction

The common point shared by standard computed-torque control algorithms (adaptive or non-adaptive) used in the different modules of cooperative robots is their requirement for a precise model of the system. In real life, however, cooperative robots may have very complex structures, and hence, deriving a precise model for the system might become very difficult, and even impossible, at times. In addition, it is usually a tedious process to approximate the values of the model parameters, especially when such values are continuously varying. This is mainly due to the ubiquitous presence of uncertainty in the robots environment, which is usually very difficult to precisely model or quantify. In fact, one of the main challenges of today's robotic research is to build reliable and robust robots that are able to achieve with as much autonomy as possible their goals regardless of the dynamical complexities and environmental uncertainties. Fuzzy control, and soft computing tools in general, have been credited in a number of applications to provide robust controllers by tolerating large amounts of noise in the input signals, and non-precise models of the systems with possibly time-varying parameters. In particular, fuzzy controllers have been efficiently applied in the field of robotics where

- a *precise* mathematical model is either hard or impossible to form,
- sensor data is noisy, and the parameters of the system are continuously varying, and
- real-time operation is a major requirement.

Fuzzy controllers are regarded as generic controllers in the sense that thanks to their qualitative nature, they can be transferred from one platform to another with minor modifications. When applied to robotic systems, fuzzy controllers lead to a smooth movement behavior of the robots, and to high insensitivity to errors and fluctuations in the sensors data. In addition, recent technological developments made it possible for fuzzy controllers to be an integral part of the system's hardware by embedding them in dedicated micro-chips.

But until recently, tools of soft computing have been mainly applied to mobile cooperative robots [87, 99, 103, 116, 119, 123], and only a few researchers have addressed their applications to base-fixed cooperative robot manipulators [35, 36, 46, 135]. The task is relatively more straightforward in the case of mobile cooperative robots as the whole problem boils down to finding an obstacle-free path but it is much more involved when dealing with base-fixed CR systems. From this chapter onward, we restrict our focus of soft computing tools to those dealing with fuzzy logic based systems in tackling a certain class of control problems.

5.2 Fuzzy Logic Controllers

A wide range of systems featuring complexities and imprecisions have been well understood and successfully addressed by humans without the need for well defined mathematical models. In fact humans have learned to make decisions even in the absence of clearly defined processes. This is carried out based on expertise and general knowledge acquired about the system or its alike. Some of humans' actions can be accomplished very effectively using a well-structured set of if-then rules they have developed implicitly over years of knowledge and experience. Fuzzy set theory [148] has been developed to mimic this powerful capability and to help designing systems that can deal effectively with complex processes. Fuzzy set theory has been very useful in modeling complex and imprecise systems. It has also been used very effectively in the area of control, as a decision making system [15]. In fact, its most successful practical applications have been made in the field of control systems.

Fuzzy logic is an extension of the *crisp*, also well known as the *binary*, logic. It was first introduced by *L. Zadeh* [148]. Fuzzy logic systems are rule-based expert systems that are comprised of a set of if-then (condition-action) rules. The main idea behind fuzzy controllers is to describe a given system and its actions using linguistic production rules. When a fuzzy logic engine is used as a controller within a control loop, it is called a fuzzy logic controller (FLC), or simply a fuzzy controller. In this thesis, the main focus is on multi-input multi-output (MIMO) FLCs. An

n-input m-output FLC can be regarded as a mapping from $U = U_1 \times U_2 \times \ldots \times U_n$ into $V = V_1 \times V_2 \times \ldots \times V_m$, where $U_i \subset \mathbb{R}$, $V_j \subset \mathbb{R}$, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$, are the input and output spaces, respectively. A more detailed description of fuzzy logic controllers is found in Appendix A.

5.3 Adaptive Fuzzy Controllers

As outlined in the earlier section, fuzzy logic control systems possess several features making them excellent candidates for a wide range of applications characterized by ill-defined dynamics and unstructured environments. Nevertheless, fuzzy logic controllers (FLCs) may lead to a poor performance when applied to plants with large unknown parameter and/or modeling uncertainties as they suffer from the lack of an efficient and systematic online adaptation mechanism that adapts the controller to varying working conditions of the system. Static fuzzy controllers depend heavily on the expertise of the designer who has to run the system through several off-line trial-and-error tune-up cycles before finally integrating the controller within the system. In this case, if there are changes in the system's dynamics or its working conditions, the whole tune-up process has to be restarted. Another major shortcoming of FLCs is the lack of formal synthesis techniques that guarantee the basic requirements for fuzzy controllers' global stability. These limitations have stood as an open problem for several years until the pioneer work of Wang presented in [127, 128]. In order to overcome them, a new class of FLCs has emerged. This type of fuzzy logic system is known as *adaptive fuzzy control*. An adaptive fuzzy controller (AFC) is an FLC which is equipped with an online adaptation mechanism to continuously tune up a set of the fuzzy logic system's parameters stored in the knowledge base.

Given their merits over conventional adaptive controllers, AFCs have recently drawn the attention of several researchers [8, 36, 46, 107, 115, 140]. In fact, a standard adaptive control algorithm usually assumes that the dynamic structure of the system is perfectly known and is free of any significant external (unmodeled) disturbances. In other words, the controller is only robust to parametric, or structured (also called modeled) uncertainties and to minor unstructured uncertainties. In addition, the unknown physical parameters must have a constant or slowly varying nominal values and an explicit linear parameterization of the uncertain dynamic parameters has to exist. Although the latter condition is guaranteed for every robotic structure, it might not be the case for many other systems with different dynamic properties. It is also worth mentioning that while all the aforementioned conditions are necessary, they are nevertheless insufficient for insuring a satisfactory tracking performance and stability of a conventional adaptive controller. Due to the many constraints needed to implement a conventional adaptive controller, researchers have often ignored the effect of unstructured uncertainties and external disturbances on the controller's performance and its stability. This is not very realistic since modeling imperfections of complex systems, such as closed-chain robotic manipulators, are inevitable in a large number of cases. AFCs, on the other hand, are well equipped to deal with the aforementioned shortcomings. In addition, the universal approximation theorem has been another main driving force behind the increasing popularity of AFCs as it shows that fuzzy systems are theoretically capable of uniformly approximating any continuous real function to any degree of accuracy. Moreover, AFCs 'inherits' all the advantages of FLCs such as incorporating fuzzy logic information from expert human operators. This being said, it is worth mentioning that due to possible difficulties in fully assessing the knowledge base, it might not be efficient to build the whole controller based on this partial knowledge. It nevertheless helps in providing key information on the system behavior around different working points.

5.4 Classes of AFCs

AFCs are usually classified according to two criteria:[1]

1. The first criterion pertains to the way an FLC contributes to the control system. The role of an FLC can be either determining the output control action by means of fuzzy rules with the controlled output as their consequent part, or making use of the human expertise to generally describe the system's behavior through linguistic fuzzy rules.

2. The second criterion pertains to whether the AFC is linear or nonlinear in its adjustable parameters.

[1]Different AFCs classification criteria may be found in the literature. In this thesis, we use those adopted in [128].

5.4.1 Direct and Indirect AFCs

A major classification criterion used in the conventional adaptive control literature states whether a controller is a direct or an indirect adaptive controller [85]. In a direct adaptive control scheme, the parameters of the controller are directly adjusted so as to achieve a certain prespecified control goal. This goal is usually achieved by minimizing the tracking error between the output of the actual system and that of a reference model. In an indirect adaptive control scheme, it is the plant's parameters that are continuously estimated online, and the controller's parameters are then determined accordingly.

Direct and indirect classification of AFCs follow, more or less, the same analogy as that of direct and indirect conventional adaptive controllers. In other words, direct AFCs incorporates fuzzy linguistic rules to define the control action. Such rules may take the following form in controlling a cooling system, for instance:

If *room temperature* is *high*, then turn *cooling rate* to *high* ;

If *room temperature* is *low*, then turn *cooling rate* to *negative high* .

On the other hand, indirect AFCs incorporates fuzzy linguistic rules to describe and model the plant, and then construct the appropriate controller assuming that the embedded fuzzy logic system perfectly represents the unknown nominal plant. In such a case, the fuzzy if-then rules describing the behavior of a cooling system, for instance, may take the following form:

If *cooling rate* is *high*, then the *room temperature* becomes *low* ;

If *cooling rate* is *negative high*, then the *room temperature* becomes *high* .

5.4.2 First and Second Types of AFCs

Another classification criterion for AFCs is based on the type of adaptation technique used to tune the adjustable parameters of the FLC. The most commonly used adaptation methods, that are also used in hybrid neurofuzzy systems [49,50,68], are categorized either as linear, such as those based on the least mean square technique, or as nonlinear, such as those based on the gradient descent algorithm. Based on these two types of optimization methods, AFCs are classified into two types:

- *First-type* AFCs are those AFCs with fuzzy logic systems being *linear* in their adjustable parameters.
- *Second-type* AFCs are those AFCs with fuzzy logic systems being *nonlinear* in their adjustable parameters.

Before we proceed to a formal representation of the first and second types of AFCs, we introduce the following preliminary background.

Definition 5.4.1 *A fuzzy* singleton *A defined on the universe of discourse $\mathcal{X}$ is a fuzzy set defined by*

$$\mu_A(x) = \begin{cases} 1 & if\ x = a, \\ 0 & otherwise, \end{cases}$$

where a is a prespecified crisp parameter in $\mathcal{X}$ and $x \in \mathcal{X}$.

Due to its simplicity and low computational requirements, the fuzzy singleton is often used as a fuzzifier in several applications.

Definition 5.4.2 *The output $y = (y_1, \dots, y_m)^T$ of an n-input m-output FLC with a center-average defuzzifier, fuzzy rules as defined in (A.1), sum-product inference,* *and singleton output fuzzifier, is given by*

$$
\begin{aligned}
y_j &= f(x) \\
&= \frac{\sum_{l=1}^{L} \bar{y}_j^{(l)} \left(\prod_{i=1}^{n} \mu_{A_i^{(l)}}(x_i) \right)}{\sum_{l=1}^{L} \left(\prod_{i=1}^{n} \mu_{A_i^{(l)}}(x_i) \right)}, \quad j = 1, \ldots, m,
\end{aligned}
\tag{5.1}
$$

where L is the total number of the fuzzy rules, $\prod$ and $\sum$ denote the fuzzy t-norm and t-conorm operations used, respectively, and $\bar{y}_j^{(l)}$ is the point in V_j at which $\mu_{B_j^{(l)}}$ achieves its maximum value which is assumed to be 1.

Recall, from Section 5.2, that V_j is the universe of discourse of y_j. Here, we adopt the ordinary discrete product and sum as the t-norm and the t-conorm, respectively, for all the AFCs that are discussed throughout this thesis.

FLCs are well known to be powerful universal approximators. In other words, an FLC is capable of uniformly approximating any well-defined nonlinear function over a compact set U to any degree of accuracy.

Theorem 5.4.1 *For any given real continuous function g on the compact set $U \subset \mathbb{R}^n$ and arbitrary $\epsilon > 0$, there exists a function $f(x)$ in the form of (5.1) such that*

$$\sup_{x \in U} \| g(x) - f(x) \| < \epsilon.$$

The above universal approximation theorem, the proof of which can be found in [68, 128], shows the power of fuzzy logic systems in approximating continuous nonlinear functions and it justifies the recent increase in applying fuzzy logic systems to a wide spectrum of nonlinear systems.

If $\bar{y}_j^{(l)}$, $l = 1, \ldots, L$, $j = 1, \ldots, m$ are chosen as the FLC's free parameters, the FLC defined by (5.1) becomes an adaptive one. In this case, it can take the form of a neural network such as the one shown in figure 5.1 [107, 140]. Equation (5.1) can then be rewritten as

$$y_j = \sum_{l=1}^{L} \bar{y}_j^{(l)} \xi(x) = \Theta_j^T \xi(x), \quad j = 1, \dots, m, \tag{5.2}$$

where

$$\begin{aligned} \Theta_j &= (\bar{y}_j^{(1)}, \dots, \bar{y}_j^{(L)})^T, \\ \xi(x) &= (\xi_1(x), \dots, \xi_L(x))^T, \text{ and} \\ \xi_l(x) &= \frac{\prod_{i=1}^n \mu_{A_i^{(l)}}(x_i)}{\sum_{k=1}^L \left(\prod_{i=1}^n \mu_{A_i^{(k)}}(x_i) \right)}, \quad l = 1, \dots, L. \end{aligned} \tag{5.3}$$

The vector $\xi(x) \in \mathbb{R}^L$ is known as the fuzzy basis function vector, or the antecedent function vector, and it is provided by the neurons in the hidden layer of the neural network. The vector $\Theta_j \in \mathbb{R}^L$, $j = 1, \dots, m$, is called the parameter vector. The adjustable weights are $\bar{y}_j^{(l)}$, $l = 1, \dots, L$, $j = 1, \dots, m$. Thus the output of the MIMO FLC can be rewritten as

$$y = (y_1, \dots, y_m)^T = \Theta^T \xi(x), \tag{5.4}$$

where Θ is an $(L \times m)$ matrix with Θ_j as its $(L \times 1)$ jth column. Mathematically,

$$\Theta = [\Theta_1, \Theta_2, \dots, \Theta_m]. \tag{5.5}$$

In the case where in addition to $\bar{y}_j^{(l)}$, other parameters in which the FLC is nonlinear, such as the parameters of the input fuzzifiers, for instance, are chosen to be adjustable parameters as well, the corresponding AFC becomes a second-type

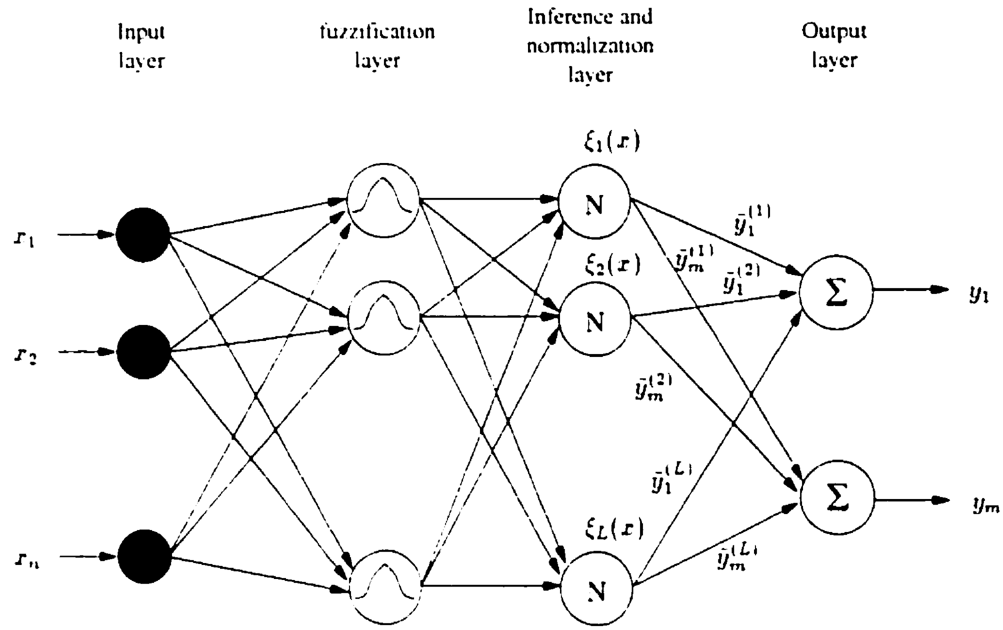


Figure 5.1: The Structure of an Adaptive MIMO FLC.

one. For example, if Gaussian fuzzy sets are used to fuzzify the input space with

$$\mu_{A_i^{(l)}}(x_i) = \exp\left[-\left(\frac{x_i - \bar{x}_i^{(l)}}{\sigma_i^{(l)}}\right)^2\right].$$

then equation (5.1) becomes

$$\begin{aligned} y_j &= f(x) \\ &= \frac{\sum_{l=1}^{L} \bar{y}_j^{(l)} \left(\prod_{i=1}^{n} \exp\left[-\left(\frac{x_i - x_i^{(l)}}{\sigma_i^{(l)}}\right)^2\right]\right)}{\sum_{l=1}^{L} \left(\prod_{i=1}^{n} \exp\left[-\left(\frac{x_i - x_i^{(l)}}{\sigma_i^{(l)}}\right)^2\right]\right)}, \quad j = 1, \ldots, m. \end{aligned} \tag{5.6}$$

Since equation (5.6) is nonlinear in $\bar{x}_i^{(l)}$ and $\sigma_i^{(l)}$, choosing these parameters to be adjustable in addition to parameter $\bar{y}_j^{(l)}$, for $i = 1, \ldots, n$, $l = 1, \ldots, L$, and

$j = 1, \ldots, m$, results in a second-type AFC.

It is important to point out that regardless of the linearity or the nonlinearity of the AFC in its free parameters, the embedded fuzzy logic system, and hence the AFC itself, is always computationally nonlinear. It is also worth mentioning that since the number of free parameters in a first-type AFC is less than that of a second-type one, the former type of controller is computationally less demanding to converge to an optimal set of parameters which leads to a minimal tracking error. However, this set of optimal parameters may not always lead to a satisfactory minimal tracking error. This is because the free parameters search space in the case of a first-type AFC is limited compared to that of a second-type AFC. Although it is usually very time consuming to search for optimal parameter values of a second-type AFC, an optimal second-type AFC is most likely to outperform an optimal first-type AFC.

5.5 Design of First-Type Direct AFCs

Although different strategies have been proposed in the literature to design AFCs, depending on their types and the problem specifications, the general design approach and the global design procedure have been similar. As for this work, the main focus has been on the design of a first-type direct AFC for the control of cooperative manipulator systems. In the rest of this thesis, only first-type direct AFCs are considered. More material about the other types of AFCs and their features can be found in [8, 33, 60, 91, 115, 128, 140].

Consider a pth-order m-input m-output nonlinear system with the following state space representation:

$$\begin{cases} x^{(p)} &= f(x, \dot{x}, \ldots, x^{(p-1)}) + g(x, \dot{x}, \ldots, x^{(p-1)})\, u + d, \\ y &= x, \end{cases} \tag{5.7}$$

where f and g are *unknown* continuous functions. $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^m$ are the input and the output of the system, respectively. $\underline{x} = (x, \dot{x}, \ldots, x^{(p-1)})^T \in \mathbb{R}^n$ is the state space vector, with $x \in \mathbb{R}^m$ $(n = p \cdot m)$, and $d \in \mathbb{R}^m$ is vector term originating from the effect of the external disturbances. Here, the notation $x^{(q)}$ denotes the qth time-derivative of vector x. In other words,

$$x^{(q)} = \frac{\partial^q x}{\partial t^q},$$

where q is a positive integer.

5.5.1 Control Objectives

In general, the control objectives are very specific to the problem in hand and they generally differ from one problem to another. However, a common control objective for the large majority of control problems involving first-type direct AFCs is to make the system's output y track a given bounded reference output signal $y_r(t)$. More formally, the main control objective can be stated as:

determine a feedback control u in the form $u = u(\underline{x}|\Theta)$, which is based on a fuzzy logic system with Θ as its free parameter matrix, and an adaptation law for adjusting Θ under the constraint that the 2-norm $\|e\|_2$ of the tracking error $e = y_r - y$ is minimized.

5.5.2 Controller Design

Ideally, if functions f and g are known and $d = 0$, then the feedback control

$$u = g^{-1}(\underline{x})(v - f(\underline{x})),$$

with a certain external input $v \in \mathbb{R}^m$ yields the linearized system

$$y^{(p)} = v. \tag{5.8}$$

The reference trajectory $y_r(t)$ can then be asymptotically tracked in the linear external controller

$$v = y^{(p)} + \Lambda_0 e + \Lambda_1 \dot{e} + \cdots + \Lambda_{p-1} e^{(p-1)}, \tag{5.9}$$

where $\Lambda_i \in \mathbb{R}^m$, $i = 0, \ldots, p-1$, is a diagonal gain matrix of the form $\Lambda_i = \text{diag}(k_{i1}, k_{i2}, \ldots, k_{im})$, with $k_{ij} \in \mathbb{R}$ for $j = 1, \ldots, m$. Substituting (5.8) into (5.9) yields

$$e^{(p)} + \Lambda_{p-1} e^{(p-1)} + \cdots + \Lambda_1 \dot{e} + \Lambda_0 e = 0. \tag{5.10}$$

If the coefficients k_{ij}, $i = 0, \ldots, p-1$, $j = 1, \ldots, m$, are chosen in such a way that all the roots (every single element of vector $e^{(q)}$ for $q = 0, \ldots, p$) of equation (5.10) belong to the open left-half of the s-plane, then the tracking error converges to zero (i.e., $\lim_{t\to\infty} e(t) = 0$), which is the main control objective. However, in many real-life complex systems, such as cooperative robotic systems, it is either very difficult or impossible to precisely determine the nonlinear functions f and g. Besides, in such systems, the vector d is almost never equal to zero. This is mainly due to the inevitable ubiquitous existence of parametric and modeling uncertainties and ill-defined external disturbances in these systems. As a result, equation (5.10) becomes instead expressed as

$$e^{(p)} + \Lambda_{p-1}e^{(p-1)} + \cdots + \Lambda_1\dot{e} + \Lambda_0 e = \eta(t), \tag{5.11}$$

where $\eta(t)$ is an unknown time-varying vector in $\mathbb{R}^m$. Equations (5.11) clearly shows that the tracking error e does not converge to zero where $\eta(t) \neq 0$. Taking the above constraints into account, the feedback control u can be computed as

$$u = \hat{g}^{-1}(\underline{x})(v - \hat{f}(\underline{x}) - \lambda(t)),$$

where v is as defined in (5.8), $\lambda(t)$ is a time-varying robust compensator, and $\hat{g}^{-1}(\underline{x})$ and $\hat{f}(\underline{x})$ are time-varying estimates of $g^{-1}(\underline{x})$ and $f(\underline{x})$, respectively. Thus, the jth column of the control law u becomes expressed as

$$u_j = [\hat{g}^{-1}(\underline{x})]_j^T v + [\hat{g}^{-1}(\underline{x})]_j^T(-\hat{f}(\underline{x}) - \lambda(t)), \quad j = 1, \ldots, m, \tag{5.12}$$

with $[\hat{g}^{-1}(\underline{x})]_j^T$ being the jth row of matrix $\hat{g}^{-1}(\underline{x})$. The basic idea now is to replace $[\hat{g}^{-1}(\underline{x})]_j$ and $\hat{f}(\underline{x})$ with first-type direct MIMO AFCs $\gamma_j(\underline{x}|\Theta_j)$ and $\beta(\underline{x}|\Theta)$, respectively, $j = 1, \ldots, m$. These AFCs are in the form given by equations (5.2), (5.3), (5.4), and (5.5), where $\Theta_j \in \mathbb{R}^{L_{\gamma_j} \times m}$ and $\Theta \in \mathbb{R}^{L_\beta \times m}$, $j = 1, \ldots, m$, are the adjustable parameters of $\gamma_j(\underline{x}|\Theta_j)$ and $\beta(\underline{x}|\Theta)$, respectively, and L_{γ_j} and L_β are their respective number of rules. Equation (5.12) can then be rewritten as

$$u_j = [\gamma_j(\underline{x}|\Theta_j)]^T v + [\gamma_j(\underline{x}|\Theta_j)]^T(-\beta(\underline{x}|\Theta) - \lambda(t)), \quad j = 1, \ldots, m. \tag{5.13}$$

The robust compensator $\lambda(t)$ has a major effect on the overall controller's efficiency and stability. It plays a major role in the overall control scheme as it also dictates how the controller's free parameters are updated. Several strategies for the design of robust compensators have been discussed in the literature [91, 127, 128]. The techniques used for this purpose have been quite diverse as they are very dependent on the properties of the control problem and also on the control objectives set for each particular problem. The following is a general outline for the design steps of a first-type direct AFC.

Algorithm: General Outline of First-Type Direct AFC

Step 1: Off-line processing.

- Specify the coefficients k_{ij}, $i = 0, \ldots, p-1$, $j = 1, \ldots, m$, such that all the roots of equation (5.10) are initially in the open left-half of the s-plane.
- Design a robust compensator $\lambda(t)$ that would maintain the global stability of the controller while leading to the satisfaction of the control objective(s).
- Determine certain adaptive laws to update the AFCs adjustable parameters Θ_j and Θ, $j = 1, \ldots, m$, so that the control objectives are achieved.
- Construct the fuzzy rules and the membership functions for the fuzzy logic systems of $\gamma_j(\underline{x}|\Theta_j)$ and $\beta(\underline{x}|\Theta)$, $j = 1, \ldots, m$.

Step 2: Online adaptation.

- Apply the feedback control (5.13) to the plant given in (5.7) while using the robust compensator designed in Step 1.

- Using the adaptive laws derived in Step 1 to continuously adjust the AFCs free parameters Θ_j and Θ, $j = 1, \ldots, m$.

Deriving the adaptive laws for the AFCs free parameters is usually achieved by analyzing the stability of the overall controller using a Lyapunov stability approach. The main idea is to choose $\dot{\Theta}_j$ and $\dot{\Theta}$, $j = 1, \ldots, m$, as to annihilate certain terms that may lead to a positive time-derivative of the Lyapunov function candidate.

It is worth mentioning that the discussion being carried out in this section about the design of first-type direct AFCs is very brief as it is only meant for giving a general idea on how such a type of controllers operate. It is difficult to provide a thorough analysis because the technical details of the design and stability of AFCs are very problem specific and they vary from a control system to another. Details about several AFCs as customized for different applications can be found in [8, 33, 60, 91, 115, 128, 140]. We will provide in subsequent chapters more material on the design of special first-type direct AFCs for the control of cooperative manipulators.

5.6 AFCs Versus Neural Controllers

When a neural network [68, 85] is used as a controller within a control loop, it is referred to as a *neural controller* [38, 53, 54, 85]. As it is discussed in [68, 85, 128], two of the main driving forces for using artificial neural networks (ANNs) have been their universal approximation capabilities and the well established training algorithms that enable them to tune their adjustable parameters (i.e., weights and biases) to minimize the networks cumulative error (between training data and actual output). These two major assets have drawn the attention of several researchers who have successfully applied different types of ANNs in a large number of applications, especially in the field of nonlinear control [85]. On the other hand AFCs are also proven to be universal approximators and they are also equipped with online adaptation mechanisms to tune their free parameters. In addition, these adaptation techniques are very similar to those that are often used in the training of most ANNs. Some of these adaptive algorithms are the gradient decent, the least squared error, and the Lyapunov stability approach techniques, for instance. The complementary correlation between ANNs and (adaptive) fuzzy logic systems is metaphorically described by Wang [128] as

> *"In some sense, artificial neural networks try to emulate the 'hardware' of the human brain, whereas adaptive fuzzy systems try to emulate the 'software' in the human brain."*

At first sight, AFCs and neural controllers seem to be quite alike. Nevertheless, some key differences do exist. A major advantage of an AFC over a neural one is that the parameters of the former type of controllers have clear physical meanings. These parameters often represent the characteristics of the membership functions defined in the knowledge base of the fuzzy logic system. For instance, in equation (5.6), $\bar{y}_j^{(l)}$ represents the center of the fuzzy singleton function in the consequent part of the lth rule. The parameters $x_i^{(l)}$ and $\sigma_i^{(l)}$ denote the center and the width of the Gaussian membership function in the antecedent part of the lth rule, respectively. This feature makes it possible to develop some sort of heuristics on how to choose the initial values of such parameters. On the other hand, the parameters of ANNs, in general, do not have a clear relationship with respect to the network's input-output data. As such, their initial values are usually chosen randomly. It is important to mention that the parameters initial values have a high impact on the convergence speed of the controller, especially if it uses a gradient-based training algorithm.

Another fundamental difference between AFCs and neural controllers is that the formers are capable of incorporating linguistic fuzzy information in a systematic manner, whereas this is not possible for the latters. This feature is important especially in highly uncertain complex systems for which human operators acquired a certain expertise on how these systems behave under different working conditions. For instance, controlling an aircraft in varying weather conditions, or an inverted pendulum under different types and intensities of external disturbances, is a very complex problem from a theoretical point of view, nevertheless humans have been quite successful in accomplishing these tasks without referring to their mathematical models. The human knowledge can be easily converted into a set of linguistic if-then rules which can be systematically incorporated in a fuzzy logic system to be used in an AFC.

It is also important to mention that the efficiency in neural controllers is heavily dependent on the structure of the ANNs, such as the number of hidden layers, the number of neurons in each layer, and the activation functions used in each layer, just to name a few. These parameters are usually determined by a trial-and-error process. In fact, determining the appropriate structure of a given ANN could become a tedious procedure especially if the system being controlled is a complex one. Besides, even when they show satisfactory performances like in [31, 135], their behavior is still not easily understood or interpreted. The performance of these neural network based adaptive techniques as applied to other types of problems is not guaranteed and cannot be easily quantified. In addition, the number of parameters on which a neural network depend is usually high and is proportional to the complexity of its architecture. This dramatically enlarges the parameters' tuning space and makes it even more difficult for the controller to achieve its main goals in terms of tracking performances and stability. It is worth pointing out that in many cases, like in [7, 12, 16, 66], a formal mathematical proof of the control system's stability could not even be accomplished.

5.7 Conclusion

We have reviewed in this chapter the major features of a class of soft computing based controllers namely adaptive fuzzy controllers and compared them favorably with standard adaptive controller approaches and even with the recently much talked about neural based controllers. In the next two chapters, we provide novel schemes of AFC systems that are readily implementable to complex dynamic systems such as those of cooperative robotics. We believe that this should contribute in a substantial manner to the increasing body of literature being published in the field of soft computing tools' applications to robotic systems.

Chapter 6

A Robust Adaptive Fuzzy Approach for the Control of CR

6.1 Introduction

Following the motivations that led us to investigate adaptive fuzzy controllers and after outlining the main features of such a class of controllers, we develop in this chapter a customized direct model-reference adaptive fuzzy controller (AFC) for the control of multi-arm cooperative robotic systems. To our best knowledge, this is the first control implementation of its kind to this complex type of structure. The controller is designed in such a way as to attain the control objectives defined in Sections 1.3 and 3.1 under significant amount of structured and unstructured uncertainties. The convergence and the stability properties of the proposed AFC are also formally analyzed through a Lyapunov stability approach. Afterwards, computer simulation results are illustrated to confirm the theoretical properties of the controller and to compare them with those of the CAC presented in Chapter 4.

6.2 AFC Design for CR

To eliminate the error prone dynamic modeling process in a coordinated-manipulator system, an adaptive fuzzy controller (AFC) is developed. The AFC developed here possesses an adaptation mechanism allowing it to learn the system's dynamics without the need for prior knowledge of the system's dynamic structure or the manipulators physical parameters. In other words, nothing about the matrices $D_i(q_i)$, $C_i(q_i, \dot{q}_i)$, and $G_i(q_i)$, $i = 1, \ldots, m$, is assumed to be known. To ensure that $D_i(q_i)$, $C_i(q_i, \dot{q}_i)$, and $G_i(q_i)$, can actually be approximated by FLCs, the following assumption is adopted

Assumption 6.2.1 *The load distribution matrix $c_i(t)$ is either exclusively dependent on q_i, $\dot{q}_i$, and/or $\ddot{q}_i$, or it is slowly varying in time. In other words, for $i = 1, \ldots, m$,*

$$c_i(t) = c_i(q_i, \dot{q}_i, \ddot{q}_i), \quad or$$

$$\dot{c}_i(t) \approx 0.$$

Note that a constant matrix $c_i(t)$ satisfies assumption 6.2.1. This is very common indeed in real world applications.

Let $P_i(t) \in \mathbb{R}^{k_i}$, $i = 1, \ldots, m$, be a time-varying function representing the uncertainty in the payload's variation. This variation depends on the term $\{D_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + G_i(q_i)\}$ in which the only time-varying parameters are $\{q_i, \dot{q}_i, \ddot{q}_i\}$.

Hence, the function $P_i(t)$ is directly dependent on $\{q_i, \dot{q}_i, \ddot{q}_i\}$ [140].

$$P_i(t) = P_i(q_i, \dot{q}_i, \ddot{q}_i), \quad i = 1, \ldots, m.$$

Similarly, the rest of the system's dynamics can be expressed as a function $R_i(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i})$ which is dependent on $\{q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}\}$. In most practical cases, unmodeled uncertainties, τ_{d_i}, $i = 1, \ldots, m$, are mostly dominated by unstructured friction forces and unknown external disturbances acting on the manipulators joints. So, it is only normal to consider that the time variations of such uncertainties are heavily dependent on q_i and $\dot{q}_i$. This should be the case for a wide range of real world systems.

Assumption 6.2.2 *Modeling uncertainties τ_{d_i}, $i = 1, \ldots, m$, are dominantly dependent on q_i and $\dot{q}_i$. That is, there exists a function $T_{d_i}(q_i, \dot{q}_i) \in \mathbb{R}^{k_i}$, which is dependent on $\{q_i, \dot{q}_i\}$, such that*

$$\tau_{d_i}(t) = T_{d_i}(q_i, \dot{q}_i) + \varepsilon_i(t), \tag{6.1}$$

where $\|\varepsilon_i(t)\| \approx 0$.

Therefore, the controller (4.19) can be rewritten as

$$\tau_i = R_i(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}) + P_i(q_i, \dot{q}_i, \ddot{q}_i) - T_{d_i}(q_i, \dot{q}_i) - K_{s_i} s_i - J_{\phi_i}^T(q_i)(K_i \tilde{x} + f_{d_i}). \tag{6.2}$$

Let

$$U_i(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}) = R_i(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}) + P_i(q_i, \dot{q}_i, \ddot{q}_i) - T_{d_i}(q_i, \dot{q}_i). \tag{6.3}$$

Then, equation (6.2) can be rewritten as

$$\tau_i = U_i(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}) - K_{s_i} s_i - J_{\phi_i}^T(q_i)(K_i \tilde{x} + f_{d_i}).$$

Since the function $U_i(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i})$ is a completely unknown vector, according to the universal approximation theorem, it can be approximated by a MIMO fuzzy logic system $\hat{U}_i(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}|\Theta_i)$ which is defined by equations (5.2), (5.3), (5.4), and (5.5). Mathematically,

$$\hat{U}_i(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}|\Theta_i) = \begin{bmatrix} \hat{U}_{i1}(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}|\Theta_{i1}) \\ \hat{U}_{i2}(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}|\Theta_{i2}) \\ \vdots \\ \hat{U}_{ik_i}(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}|\Theta_{ik_i}) \end{bmatrix}$$

$$= \begin{bmatrix} \Theta_{i1}^T \xi_i(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}) \\ \Theta_{i2}^T \xi_i(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}) \\ \vdots \\ \Theta_{ik_i}^T \xi_i(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}) \end{bmatrix}.$$

where the input vector to the FLC is $x_i = (q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i})^T$ and Θ_{ij} is the jth column of matrix Θ_i. Note that the input vector x_i is composed of $(5k_i)$ real elements (i.e., $x_i \in \mathbb{R}^{5k_i}$.) So, if κ_i fuzzy labels are assigned to each of these elements, the total number of fuzzy rules, L_i, in the FLC of each robot i is $(\kappa_i)^{5k_i}$. This is usually a very large number when tackling real-life cooperative robotic systems. For instance, if κ_i and k_i were chosen to be 3 and 2, respectively, the FLC of each of the m robots has to fire 3^{10} $(= 59,049)$ rules in order to compute and dispatch the control signals to the k_i actuators of each arm. This is obviously impractical as it would consume a great amount of computational resources considering the relatively simple configuration of such a cooperative manipulator system.

6.3 Rule Reduction Technique

To alleviate this computational burden, a fuzzy rule reduction algorithm similar to the one proposed in [140] is adopted here. Using the recursive Newton-Euler method, $\dot{q}_{r_i}$ can be replaced by $\dot{q}_i$. The function $U_i(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i})$, and hence the function $\hat{U}_i(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}|\Theta_i)$, can then be rewritten as

$$U_i(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}) = U_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}), \quad \text{and}$$
$$\hat{U}_i(q_i, \dot{q}_i, \ddot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}|\Theta_i) = \hat{U}_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}|\Theta_i).$$

To reduce even further the total number of fuzzy rules required, a decomposition procedure is suggested to decompose the function $U_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i})$ into three different functions. Nominally,

$$U_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}) = U_i^1(q_i, \dot{q}_i) + U_i^2(q_i, \ddot{q}_i) + U_i^3(q_i, \ddot{q}_{r_i}). \tag{6.4}$$

Similarly, its approximated value generated by the MIMO-FLC is expressed as

$$\hat{U}_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}|\Theta_i) = \hat{U}_i^1(q_i, \dot{q}_i|\Theta_i^1) + \hat{U}_i^2(q_i, \ddot{q}_i|\Theta_i^2) + \hat{U}_i^3(q_i, \ddot{q}_{r_i}|\Theta_i^3), \tag{6.5}$$

where

$$\begin{aligned} \hat{U}_i^1(q_i, \dot{q}_i|\Theta_i^1) &= (\Theta_i^1)^T \xi_i^1(q_i, \dot{q}_i) \\ \hat{U}_i^2(q_i, \ddot{q}_i|\Theta_i^2) &= (\Theta_i^2)^T \xi_i^2(q_i, \ddot{q}_i) \\ \hat{U}_i^3(q_i, \ddot{q}_{r_i}|\Theta_i^3) &= (\Theta_i^3)^T \xi_i^3(q_i, \ddot{q}_{r_i}). \end{aligned} \tag{6.6}$$

The adaptation laws of the consequent parameter matrices of the three fuzzy logic components of the FLC are defined by

$$\begin{aligned}
\dot{\Theta}_i^1 &= -(\Gamma_i^1)^{-1}\, \xi_i^1(q_i, \dot{q}_i)\, s_i^T \\
\dot{\Theta}_i^2 &= -(\Gamma_i^2)^{-1}\, \xi_i^2(q_i, \ddot{q}_i)\, s_i^T \\
\dot{\Theta}_i^3 &= -(\Gamma_i^3)^{-1}\, \xi_i^3(q_i, \ddot{q}_{r_i})\, s_i^T ,
\end{aligned} \tag{6.7}$$

where $\Gamma_i^j \in \mathbb{R}^{(\kappa_i)^{2k_i} \times (\kappa_i)^{2k_i}}$, for $i = 1, \ldots, m$ and $j = 1, 2, 3$, is a positive definite gain matrix. Notice that with this fuzzy rule reduction technique, the total number of rules L_i that are fired by the three fuzzy logic components of the FLC for each robot i, drops down to $3(\kappa_i)^{2k_i}$. This is a significant improvement when compared to the original number (i.e., $(\kappa_i)^{5k_i}$) and it provides a major improvement in terms of computational efficiency which would facilitate its real world implementation. Going back to the previous example with $\kappa_i = 3$ and $k_i = 2$, the value of L_i is now reduced to 3×3^4 (= 234). Based on the above results, the adaptive fuzzy control law becomes defined by

$$\tau_i = \hat{U}_i^1(q_i, \dot{q}_i|\Theta_i^1) + \hat{U}_i^2(q_i, \ddot{q}_i|\Theta_i^2) + \hat{U}_i^3(q_i, \ddot{q}_{r_i}|\Theta_i^3) - K_{s_i} s_i - J_{\phi_i}^T(q_i)(K_i \tilde{x} + f_{d_i}). \tag{6.8}$$

Notice that since the manipulators kinematics are assumed to be perfectly known (assumption 3.1.2), the last two terms of equation (6.8) can be accurately computed. Figure 6.1 depicts the block diagram of the AFC with figure 6.2 showing the block diagram of its three fuzzy logic components.

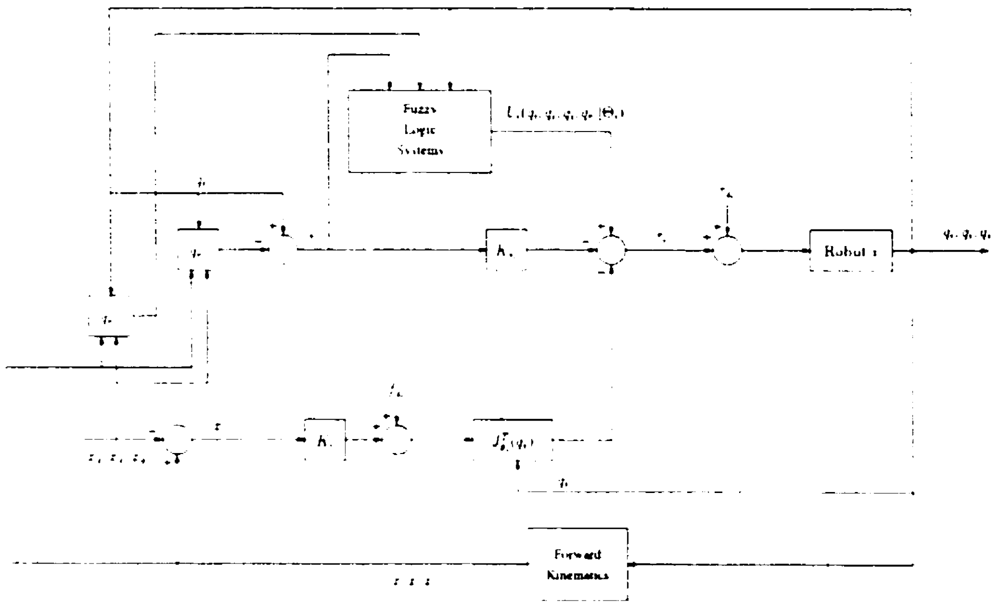


Figure 6.1: Block Diagram of the AFC Architecture.

6.4 Stability Analysis

Substituting the controller equation (6.8) into the dynamics equation (3.12) results in the following error dynamics equation

$$D_i(q_i)\dot{s}_i + C_i(q_i, \dot{q}_i)s_i - J_{\phi_i}^T(q_i)\tilde{f}_i = \tilde{U}_i - K_{s_i}s_i - J_{\phi_i}^T(q_i)K_i\tilde{x}, \qquad (6.9)$$

where $\tilde{f}_i$ is the internal force error in robot i, and $\tilde{U}_i$ is the error of approximating the robot's dynamics. That is

$$\begin{aligned} \tilde{f}_i &\stackrel{\text{def}}{=} f_i - f_d, \\ \tilde{U}_i &\stackrel{\text{def}}{=} \hat{U}_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}|\Theta_i) - U_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}). \end{aligned} \qquad (6.10)$$

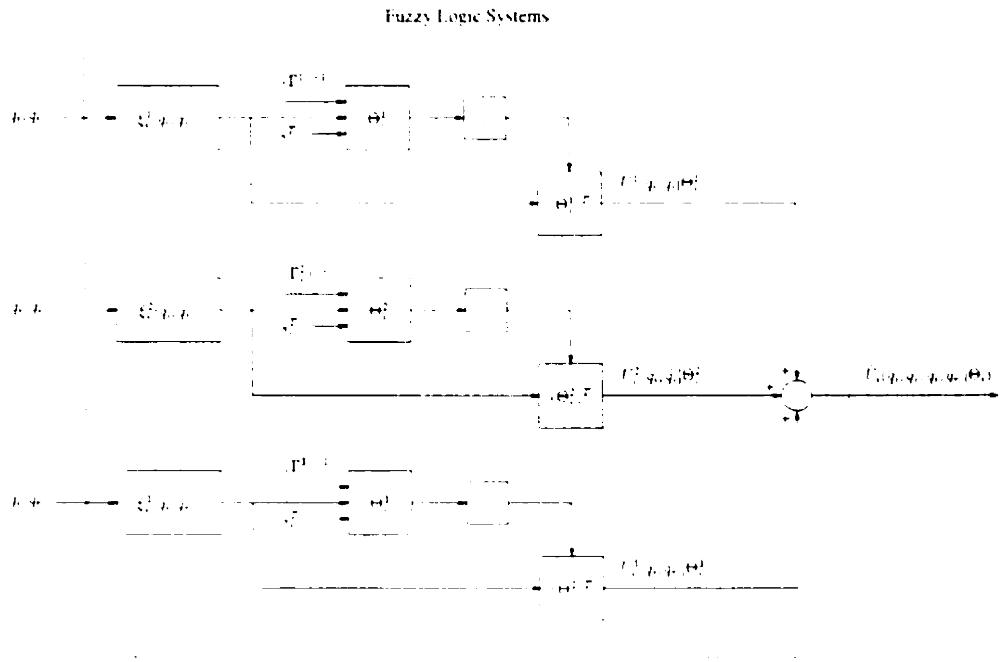


Figure 6.2: Block Diagram of the Fuzzy Logic Modules.

Let Θ_i^* denote the optimal parameter matrix of the fuzzy logic engine $\hat{U}_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}|\Theta_i)$. Equation (6.10) can then be rewritten as

$$\bar{U}_i = \hat{U}_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}|\Theta_i^*) + w_i. \tag{6.11}$$

where w_i is known as the minimum approximation error of the fuzzy logic engine $\hat{U}_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}|\Theta_i)$ and is defined by

$$w_i = \hat{U}_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}|\Theta_i^*) - U_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}).$$

with

$$\begin{aligned}\tilde{U}_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}|\tilde{\Theta}_i) &= \hat{U}_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}|\Theta_i) - \hat{U}_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}|\Theta_i^*) \\ &= \tilde{\Theta}_i^T \xi_i(q_i, \dot{q}_i, \ddot{q}_i, \ddot{q}_{r_i}) \\ \tilde{\Theta}_i &= \Theta_i - \Theta_i^*.\end{aligned}$$

Using equations (6.11) and (6.4), the error dynamics (6.9) can be reformulated as

$$\begin{aligned}D_i(q_i)\dot{s}_i + C_i(q_i, \dot{q}_i)s_i - J_{\phi_i}^T(q_i)\tilde{f}_i = \tilde{U}_i^1(q_i, \dot{q}_i|\tilde{\Theta}_i^1) + \tilde{U}_i^2(q_i, \ddot{q}_i|\tilde{\Theta}_i^2) + \tilde{U}_i^3(q_i, \ddot{q}_{r_i}|\tilde{\Theta}_i^3) \\ - K_{s_i}s_i - J_{\phi_i}^T(q_i)K_i\dot{x} + \epsilon_i(t).\end{aligned} \tag{6.12}$$

where

$$\tilde{U}_i^1(q_i, \dot{q}_i|\tilde{\Theta}_i^1) \stackrel{\text{def}}{=} \hat{U}_i^1(q_i, \dot{q}_i|\Theta_i^1) - \hat{U}_i^1(q_i, \dot{q}_i|\Theta_i^{1*}) = (\tilde{\Theta}_i^1)^T \xi_i^1(q_i, \dot{q}_i) \tag{6.13}$$

$$\tilde{U}_i^2(q_i, \ddot{q}_i|\tilde{\Theta}_i^2) \stackrel{\text{def}}{=} \hat{U}_i^2(q_i, \ddot{q}_i|\Theta_i^2) - \hat{U}_i^2(q_i, \ddot{q}_i|\Theta_i^{2*}) = (\tilde{\Theta}_i^2)^T \xi_i^2(q_i, \ddot{q}_i) \tag{6.14}$$

$$\tilde{U}_i^3(q_i, \ddot{q}_{r_i}|\tilde{\Theta}_i^3) \stackrel{\text{def}}{=} \hat{U}_i^3(q_i, \ddot{q}_{r_i}|\Theta_i^3) - \hat{U}_i^3(q_i, \ddot{q}_{r_i}|\Theta_i^{3*}) = (\tilde{\Theta}_i^3)^T \xi_i^3(q_i, \ddot{q}_{r_i}) \tag{6.15}$$

$$\epsilon_i(t) \stackrel{\text{def}}{=} w_i^1 + w_i^2 + w_i^3.$$

Θ_i^{k*} is the optimal parameter matrix of the fuzzy logic engine $\hat{U}_i^k$, for $k = 1, 2, 3$, and

$$\begin{aligned}\tilde{\Theta}_i^k &= \Theta_i^k - \Theta_i^{k*} \\ w_i^1 &= \hat{U}_i^1(q_i, \dot{q}_i|\Theta_i^{1*}) - U_i^1(q_i, \dot{q}_i) \\ w_i^2 &= \hat{U}_i^2(q_i, \ddot{q}_i|\Theta_i^{2*}) - U_i^2(q_i, \ddot{q}_i) \\ w_i^3 &= \hat{U}_i^3(q_i, \ddot{q}_{r_i}|\Theta_i^{3*}) - U_i^3(q_i, \ddot{q}_{r_i}).\end{aligned}$$

Remark 6.4.1 *If the fuzzy logic systems $\hat{U}_i^1(q_i, \dot{q}_i|\Theta_i^1)$, $\hat{U}_i^2(q_i, \ddot{q}_i|\Theta_i^2)$, and $\hat{U}_i^3(q_i, \ddot{q}_{r_i}|\Theta_i^3)$, are information rich enough, that is if their input fuzzy labels uniformly and thoroughly span the whole input space and the number of their fuzzy rules is large enough to describe the systems dynamics, then from the universal approximation theorem (theorem 5.4.1), the term $\epsilon_i(t)$ can be approximated to zero with any degree of accuracy [127, 128]. Since $\epsilon_i(t)$ can be smaller than any machine precision, it can be considered to be practically zero.*

$$\epsilon_i(t) = 0. \tag{6.16}$$

Theorem 6.4.1 *Suppose that the gains γ_i are chosen to have the same value $\gamma > 0$ for all m manipulators. If the matrix gains $K_{s_i} > \eta_i$ and $K_i > 0$, $i = 1, \ldots, m$, then the adaptive fuzzy controller (6.8) gives rise to an asymptotic convergence of s_i and $\tilde{x}$ to zero.*

Proof: Consider the following Lyapunov function candidate

$$V = \sum_{i=1}^{m} V_i,$$

where

$$V_i = \frac{1}{2}\left\{ s_i^T D_i(q_i) s_i + \tilde{x}^T K_i \tilde{x} + \sum_{k=1}^{3}\sum_{j=1}^{k_i} (\tilde{\Theta}_{ij}^k)^T \Gamma_i^k (\tilde{\Theta}_{ij}^k) \right\},$$

with $\tilde{\Theta}_{ij}^k$ denoting the jth column of matrix $\tilde{\Theta}_i^k$. Since $D_i(q_i)$, K_i, and Γ_i^k are positive definite matrices, V_i, and hence V, are non-negative scalars. From the error dynamics (6.12) and equation (3.13).

$$\begin{aligned}\dot{V}_i = & - s_i^T(K_{s_i} - \frac{1}{2}\dot{c}_i(t)Q_0(q_i, \dot{q}_i))s_i - s_i^T J_{\phi_i}^T(q_i)K_i\tilde{x} \\ & + s_i^T J_{\phi_i}^T(q_i)\tilde{f}_i + \tilde{x}^T K_i\dot{\tilde{x}} + \sum_{k=1}^{3}\sum_{j=1}^{k_i}(\tilde{\Theta}_{ij}^k)^T\Gamma_i^k(\dot{\tilde{\Theta}}_{ij}^k) \\ & + s_i^T\left[\hat{U}_i^1(q_i, \dot{q}_i|\hat{\Theta}_i^1) + \hat{U}_i^2(q_i, \ddot{q}_i|\hat{\Theta}_i^2) + \hat{U}_i^3(q_i, \ddot{q}_{r_i}|\hat{\Theta}_i^3) + \epsilon_i(t)\right].\end{aligned}$$

Substituting equations (4.18), (6.13), (6.14), and (6.15), leads to

$$\begin{aligned}\dot{V}_i = & -s_i^T(K_{s_i} - \frac{1}{2}\dot{c}_i(t)Q_0(q_i, \dot{q}_i))s_i - \gamma_i\tilde{x}^T K_i\tilde{x} + (\dot{\tilde{x}} + \gamma_i\tilde{x})^T\tilde{f}_i \\ & + \sum_{k=1}^{3}\sum_{j=1}^{k_i}(\tilde{\Theta}_{ij}^k)^T\Gamma_i^k(\dot{\tilde{\Theta}}_{ij}^k) \\ & + s_i^T\left[(\tilde{\Theta}_i^1)^T\xi_i^1(q_i, \dot{q}_i) + (\tilde{\Theta}_i^2)^T\xi_i^2(q_i, \ddot{q}_i) + (\tilde{\Theta}_i^3)^T\xi_i^3(q_i, \ddot{q}_{r_i}) + \epsilon_i(t)\right] \\ = & \dot{V}_{i1} + \dot{V}_{i2} + \dot{V}_{i3},\end{aligned}$$

where

$$\begin{aligned}\dot{V}_{i1} = & -s_i^T(K_{s_i} - \frac{1}{2}\dot{c}_i(t)Q_0(q_i, \dot{q}_i))s_i - \gamma_i\tilde{x}^T K_i\tilde{x} \\ \dot{V}_{i2} = & (\dot{\tilde{x}} + \gamma_i\tilde{x})^T\tilde{f}_i \\ \dot{V}_{i3} = & \sum_{k=1}^{3}\sum_{j=1}^{k_i}(\tilde{\Theta}_{ij}^k)^T\Gamma_i^k(\dot{\tilde{\Theta}}_{ij}^k) + s_i^T\left[(\tilde{\Theta}_i^1)^T\xi_i^1(q_i, \dot{q}_i) + (\tilde{\Theta}_i^2)^T\xi_i^2(q_i, \ddot{q}_i)\right. \\ & \left.+(\tilde{\Theta}_i^3)^T\xi_i^3(q_i, \ddot{q}_{r_i}) + \epsilon_i(t)\right]. \end{aligned} \tag{6.17}$$

From the property of internal forces, we have $\sum_{i=1}^{m} f_i = 0$ and $\sum_{i=1}^{m} f_{d_i} = 0$. Thus, setting γ_i to a unique value γ for all m manipulators yields

$$\sum_{i=1}^{m}\dot{V}_{i2} = 0.$$

Substituting equations (6.7) and (6.16) into (6.17) leads to

$$\dot{V}_{i3} = 0.$$

From equation (3.14),

$$\dot{V}_{i1} \leq -s_i^T(K_{s_i} - \eta_i I_{k_i})s_i - \gamma \tilde{x}^T K_i \tilde{x}.$$

If the gain matrices K_{s_i} and K_i are chosen in such a way as to satisfy the following constraints

$$K_{s_i} > \eta_i I_{k_i}\ , \quad K_i > 0\ .$$

then $\dot{V}_{i1} \leq 0$, implies that $\dot{V}_i \leq 0$. Hence V_i is a non-increasing function, and as such s_i, $\tilde{x}$, and $\tilde{\Theta}_{ij}^k$, for $k = 1, 2, 3$, are bounded, since they appear in V_i. If the desired trajectory x_d is uniformly continuous, then Lemma 6.4.1 implies that s_i and $\tilde{x}$ are uniformly continuous. Remember that $(\dot{c}_i(t)Q_0(q_i, \dot{q}_i))$ is uniformly continuous (assumption 3.4.1). Thus, from Barbalat's lemma,

$$\lim_{t\to\infty} \dot{V}_{i1} = 0.$$

And therefore,

$$\lim_{t\to\infty} s_i = 0, \quad \text{and} \quad \lim_{t\to\infty} \tilde{x} = 0.$$

■

Lemma 6.4.1 *If s_i, $\tilde{x}$, and $\tilde{\Theta}_{ij}^k$, for $k = 1, 2, 3$, are bounded, and if the desired trajectory x_d is uniformly continuous, then s_i and $\tilde{x}$ are uniformly continuous.*

Proof: Premultiplying the error dynamics (6.12) by $J_{\phi_i}(q_i)D_i^{-1}(q_i)$, yields

$$\begin{aligned}J_{\phi_i}(q_i)\dot{s}_i - J_{\phi_i}(q_i)D_i^{-1}(q_i)J_{\phi_i}^T(q_i)\tilde{f}_i = J_{\phi_i}(q_i)D_i^{-1}(q_i)\Big[\hat{U}_i^1(q_i,\dot{q}_i|\tilde{\Theta}_i^1)\\ +\hat{U}_i^2(q_i,\ddot{q}_i|\tilde{\Theta}_i^2) + \hat{U}_i^3(q_i,\ddot{q}_{r_i}|\tilde{\Theta}_i^3)\Big] - J_{\phi_i}(q_i)D_i^{-1}(q_i)\left[C_i(q_i,\dot{q}_i) + K_{s_i}\right]s_i\\ - J_{\phi_i}(q_i)D_i^{-1}(q_i)J_{\phi_i}^T(q_i)K_i\tilde{x} + J_{\phi_i}(q_i)D_i^{-1}(q_i)\epsilon_i(t).\end{aligned} \quad (6.18)$$

Differentiating equation (4.18) with respect to time leads to

$$J_{\phi_i}(q_i)\dot{s}_i = \ddot{\tilde{x}} + \gamma_i\dot{\tilde{x}} - \dot{J}_{\phi_i}(q_i)s_i. \quad (6.19)$$

Using equations (6.19) and (6.16), and by letting $\gamma_i = \gamma$, equation (6.18) can be rewritten as

$$D_i'(q_i)\tilde{f}_i = \ddot{\tilde{x}} + \gamma\dot{\tilde{x}} + a_i\tilde{x} + b_i s_i - u_i. \quad (6.20)$$

where

$$\begin{aligned}D_i'(q_i) &= J_{\phi_i}(q_i)D_i^{-1}(q_i)J_{\phi_i}^T(q_i) && (6.21)\\ a_i &= D_i'(q_i)K_i\\ b_i &= \Big[J_{\phi_i}(q_i)D_i^{-1}(q_i)\left(C_i(q_i,\dot{q}_i) + K_{s_i}\right) - \dot{J}_{\phi_i}(q_i)\Big] && (6.22)\\ u_i &= J_{\phi_i}(q_i)D_i^{-1}(q_i)\Big[\hat{U}_i^1(q_i,\dot{q}_i|\tilde{\Theta}_i^1) + \hat{U}_i^2(q_i,\ddot{q}_i|\tilde{\Theta}_i^2) + \hat{U}_i^3(q_i,\ddot{q}_{r_i}|\tilde{\Theta}_i^3)\Big]. && (6.23)\end{aligned}$$

By computing $[D_1'(q_1)\tilde{f}_1 - D_i'(q_i)\tilde{f}_i]$ for $i = 2, \ldots, m$ and from equation (3.8), we obtain

$$R(q_1, \ldots, q_m)\tilde{f} = Q(t).$$

where $\tilde{f}^T = [\tilde{f}_1^T, \ldots, \tilde{f}_m^T]$, and $R(q_1, \ldots, q_m)$ and $Q(t)$ are defined as follows

$$R(q_1, \ldots, q_m) = \begin{bmatrix} D_1'(q_1) & -D_2'(q_2) & 0 & \cdots & 0 \\ D_1'(q_1) & 0 & -D_3'(q_3) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ D_1'(q_1) & 0 & 0 & \cdots & -D_m'(q_m) \\ I & I & I & \cdots & I \end{bmatrix}$$

$$Q(t) = \begin{bmatrix} (a_1 - a_2)\tilde{x} + (b_1 s_1 - b_2 s_2) - (u_2 - u_1) \\ (a_1 - a_3)\tilde{x} + (b_1 s_1 - b_3 s_3) - (u_3 - u_1) \\ \vdots \\ (a_1 - a_m)\tilde{x} + (b_1 s_1 - b_m s_m) - (u_m - u_1) \\ 0 \end{bmatrix}$$

From the structure of the matrix $R(q_1, \ldots, q_m)$ and since $D_i(q_i)$, $i = 1, \ldots, m$, is positive definite, it can be deduced that $R(q_1, \ldots, q_m)$ has full rank. The matrices $J_{\phi_i}(q_i)$ and $D_i^{-1}(q_i)$ are bounded. Thus, $R(q_1, \ldots, q_m)$ and $R^{-1}(q_1, \ldots, q_m)$ are also bounded. The matrices $C_i(q_i, \dot{q}_i)$ and $\dot{J}_{\phi_i}(q_i)$ are bounded for a bounded joint velocity $\dot{q}_i$ (justified later in this proof). Besides, s_i, $\tilde{x}$, and $\tilde{\Theta}_i^k$, for $k = 1, 2, 3$, are bounded. Hence, $Q(t)$ is bounded. Therefore, $\tilde{f} = R^{-1}(q_1, \ldots, q_m)Q(t)$ is bounded. From equation (6.12), the boundedness of $\dot{s}_i$ becomes straight forward. This proves that s_i is uniformly continuous. For a bounded s_i and $\tilde{x}$, it is directly implied by equation (4.18) that $\dot{\tilde{x}}$ is bounded. Then, $\tilde{x}$ is uniformly continuous. From (3.2), it can be deduced that a bounded $\dot{\tilde{x}}$ implies a bounded $\dot{\tilde{q}}_i$. Given the uniform continuousness of x_d, it can be deduced from (3.2) that the desired joint velocity

$\dot{q}_{d_i}$ is bounded. Thus, $\dot{q}_i$ is indeed bounded. This also justifies the boundedness of $C_i(q_i, \dot{q}_i)$ and $\dot{J}_{\phi_i}(q_i)$ claimed earlier in the proof. ■

Remark 6.4.2 *It is possible to reduce the effect of the error term $\epsilon_i(t)$ and enhance the robustness of the AFC by adding an additional term to equation (6.8). The robust control law would have the following form*

$$\tau_i = \hat{U}_i^1(q_i, \dot{q}_i|\Theta_i^1) + \hat{U}_i^2(q_i, \dot{q}_i|\Theta_i^2) + \hat{U}_i^3(q_i, \ddot{q}_{r_i}|\Theta_i^3) - K_{s_i} s_i - J_{\phi_i}^T(q_i)(K_i \tilde{x} + f_{d_i}) - W_i \, sgn(s_i), \quad (6.24)$$

where

$$W_i = diag(\bar{w}_{i1}, \dots, \bar{w}_{ik_i}),$$

$$\bar{w}_{ij} \geq \sup_t (|w_{ij}^1| + |w_{ij}^2| + |w_{ij}^3|), \quad j = 1, \dots, k_i,$$

and w_{ij}^k is the jth element of vector w_i^k, $k = 1, 2, 3$. This will ensure that $\dot{V}_{i3} \leq 0$. It is important to point out that, inspite of the robustness of control law (6.24), it is not preferred from a practical point of view since it leads to a chattering behavior of the manipulators. This makes the control of closed-chain robotic systems a very difficult task to achieve as it complicates even further the coordination between the manipulators.

Theorem 6.4.2 *The adaptive fuzzy controller (6.8) also leads to an asymptotic convergence of the force error $\tilde{f}_i$ to zero.*

Proof: Premultiplying equation (6.20) by $(D_i'(q_i))^{-1}$ results in

$$\tilde{f}_i = (D_i'(q_i))^{-1}(\ddot{\tilde{x}} + \gamma\dot{\tilde{x}}) + K_i \tilde{x} - (D_i'(q_i))^{-1} b_i s_i - (D_i'(q_i))^{-1} u_i. \quad (6.25)$$

where $D_i'(q_i)$, b_i, and u_i, are as defined in equations (6.21), (6.22), and (6.23), respectively. It is easy to see from equation (4.14) that matrix $D_i'(q_i)$ is invertible. Since, $\tilde{x}$ converges to zero, $\dot{\tilde{x}}$ and $\ddot{\tilde{x}}$ also converge to zero. Keeping in mind that s_i converges to zero and from equations (3.8) and (6.23),

$$\lim_{t\to\infty}\sum_{i=1}^{m}(D_i'(q_i))^{-1}J_{\phi_i}(q_i)D_i^{-1}(q_i)\left[\hat{U}_i^1(q_i,\dot{q}_i|\tilde{\Theta}_i^1)\right.$$
$$\left.+\hat{U}_i^2(q_i,\ddot{q}_i|\tilde{\Theta}_i^2)+\hat{U}_i^3(q_i,\ddot{q}_{r_i}|\tilde{\Theta}_i^3)\right]=0. \quad (6.26)$$

The joint velocity q_i, and hence the term $(D_i'(q_i))^{-1}J_{\phi_i}(q_i)D_i^{-1}(q_i)$, can be continuously changing in time and may assume an infinite number of possible values for $i=1,\ldots,m$. Since equation (6.26) has to hold for all those values, and since the term $(D_i'(q_i))^{-1}J_{\phi_i}(q_i)D_i^{-1}(q_i)$ is bounded, the only solution to equation (6.26) is to have

$$\lim_{t\to\infty}\left[\hat{U}_i^1(q_i,\dot{q}_i|\tilde{\Theta}_i^1)+\hat{U}_i^2(q_i,\ddot{q}_i|\tilde{\Theta}_i^2)+\hat{U}_i^3(q_i,\ddot{q}_{r_i}|\tilde{\Theta}_i^3)\right]=0.$$

Then, from equation (6.25),

$$\lim_{t\to\infty}\tilde{f}_i=0.$$

■

The enhanced position and internal force tracking capabilities of the proposed AFC in the face of parametric and unstructured uncertainties are illustrated in the following section.

6.5 Results and Discussion

In order to confirm the theoretical properties pertaining to the AFC's stability and tracking ability in the face of significant structured and unstructured uncertainties. three experiments are carried out on the cooperative robotic system described in Section 3.5. In each of them. the object mass is considered as the only parametric uncertainty of the AFC. This implicitly implies that the object's inertia represents another parametric uncertainty (see table 3.1.) The initial value of $\hat{\varphi}_i$. $i = 1.2$. is chosen to be zero. As for the AFC. it is assumed to have no apriori knowledge of the system's dynamics. The initial values of Θ_i^k are set to zero for $i = 1,2$ and $k = 1,2,3$. The universe of discourse for each fuzzy input is divided into five fuzzy labels. each defined by a Gaussian membership function. These membership functions are

$$
\begin{aligned}
\mu_{A_{i1}^{(l)}}(x_i) &= \frac{1}{1 + e^{(x_i + \pi/2)}} \\
\mu_{A_{i2}^{(l)}}(x_i) &= \frac{1}{e^{-(\frac{x_i + 3}{2})^2}} \\
\mu_{A_{i3}^{(l)}}(x_i) &= \frac{1}{e^{-(\frac{x_i}{2})^2}} \\
\mu_{A_{i4}^{(l)}}(x_i) &= \frac{1}{e^{-(\frac{x_i - 3}{2})^2}} \\
\mu_{A_{i5}^{(l)}}(x_i) &= \frac{1}{1 + e^{-(x_i - \pi/2)}}
\end{aligned}
$$

The membership functions are shown in figure 6.3.

The first experiment is meant to test the validity of the AFC in the case where the system's dynamics has no unstructured uncertainties or external disturbances (i.e.. α is set to zero in equations (3.16) and (3.17).) In other words. the model

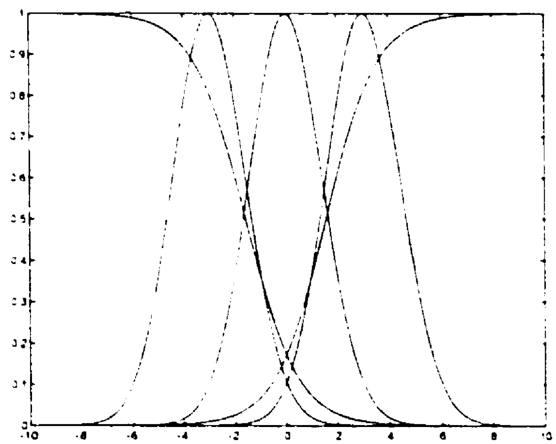


Figure 6.3: Membership Functions Used in the AFC.

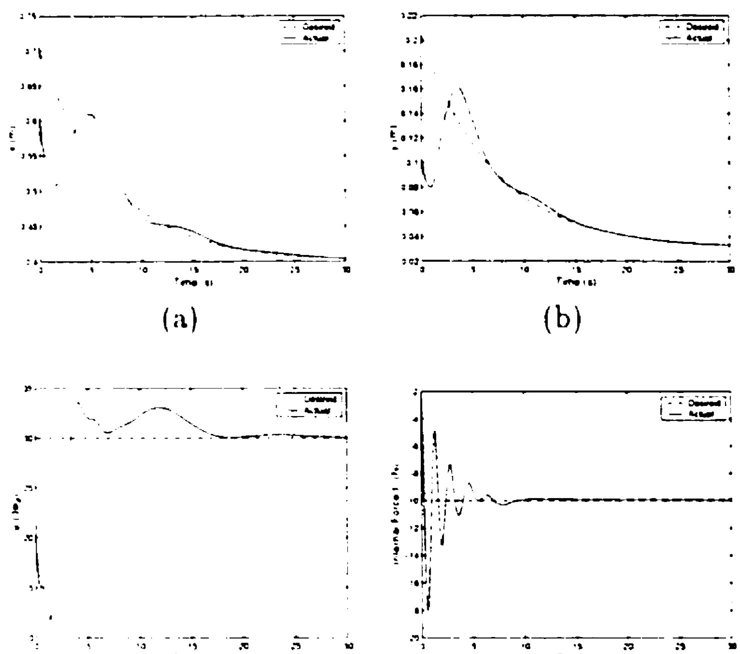
is assumed to be subject to parametric uncertainties only. The controller's tracking performance is shown in figure 6.4. Both, the position and the internal force tracking errors converge to zero. To compare the tracking ability of the AFC with its conventional counterpart, the CAC, the tracking errors of the two controllers are superimposed on each others as shown in figure 6.5. Although all the tracking errors decay to zero with both controllers, the AFC in this particular case shows a slower convergence rate than that of the CAC. In fact, this is expected from the AFC as it starts the learning of the robots overall dynamics without any initial prior knowledge on it (i.e., Θ_i^k, $i = 1, 2$, $k = 1, 2, 3$ are initially set to zero.) To compare the amount of efforts generated by both controllers, a performance index is introduced. Let $\tau_i^{(j)}(t)$, $j = 1, 2, 3$ and $i = 1, 2$, denote the torque at the jth joint of robot i at a time instant t. The amount of effort dissipated by joint j on robot i can then be measured by a performance index, $\beta_i^{(j)}$, defined as $\beta_i^{(j)} = \int_{t_0}^{t_f} [\tau_i^{(j)}(t)]^2 dt$, where t_0 and t_f are the time instants at which the robots motion starts and ends, respectively. Let β_i and β denote the total efforts consumed by robot i and by the two robots together, respectively. That is, $\beta_i = \sum_{j=1}^{3} \beta_i^{(j)}$, and $\beta = \beta_1 + \beta_2$. A comparison of the CAC's and AFC's effort performance indices is provided in table 6.1. As it can be seen, the AFC consumes less overall effort than the pure CAC. The computed torques in the two robots are illustrated in figure 6.6.

Table 6.1: Effort Indices of the CAC and the AFC in the Presence of Parametric Uncertainties Only ($\alpha = 0$.)

	$\beta_1^{(1)}$	$\beta_1^{(2)}$	$\beta_1^{(3)}$	β_1	$\beta_2^{(1)}$	$\beta_2^{(2)}$	$\beta_2^{(3)}$	β_2	β
CAC	360.20	272.20	8.87	641.35	414.68	404.86	36.93	856.47	1,497.82
AFC	357.01	247.80	9.66	614.48	415.20	318.38	18.29	751.87	1,366.35

The second experiment is carried out to study the tracking behavior of the AFC in the face of parametric and modeling uncertainties. For this purpose, an unmodeled external disturbance term is added to the manipulators dynamics in addition to the payload's mass uncertainty of the first experiment. This is achieved by setting α equal to 1 in equations (3.16) and (3.17). The AFC's tracking performance is provided in figure 6.7. As it is expected, and unlike the CAC, the AFC shows a satisfactory performance even when, both, structured and unstructured uncertainties coexist in the system's dynamics. Figure 6.8 provides a comparison of the tracking errors of both controllers. In this case, it is clear how the AFC outperforms its conventional counterpart, the CAC. This figure also shows that the tracking errors of the AFC do not converge to zero exactly but they converge rather to a certain envelope which is in the vicinity of zero. We believe that this is mainly due to the fact that the effect of the external disturbance factor $\rho(t)$ is no longer negligible. The time-varying term $\rho(t)$ does not depend on any of the AFC's inputs, and hence increasing α violates a close to zero value of $\epsilon_i(t)$, which in turn invalidates equation (6.16). Figure 6.9 shows the AFC's computed torques for the two robots.

A third experiment is carried out to compare the two controllers tracking performances under parametric uncertainties and different degrees of modeling uncertainties. This is achieved by letting α span a spectrum of values in the interval $[0, 2.5]$. The tracking errors of both controllers are plotted in figure 6.10. Again, the tracking superiority of the AFC over that of the CAC is very clear. The CAC shows very little tolerance to the increasing effect of the modeling uncertainty caused by the increase of α. On the other hand, the AFC proves to be more robust and it shows a higher flexibility in tolerating, both, structured and unstructured uncertainties. The AFC's tracking errors do not converge exactly to zero for relatively high values of α. This is mainly due to the higher impact of $\rho(t)$ on the system's dynamics for higher values of α, as explained earlier. However, even in the presence of significant modeling uncertainties, the AFC's tracking ability is still satisfactory as its tracking errors still converge to a region very close to zero. In practice, this would be very much tolerated.



(a) (b) (c) (d)

Figure 6.4: Tracking Performance of the AFC in the Presence of Parametric Uncertainties Only ($\alpha = 0$.)

CAC AFC

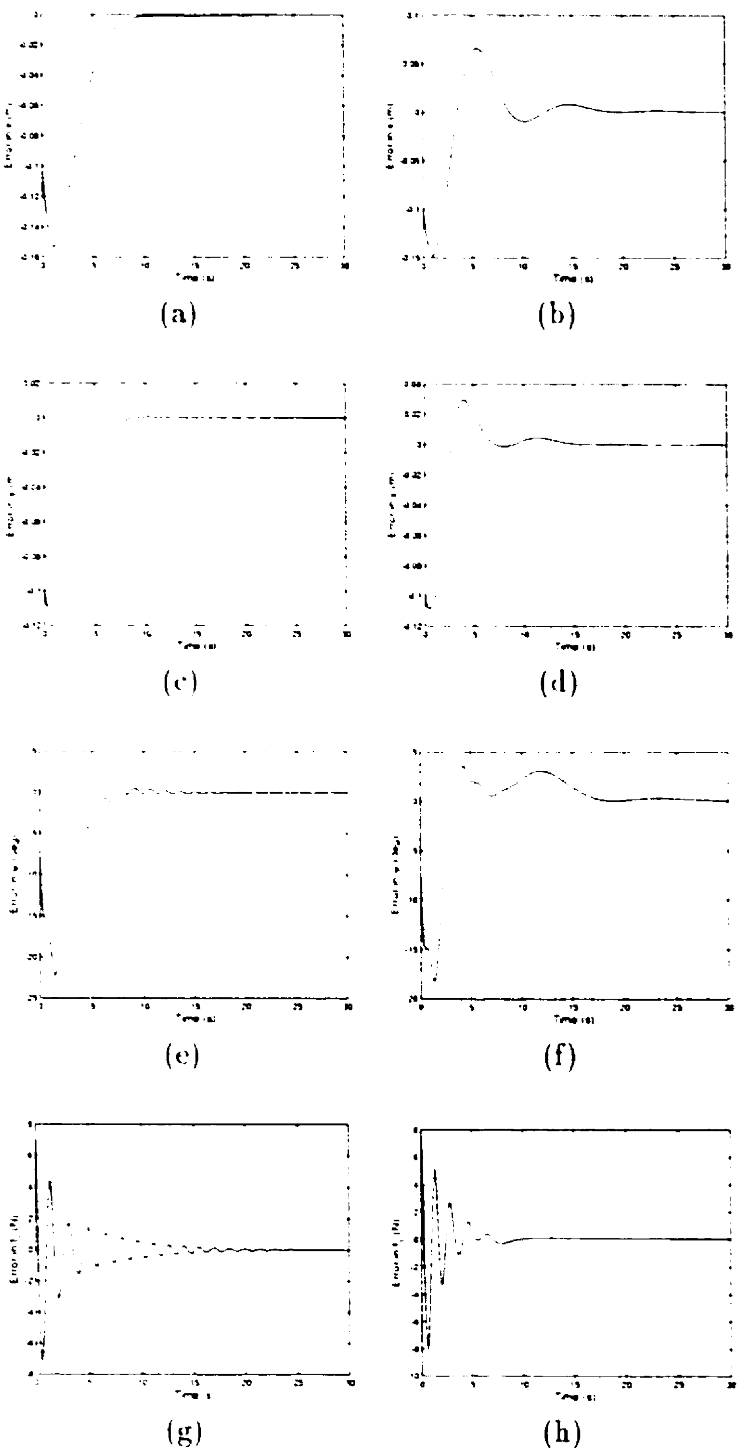


Figure 6.5: Tracking Errors of the CAC and the AFC in the Presence of Parametric Uncertainties Only ($\alpha = 0.$)

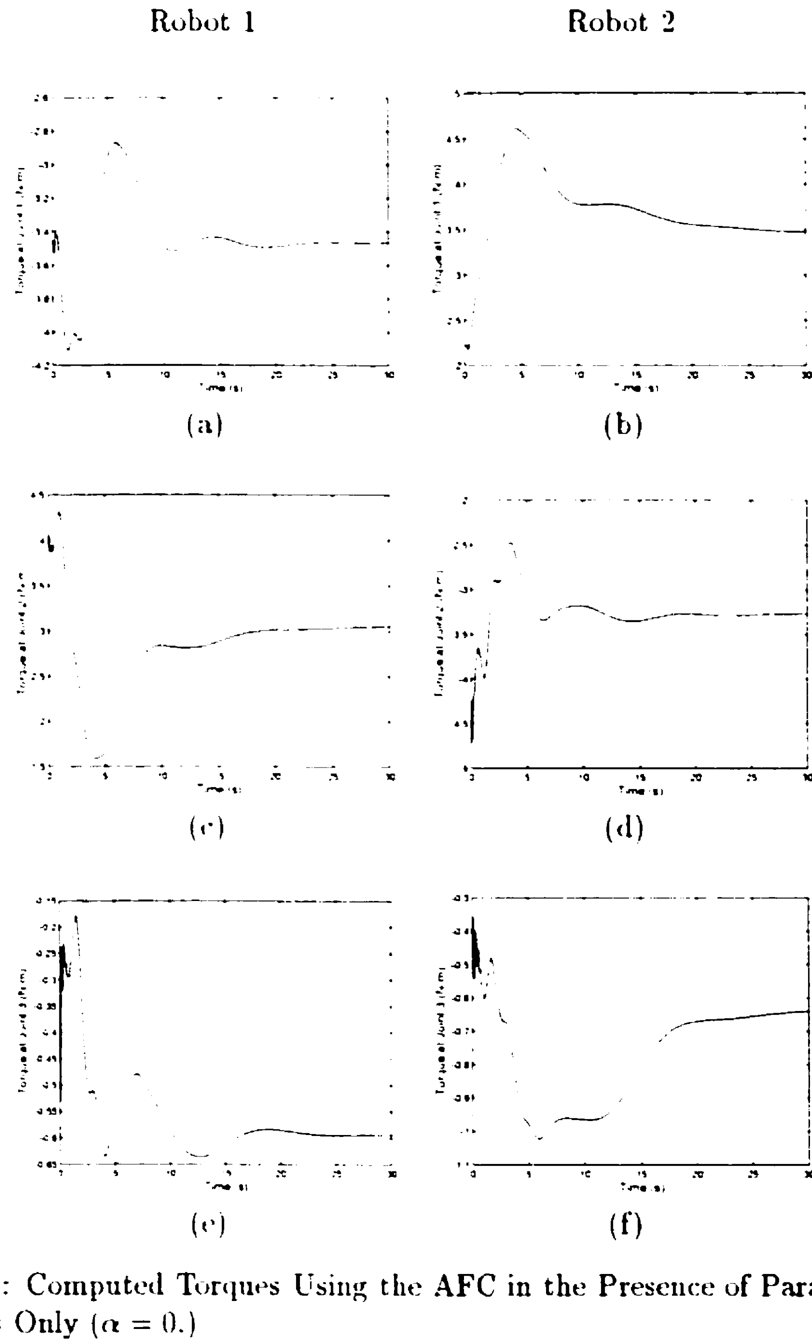


Figure 6.6: Computed Torques Using the AFC in the Presence of Parametric Uncertainties Only ($\alpha = 0$.)

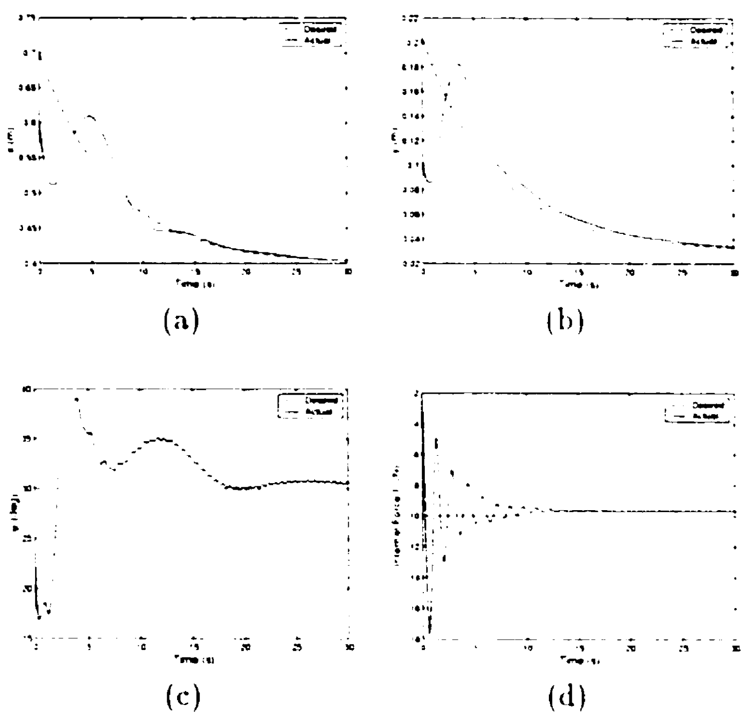


Figure 6.7: Tracking Performance of the AFC in the Presence of Parametric and Modeling Uncertainties ($\alpha = 1.$)

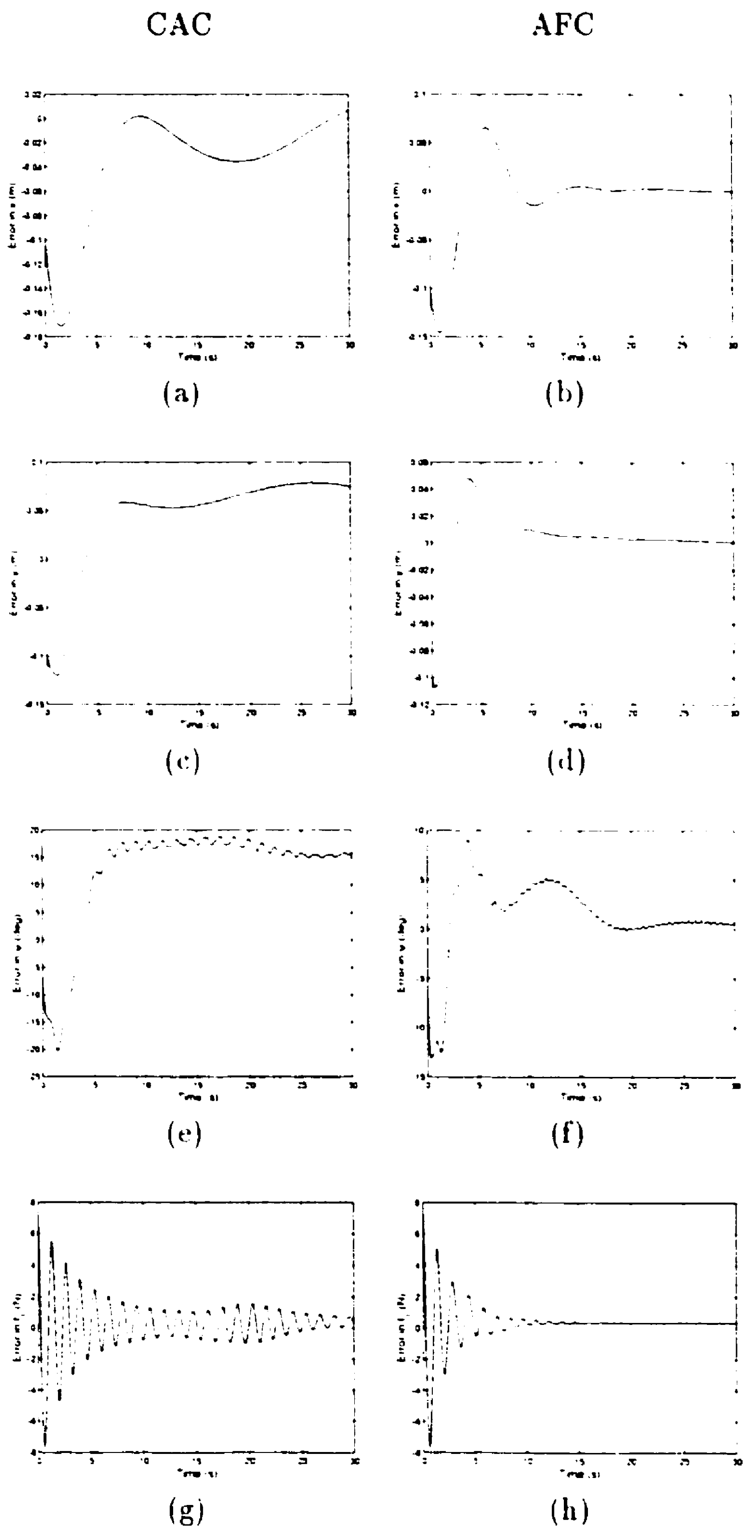


Figure 6.8: Tracking Errors of the CAC and the AFC in the Presence of Parametric and Modeling Uncertainties ($\alpha = 1.$)

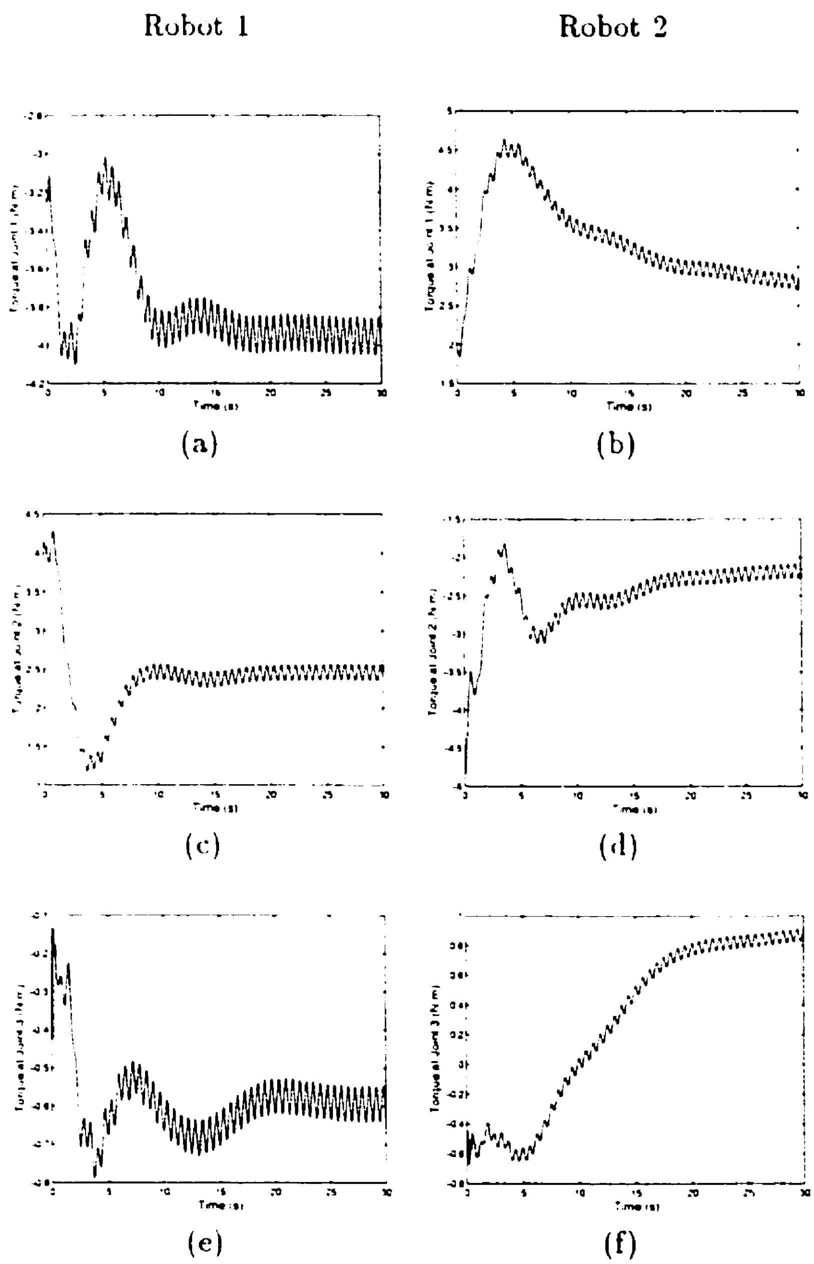


Figure 6.9: Computed Torques Using the AFC in the Presence of Parametric and Modeling Uncertainties ($\alpha = 1.$)

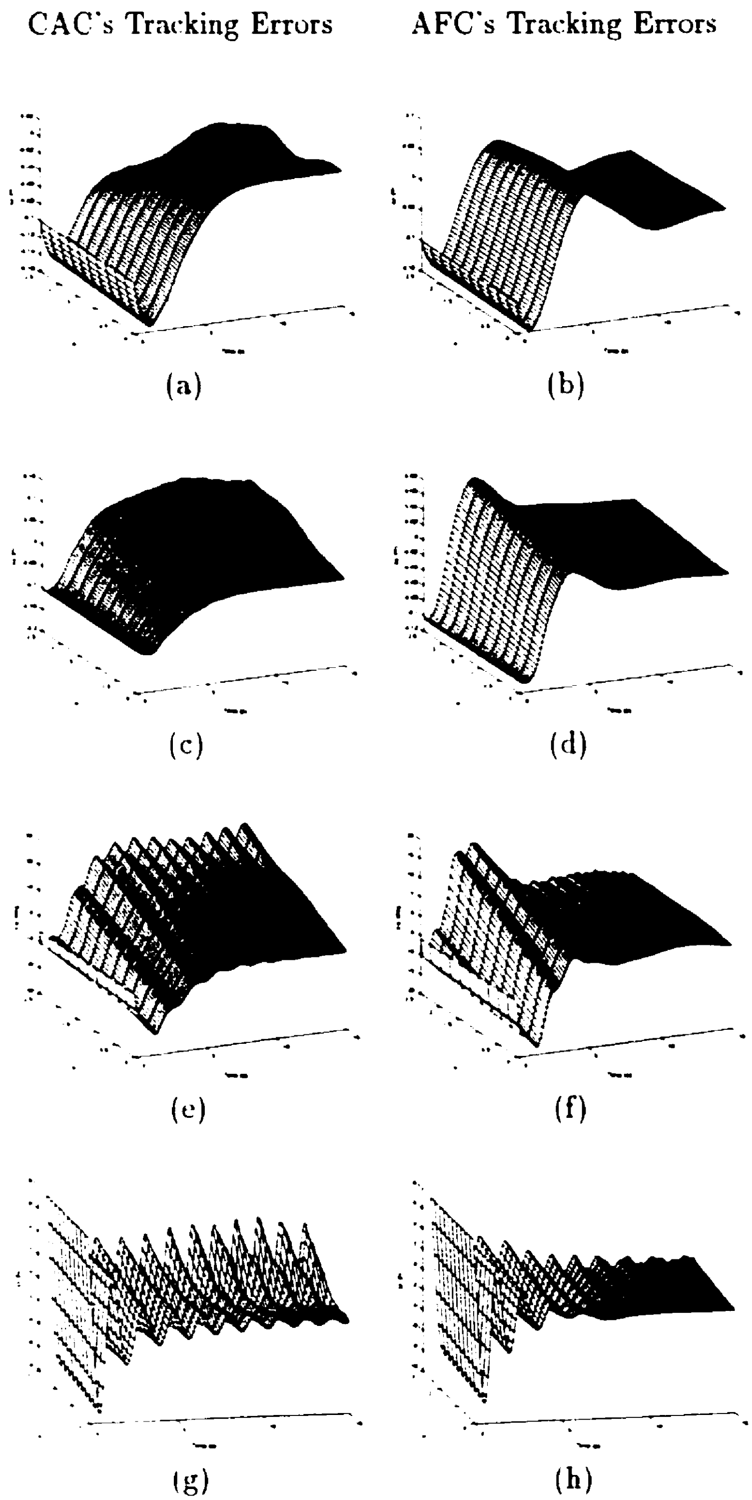


Figure 6.10: The Sensitivity of the CAC Versus that of the AFC under Parametric Uncertainties and Varying Intensities of Unstructured Uncertainties.

6.6 Conclusion

In this chapter, a customized decentralized direct adaptive fuzzy controller is proposed for the control of multiple strongly coupled manipulators handling a common object. The controller makes use of a MIMO fuzzy logic engine and an online adaptation scheme to fully assess and approximate the overall system's dynamics starting from no apriori knowledge of it. The proposed controller has been shown to be robust in the face of a substantial amount of parametric and modeling uncertainties with varying intensity levels. It has been proven that, under a few reasonable assumptions, even with the existence of these uncertainties in the system's dynamics, the position and the internal forces tracking errors always converge to zero. However, inspite of the considerable decrease in the number of fuzzy rules as a result of the rule decomposition scheme used, and the resulting significant computational time improvement of the adaptive fuzzy controller, it is still computationally slower than the conventional adaptive controller. This is one of the major tradeoffs of this particular adaptive scheme. A higher execution time efficiency based scheme is the subject of the next chapter, in which a novel approach based on a hybrid structure of a knowledge based adaptive controller is proposed.

Chapter 7

A Novel Hybrid Adaptive Scheme for the Control of CR

7.1 Introduction

In the previous chapter, a direct model-reference AFC has been proposed for the control of cooperative multiple manipulator systems. Although the proposed scheme has been shown to achieve desired control objectives in the face of significant structured and unstructured uncertainties, it still suffers from a few shortcomings:

- The AFC operates within the closed loop of the control system, which puts a substantial additional computational burden on the system.
- Although the rule-reduction mechanism used in the previous chapter significantly reduces the number of rules used by the fuzzy logic engine, this number may still be large considering that the AFC has to operate at the same high

bandwidth as the hardware devices located within the control loop.

- Although the AFC shows a high ability in learning the system's dynamics even with no prior partial knowledge of it, this might be still considered unrealistic. The main reason is that in the vast majority of real-life situations, a certain amount of information is usually available about the robots dynamics. In other words, the proposed AFC lacks the ability to incorporate the already acquired knowledge about the system's dynamics and hence does not have provision to include it for the controller advantage.

In this chapter, a new knowledge based controller is suggested to overcome the shortcomings of the previous AFC [34,37]. The controller proposed here is a hybrid soft computing based controller (HIC) that makes use of an innovative heirarchical structure to allow for a conventional adaptive controller and an adaptive fuzzy controller to operate simultaneously within a hierarchical framework to achieve the system's overall control objectives defined in Sections 1.3 and 3.1. The HIC is designed to maintain the high tracking ability of the previous AFC in the face of significant parametric and modeling uncertainties, while overcoming its drawbacks. The convergence and the stability properties of the suggested HIC are also formally synthesized using a Lyapunov stability approach. Computer simulation results are provided to confirm the theoretical properties of the HIC and to compare them with those of the pure CAC presented in Chapter 4 and the AFC presented in the previous chapter.

7.2 Hybrid Knowledge Based Controller Design

Soft computing tools are generally known to operate at a lower bandwith (expert level) than that of the control loop hardware [15]. Hence, for best possible results and to improve the systems reliability, soft computing based controllers have been traditionally implemendted within the upper layer of a hierachical structure. This is an excellent blend combining the benefits of hard controllers capabilities with those of soft computing tools in terms of supervision and learning [15]. A generic hierarchical architecture of an expert controller is shown in figure 7.1. The supervisory, or the expert, controller contributes to the overall closed-loop control by integrating and analyzing information fedback from the control system itself as well as other surrounding subsystems that may affect the plant's operating condition. In addition, the expert controller can even incorporate heuristic control knowledge about the system's desired behavior under various operating conditions. This can be achieved by either gathering data from offline experiments, or by gathering information in the form of if-then rules from a human control system operator who is expert in tuning the controller under these different operating points. The supervisory controller then integrates this information and uses its inference engine to decide on the appropriate update signal to be dispatched to the hard controller so that the overall closed-loop control system achieves its desired specifications. Among the merits of the hierarchical structure is the fact that the knowledge base at the higher level of the hierarchy can be updated at every new steady-state operating condition based on the current status of the process dynamics.

To improve the tracking performance of control law (4.17), a hybrid intelligent

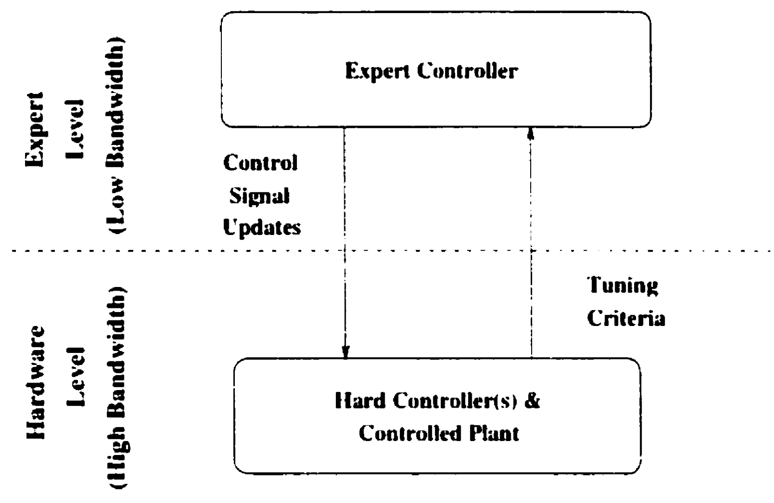


Figure 7.1: A Generic Hierarchical Structure of an Expert Controller.

controller (HIC) is developed. The proposed HIC consists of two different types of adaptive controllers operating simultaneously within a hierarchical framework. The hierarchical structure of the HIC consists of two layers. The execution (or the hard algorithmic) layer consists of a CAC similar in spirit to the one described by (4.19) but with some differences, a major one of which is that the CAC used here does not require a very precise model of the system's dynamics. A more detailed analysis is provided later in this section. The execution layer operates at a relatively higher bandwidth similar to the one at which hardware devices normally operate. The low computational complexity of CACs, in general, justifies the placement of the proposed CAC in this layer. The second (upper) layer is the expert layer and it contains the proposed direct AFC. The main task of this AFC is to compensate for the uncertainties of the CAC in the execution layer. Unlike many supervisory type of controllers discussed in the literature, which dictate updates to the parameters of the hard controller [15], the proposed AFC dispatches a control signal which is added to the torque generated by the CAC to enhance the overall tracking ability of the system. Due to the AFCs' high computational complexity as compared to that of CACs, the expert layer better operates at a lower bandwidth than that of the hardware layer to provide a higher computational efficiency of the overall HIC.

7.2.1 CAC Design

Unstructured uncertainties might not only be due to errors in modeling external disturbances, but might also be due to other factors such as errors in the regression matrix resulting from imperfect physical parameter values that are not represented in the adaptation vector φ_i, $i = 1, \ldots, m$, for instance. Let

$$\tilde{D}_i(q_i) \overset{\text{def}}{=} D_i(q_i) - D_i^*(q_i), \tag{7.1}$$

$$\tilde{C}_i(q_i, \dot{q}_i) \overset{\text{def}}{=} C_i(q_i, \dot{q}_i) - C_i^*(q_i, \dot{q}_i), \quad \text{and} \tag{7.2}$$

$$\tilde{G}_i(q_i) \overset{\text{def}}{=} G_i(q_i) - G_i^*(q_i), \tag{7.3}$$

be the modeling errors of $D_i(q_i)$, $C_i(q_i, \dot{q}_i)$, and $G_i(q_i)$, respectively, where $D_i^*(q_i)$, $C_i^*(q_i, \dot{q}_i)$, and $G_i^*(q_i)$ are their respective best possible approximations in the presence of these errors. Given such a partial knowledge of the system's dynamics, equation (3.15) takes now the following form

$$D_i^*(q_i)\ddot{q}_i + C_i^*(q_i, \dot{q}_i)\dot{q}_i + G_i^*(q_i) = Y_i^*(q_i, \dot{q}_i, \ddot{q}_i, c_i(t))\,\varphi_i^*, \quad i = 1, \ldots, m, \tag{7.4}$$

where $\varphi_i^* \in \mathbb{R}^{k_{\varphi_i^*}}$ is a vector of the physical parameters of the ith manipulator and the object, and $k_{\varphi_i^*} > 0$ is the length of that vector. The matrix $Y_i^*(q_i, \dot{q}_i, \ddot{q}_i, c_i(t)) \in \mathbb{R}^{k_i \times k_{\varphi_i^*}}$ is its corresponding regression matrix. Let $\hat{D}_i^*(q_i)$, $\hat{C}_i^*(q_i, \dot{q}_i)$, $\hat{G}_i^*(q_i)$, and

$\hat{\varphi}_i^*$ be the time-varying estimates of $D_i^*(q_i)$, $C_i^*(q_i, \dot{q}_i)$, $G_i^*(q_i)$, and φ_i^*, respectively. Then, from equation (7.4)

$$Y_i^*(q_i, \dot{q}_i, \ddot{q}_i, c_i(t))\,\hat{\varphi}_i^* = \hat{D}_i^*(q_i)\ddot{q}_i + \hat{C}_i^*(q_i, \dot{q}_i)\dot{q}_i + \hat{G}_i^*(q_i), \quad i = 1, \ldots, m,$$

and hence, equation (4.20) can be reformulated as

$$\hat{D}_i^*(q_i)\ddot{q}_{r_i} + \hat{C}_i^*(q_i, \dot{q}_i)\dot{q}_{r_i} + \hat{G}_i^*(q_i) = Y_i^*(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t))\,\hat{\varphi}_i^*, \quad i = 1, \ldots, m. \quad (7.5)$$

The reference joint velocity, $\dot{q}_{r_i}$, and acceleration, $\ddot{q}_{r_i}$, for robot i, are as defined in equations (4.13) and (4.15), respectively. The control law of the CAC for the ith manipulator becomes then defined as

$$\tau_i^{(c)} = Y_i^*(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t))\,\hat{\varphi}_i^* - K_{s_i}s_i - J_{\phi_i}^T(q_i)(K_i\tilde{x} + f_{d_i}) - \hat{\tau}_{d_i}, \quad (7.6)$$

with the adaptation law

$$\dot{\hat{\varphi}}_i^* = -\Gamma_i^{-1}Y_i^{*T}(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t))s_i, \quad (7.7)$$

where K_{s_i}, K_i, and Γ_i, are positive definite gain matrices, $\hat{\tau}_{d_i}$ is the time-varying estimate of τ_{d_i}, and s_i is the sliding mode parameter defined in (4.16).

7.2.2 AFC Design

To compensate for the effects of the uncertainties on the CAC's tracking performance, a direct AFC is designed to approximate the overall error in the system's dynamics. For each robot i, the error can be expressed as an unknown nonlinear function

$$E_i = \tilde{D}_i(q_i)\ddot{q}_{r_i} + \tilde{C}_i(q_i, \dot{q}_i)\dot{q}_{r_i} + \tilde{G}_i(q_i) + \tilde{\tau}_{d_i}. \quad (7.8)$$

The external disturbance vector can always be written in the following form

$$\tau_{d_i} = T_{d_i}(q_i, \dot{q}_i) + \varepsilon_i(t),$$

where $T_{d_i}(q_i, \dot{q}_i)$ is a time-varying function that depends solely on $\{q_i, \dot{q}_i\}$, and $\varepsilon_i(t)$ is a function that depends on the time t. The time-varying estimate of τ_{d_i}, $\hat{\tau}_{d_i}$ can then be expressed as

$$\hat{\tau}_{d_i} = \hat{T}_{d_i}(q_i, \dot{q}_i) + \hat{\varepsilon}_i(t),$$

where $\hat{T}_{d_i}(q_i, \dot{q}_i)$ and $\hat{\varepsilon}_i(t)$ are the time-varying estimates of $T_{d_i}(q_i, \dot{q}_i)$ and $\varepsilon_i(t)$, respectively. Hence, the estimation error of τ_{d_i} is formulated as

$$\tilde{\tau}_{d_i} = \tilde{T}_{d_i}(q_i, \dot{q}_i) + \tilde{\varepsilon}_i(t), \tag{7.9}$$

where

$$\tilde{T}_{d_i}(q_i, \dot{q}_i) \stackrel{\text{def}}{=} \hat{T}_{d_i}(q_i, \dot{q}_i) - T_{d_i}(q_i, \dot{q}_i), \quad \text{and}$$
$$\tilde{\varepsilon}_i(t) \stackrel{\text{def}}{=} \hat{\varepsilon}_i(t) - \varepsilon_i(t).$$

For the rest of this synthesis, we assume that $\tilde{\varepsilon}_i(t)$ is insignificant so that $\tilde{\varepsilon}_i(t)$ is dominantly dependent on $\tilde{T}_{d_i}(q_i, \dot{q}_i)$.

Assumption 7.2.1 *The estimation error $\tilde{\varepsilon}_i(t)$ is in the vicinity of zero. In other words,*

$$\tilde{\varepsilon}_i(t) \approx 0.$$

This is a realistic assumption as it can be achieved using a disturbance attenuation technique, similar to the one discussed in [39, 114] to compute for $\hat{\varepsilon}_i(t)$. Equation (7.8) can now be reformulated as

$$E_i = \bar{D}_i(q_i)\ddot{q}_{r_i} + \bar{C}_i(q_i,\dot{q}_i)\dot{q}_{r_i} + \bar{G}_i(q_i) + \bar{T}_{d_i}(q_i,\dot{q}_i) + \bar{\varepsilon}_i(t).$$

To ensure that $D_i(q_i)$, $C_i(q_i,\dot{q}_i)$, and $G_i(q_i)$, can be actually approximated by FLCs, the following assumption is adopted

Assumption 7.2.2 *The load distribution matrix $c_i(t)$ is either dependent on q_i and $\dot{q}_i$ only, or is slowly varying in time. In other words, for $i = 1,\ldots,m$,*

$$c_i(t) = c_i(q_i,\dot{q}_i), \quad or$$

$$\dot{c}_i(t) \approx 0.$$

Note that a constant matrix $c_i(t)$ satisfies assumption 7.2.2. This is very common indeed in real world applications. Neglecting the effect of $\bar{\varepsilon}_i(t)$ and according to the universal approximation theorem (theorem 5.4.1), E_i can be approximated by a MIMO fuzzy logic system $\hat{U}_i(q_i,\dot{q}_i,\dot{q}_{r_i},\ddot{q}_{r_i}|\Theta_i)$ which is in the form defined by equations (5.2), (5.3), (5.4), and (5.5). The FLC $\hat{U}_i(q_i,\dot{q}_i,\dot{q}_{r_i},\ddot{q}_{r_i}|\Theta_i)$ is directly dependent on the parameters $\{q_i,\dot{q}_i,\dot{q}_{r_i},\ddot{q}_{r_i}\}$ as they are the only time-varying quantities of the ith manipulator.

7.3 Rule Reduction Technique

The input vector $(q_i,\dot{q}_i,\dot{q}_{r_i},\ddot{q}_{r_i})^T$ of $\hat{U}_i(q_i,\dot{q}_i,\dot{q}_{r_i},\ddot{q}_{r_i}|\Theta_i)$ is composed of $(4k_i)$ real elements. So, if κ_i fuzzy labels are assigned to each of these elements, the total number of fuzzy rules, L_i, in $\hat{U}_i(q_i,\dot{q}_i,\dot{q}_{r_i},\ddot{q}_{r_i}|\Theta_i)$ is $(\kappa_i)^{4k_i}$.This could become a very large number in many real-life robotic applications, knowing that such a number of fuzzy rules has to be fired simultaneously for each single robot. For instance, if κ_i and k_i are assigned the values 3 and 2, respectively, each FLC $\hat{U}_i(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}|\Theta_i)$, for $i = 1, \ldots, m$, needs to fire 3^8 $(= 6,561)$ rules in order to compute and dispatch the control signals to the k_i actuators of each manipulator. Bearing in mind the nonlinear nature of the fuzzy rules and the relatively simple configuration of the manipulators considered in this example, the practicality of such a type of FLCs becomes questionable.

To alleviate the computational complexity of $\hat{U}_i(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}|\Theta_i)$, a rule reduction algorithm is proposed. A decomposition procedure is suggested to aggregate $\hat{U}_i(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}|\Theta_i)$ into two MIMO FLCs. Nominally,

$$\hat{U}_i(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}|\Theta_i) = \hat{U}_i^1(q_i, \ddot{q}_{r_i}) + \hat{U}_i^2(q_i, \dot{q}_i, \dot{q}_{r_i}),$$

where the FLCs $\hat{U}_i^1(q_i, \ddot{q}_{r_i})$ and $\hat{U}_i^2(q_i, \dot{q}_i, \dot{q}_{r_i})$ are to approximate $\{\bar{D}_i(q_i)\ddot{q}_{r_i}\}$ and $\{\bar{C}_i(q_i, \dot{q}_i)\dot{q}_{r_i} + \bar{G}_i(q_i) + \bar{T}_{d_i}(q_i, \dot{q}_i)\}$, respectively. The term $\{\bar{D}_i(q_i)\ddot{q}_{r_i}\}$ can be expressed as

$$\bar{D}_i(q_i)\ddot{q}_{r_i} = \sum_{j=1}^{k_i} \bar{D}_{ij}(q_i)\ddot{q}_{r_{ij}}, \qquad (7.10)$$

where $\bar{D}_{ij}(q_i)$ is the jth column of matrix $\bar{D}_i(q_i)$ and $\ddot{q}_{r_{ij}}$ is the jth element of vector $\ddot{q}_{r_i}$. Based on the universal approximation theorem and since $\bar{D}_{ij}(q_i)$ is an unknown nonlinear vector, it can be approximated by a MIMO FLC, $\hat{U}_{ij}^1(q_i|\Theta_{ij}^1)$, in the form of (5.2), (5.3), (5.4), and (5.5). That is,

$$\hat{U}_{ij}^1(q_i|\Theta_{ij}^1) = (\Theta_{ij}^1)^T\xi_i^1(q_i) = \begin{bmatrix} (\Theta_{ij}^1)_1^T\xi_i^1(q_i) \\ (\Theta_{ij}^1)_2^T\xi_i^1(q_i) \\ \vdots \\ (\Theta_{ij}^1)_{k_i}^T\xi_i^1(q_i) \end{bmatrix}.$$

with $\Theta_{ij}^1 \in \mathbb{R}^{L_i^1 \times k_i}$, $\xi_i^1(q_i) \in \mathbb{R}^{L_i^1}$, and $(\Theta_{ij}^1)_k$ being the kth column of Θ_{ij}^1. The parameter L_i^1 denotes the number of fuzzy rules in $\hat{U}_{ij}^1(q_i|\Theta_{ij}^1)$. The consequent parameter matrix Θ_{ij}^1 is tuned according to the following adaptation law

$$\dot{\Theta}_{ij}^1 = -(\Gamma_{ij}^1)^{-1}\xi_i^1(q_i)s_i^T\ddot{q}_{r_{ij}}, \quad i = 1,\dots,m, \ j = 1,\dots,k_i. \tag{7.11}$$

where $\Gamma_{ij}^1 \in \mathbb{R}^{L_i^1 \times L_i^1}$ is a positive definite gain matrix. If κ_i fuzzy labels is assigned to each element of q_i, then the total number of rules of $\hat{U}_{ij}^1(q_i|\Theta_{ij}^1)$ is given by $\sum_{j=1}^{k_i} L_i^1 = \sum_{j=1}^{k_i}(\kappa_i)^{k_i} = k_i(\kappa_i)^{k_i}$.

Since $\{q_i, \dot{q}_i, \dot{q}_{r_i}\}$ are the only time-varying parameters in $\{\bar{C}_i(q_i,\dot{q}_i)\dot{q}_{r_i} + \bar{G}_i(q_i) + \bar{T}_{d_i}(q_i,\dot{q}_i)\}$, the existence of a MIMO FLC $\hat{U}_i^2(q_i,\dot{q}_i,\dot{q}_{r_i})$ to approximate such a nonlinear function with those parameters as its unique inputs, is directly implied by the universal approximation theorem. Using the recursive Newton-Euler method, $\dot{q}_{r_i}$ can be replaced with $\dot{q}_i$, leaving $\hat{U}_i^2$ with two inputs only, q_i and $\dot{q}_i$. In other words, $\{\bar{C}_i(q_i,\dot{q}_i)\dot{q}_{r_i} + \bar{G}_i(q_i) + \bar{T}_{d_i}(q_i,\dot{q}_i)\}$ can be approximated by the FLC $\hat{U}_i^2(q_i,\dot{q}_i|\Theta_i^2)$ in the form of (5.2), (5.3), (5.4), and (5.5), with $\Theta_i^2 \in \mathbb{R}^{L_i^2 \times k_i}$ and L_i^2 being the number of fuzzy rules of $\hat{U}_i^2(q_i,\dot{q}_i|\Theta_i^2)$. The FLC $\hat{U}_i^2(q_i,\dot{q}_i|\Theta_i^2)$ can then be expressed in the form of

$$\hat{U}_i^2(q_i,\dot{q}_i|\Theta_i^2) = (\Theta_i^2)^T\xi_i^2(q_i,\dot{q}_i).$$

where $\xi_i^2(q_i, \dot{q}_i) \in \mathbb{R}^{L_i^2}$. The adaptation law of $\hat{U}_i^2(q_i, \dot{q}_i|\Theta_i^2)$ is defined by

$$\dot{\Theta}_i^2 = -(\Gamma_i^2)^{-1}\xi_i^2(q_i, \dot{q}_i)s_i^T. \tag{7.12}$$

where $\Gamma_i^2 \in \mathbb{R}^{L_i^2 \times L_i^2}$ is a positive definite gain matrix. Note that with κ_i fuzzy labels assigned to each element of q_i and $\dot{q}_i$, $L_i^2 = (\kappa_i)^{2k_i}$. Based on the above results, the torque offset generated by the AFC of the ith manipulator is given by

$$\tau_i^{(f)} = \sum_{j=1}^{k_i} \hat{U}_{ij}^1(q_i|\Theta_{ij}^1)\ddot{q}_{r_{ij}} + \hat{U}_i^2(q_i, \dot{q}_i|\Theta_i^2). \tag{7.13}$$

The overall control law of the HIC then becomes

$$\tau_i = \tau_i^{(c)} + \tau_i^{(f)}. \tag{7.14}$$

Figures 7.2 and 7.3 depict the block diagrams of the HIC. It is important to point out that using control law (7.14) with κ_i fuzzy labels for each element of q_i and $\dot{q}_i$, the total number of fuzzy rules, L_i, that has to be fired by the HIC at each robot i becomes $L_i = k_i(\kappa_i)^{k_i} + (\kappa_i)^{2k_i}$. This is a significant improvement compared to the original number (i.e., $(\kappa_i)^{4k_i}$) and it gives a major boost in terms of computational efficiency which would facilitate its real world implementation. Going back to the previous example with $\kappa_i = 3$ and $k_i = 2$, the value of L_i is now reduced to $(2(3)^2 + (3)^4) = 99$, compared with $6,561$ with the earlier scheme.

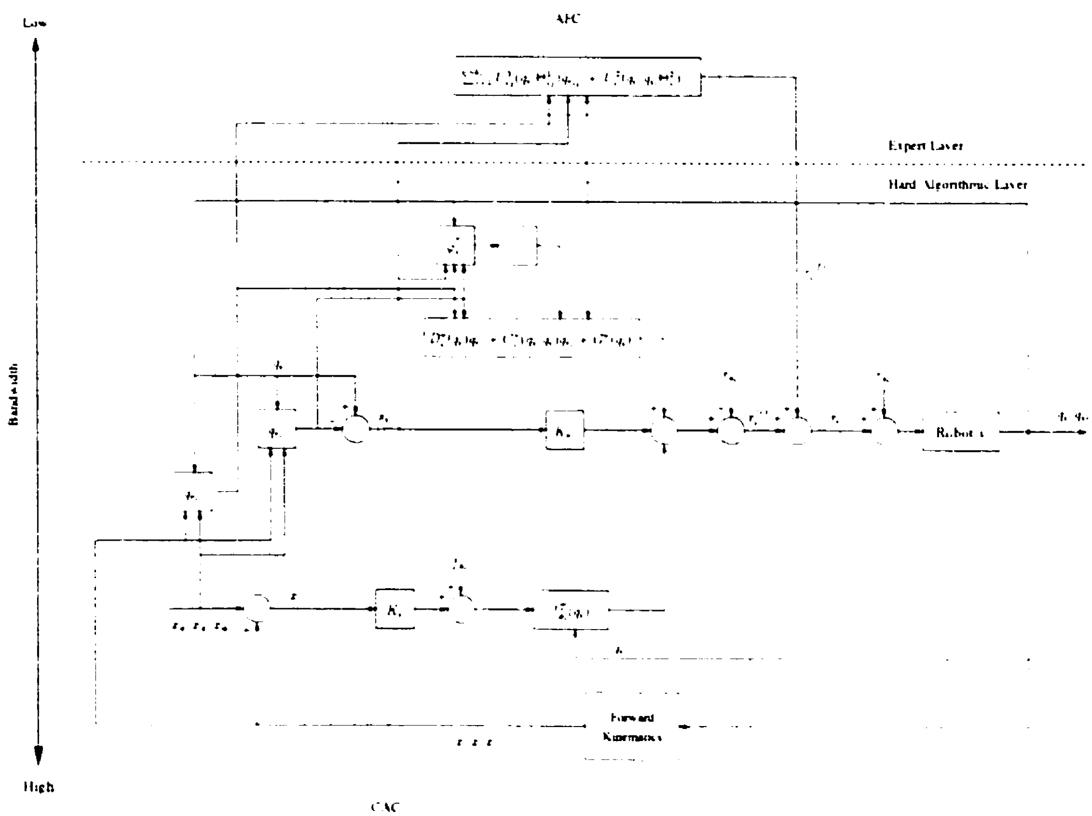


Figure 7.2: Block Diagram of the HIC Architecture.

7.4 Stability Analysis

Let $\tilde{U}^1_{ij}$ and $\tilde{U}^2_i$ denote the errors in approximating $\bar{D}_{ij}(q_i)$ and $\{\bar{C}_i(q_i, \dot{q}_i)\dot{q}_{r_i} + \bar{G}_i(q_i) + \bar{\tau}_{d_i}\}$, respectively. That is

$$\tilde{U}^1_{ij} = \hat{U}^1_{ij}(q_i|\Theta^1_{ij}) - \bar{D}_{ij}(q_i), \quad \text{and} \tag{7.15}$$

$$\tilde{U}^2_i = \hat{U}^2_i(q_i, \dot{q}_i|\Theta^2_i) - [\bar{C}_i(q_i, \dot{q}_i)\dot{q}_{r_i} + \bar{G}_i(q_i) + \bar{\tau}_{d_i}]. \tag{7.16}$$

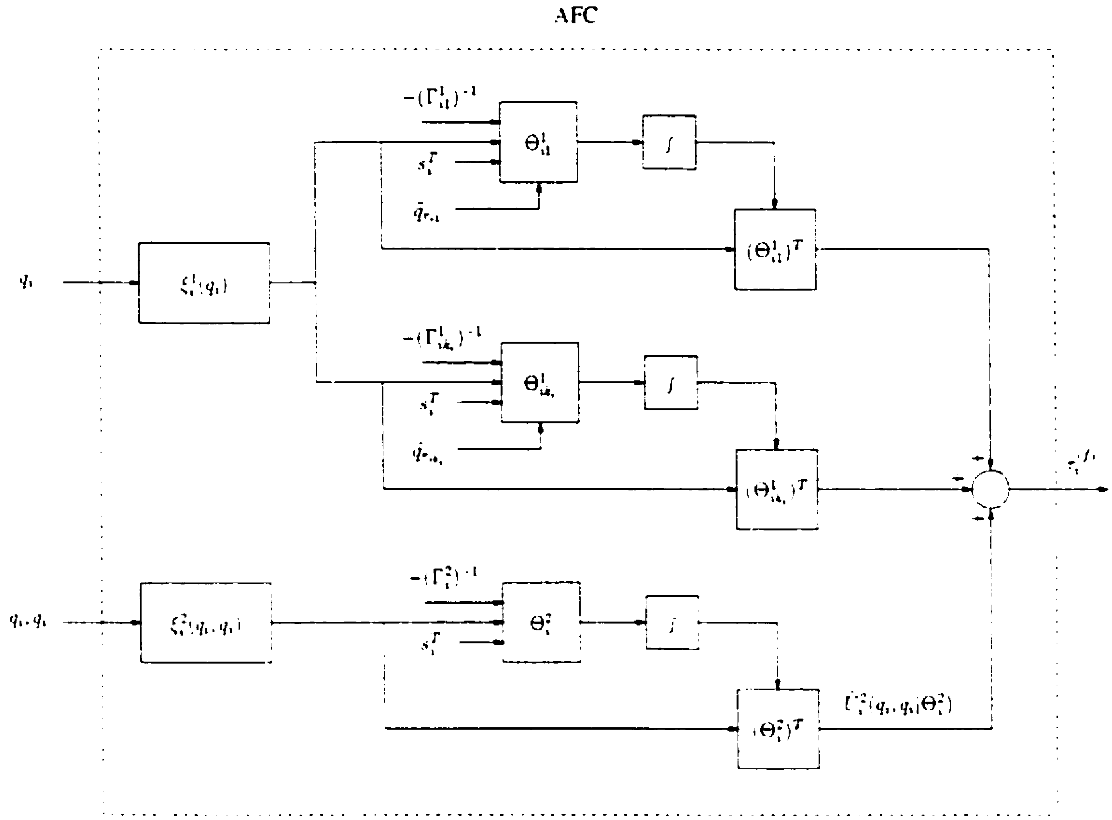


Figure 7.3: Block Diagram of the AFC.

If Θ_{ij}^{1*} and Θ_i^{2*} are the optimal parameter matrices for $\hat{U}_{ij}^1(q_i|\Theta_{ij}^1)$ and $\hat{U}_i^2(q_i,\dot{q}_i|\Theta_i^2)$, respectively, then the minimum approximation errors of the two FLCs can be expressed as

$$
\begin{aligned}
w_{ij}^1 &= \hat{U}_{ij}^1(q_i|\Theta_{ij}^{1*}) - \bar{D}_{ij}(q_i), \quad \text{and} \\
w_i^2 &= \hat{U}_i^2(q_i,\dot{q}_i|\Theta_i^{2*}) - [\bar{C}_i(q_i,\dot{q}_i)\dot{q}_{r_i} + \bar{G}_i(q_i) + \bar{T}_{d_i}(q_i,\dot{q}_i)].
\end{aligned}
$$

Using equation (7.9), equations (7.15) and (7.16) can then be reformulated as

$$\tilde{U}^1_{ij} = \hat{U}^1_{ij}(q_i|\Theta^1_{ij}) - \hat{U}^1_{ij}(q_i|\Theta^{1*}_{ij}) + w^1_{ij}, \quad \text{and} \tag{7.17}$$

$$\tilde{U}^2_i = \hat{U}^2_i(q_i, \dot{q}_i|\Theta^2_i) - \hat{U}^2_i(q_i, \dot{q}_i|\Theta^{2*}_i) + w^2_i - \bar{\varepsilon}_i(t). \tag{7.18}$$

Let $\hat{U}^1_{ij}(q_i|\tilde{\Theta}^1_{ij})$ and $\hat{U}^2_i(q_i, \dot{q}_i|\tilde{\Theta}^2_i)$ be defined as

$$\hat{U}^1_{ij}(q_i|\tilde{\Theta}^1_{ij}) \stackrel{\text{def}}{=} \hat{U}^1_{ij}(q_i|\Theta^1_{ij}) - \hat{U}^1_{ij}(q_i|\Theta^{1*}_{ij}) \tag{7.19}$$

$$\hat{U}^2_i(q_i, \dot{q}_i|\tilde{\Theta}^2_i) \stackrel{\text{def}}{=} \hat{U}^2_i(q_i, \dot{q}_i|\Theta^2_i) - \hat{U}^2_i(q_i, \dot{q}_i|\Theta^{2*}_i). \tag{7.20}$$

Then, using equation (5.4), equations (7.17) and (7.18) can be expressed as

$$\tilde{U}^1_{ij} \stackrel{\text{def}}{=} \hat{U}^1_{ij}(q_i|\tilde{\Theta}^1_{ij}) + w^1_{ij} = (\tilde{\Theta}^1_{ij})^T \xi^1_i(q_i) + w^1_{ij}, \quad \text{and} \tag{7.21}$$

$$\tilde{U}^2_i \stackrel{\text{def}}{=} \hat{U}^2_i(q_i, \dot{q}_i|\tilde{\Theta}^2_i) + w^2_i - \bar{\varepsilon}_i(t) = (\tilde{\Theta}^2_i)^T \xi^2_i(q_i, \dot{q}_i) + w^2_i - \bar{\varepsilon}_i(t), \tag{7.22}$$

where

$$\tilde{\Theta}^1_{ij} = \Theta^1_{ij} - \Theta^{1*}_{ij}, \quad \text{and}$$

$$\tilde{\Theta}^2_i = \Theta^2_i - \Theta^{2*}_i.$$

Lemma 7.4.1 *Using the control law (7.14), the overall system's error dynamics equation is given by*

$$D_i(q_i)\dot{s}_i + C_i(q_i, \dot{q}_i)s_i - J^T_{\phi_i}(q_i)\tilde{f}_i = Y^*_i(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t))\tilde{\varphi}^*_i$$
$$+ \sum_{j=1}^{k_i} \hat{U}^1_{ij}(q_i|\tilde{\Theta}^1_{ij})\ddot{q}_{r_{ij}} + \hat{U}^2_i(q_i, \dot{q}_i|\tilde{\Theta}^2_i) - K_{s_i}s_i - J^T_{\phi_i}(q_i)K_i\tilde{x} + \epsilon_i(t). \tag{7.23}$$

where

$$\tilde{f}_i \stackrel{\text{def}}{=} f_i - f_{d_i},$$

$$\tilde{\varphi}_i^* \stackrel{\text{def}}{=} \hat{\varphi}_i^* - \varphi_i^*, \quad and$$

$$\epsilon_i(t) \stackrel{\text{def}}{=} \sum_{j=1}^{k_i} w_{ij}^1 \ddot{q}_{r_{ij}} + w_i^2 - \bar{\varepsilon}_i(t). \tag{7.24}$$

Proof: Substituting equations (7.6) and (7.13) into (7.14) and equating to the dynamics equation (3.12) yields

$$D_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + G_i(q_i) - J_{\phi_i}^T(q_i)f_i - \tau_{d_i} = Y_i^*(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t))\hat{\varphi}_i^* - K_{s_i}s_i$$
$$-J_{\phi_i}^T(q_i)(K_i\tilde{x} + f_{d_i}) - \hat{\tau}_{d_i} + \sum_{j=1}^{k_i} \hat{U}_{ij}^1(q_i|\Theta_{ij}^1)\ddot{q}_{r_{ij}} + \hat{U}_i^2(q_i, \dot{q}_i|\Theta_i^2).$$

From equations (7.1), (7.2), (7.3), and (4.16), we get

$$(D_i^*(q_i) + \bar{D}_i(q_i))(\dot{s}_i + \ddot{q}_{r_i}) + (C_i^*(q_i, \dot{q}_i) + \bar{C}_i(q_i, \dot{q}_i))(s_i + \dot{q}_{r_i}) + (G_i^*(q_i) + \bar{G}_i(q_i))$$
$$-J_{\phi_i}^T(q_i)f_i - \tau_{d_i} = Y_i^*(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t))\hat{\varphi}_i^* - K_{s_i}s_i - J_{\phi_i}^T(q_i)(K_i\tilde{x} + f_{d_i})$$
$$-\hat{\tau}_{d_i} + \sum_{j=1}^{k_i} \hat{U}_{ij}^1(q_i|\Theta_{ij}^1)\ddot{q}_{r_{ij}} + \hat{U}_i^2(q_i, \dot{q}_i|\Theta_i^2).$$

This leads to

$$D_i(q_i)\dot{s}_i + C_i(q_i, \dot{q}_i)s_i - J_{\phi_i}^T(q_i)\tilde{f}_i = Y_i^*(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t))\hat{\varphi}_i^*$$
$$-(D_i^*(q_i)\ddot{q}_{r_i} + C_i^*(q_i, \dot{q}_i)\dot{q}_{r_i} + G_i^*(q_i)) + \sum_{j=1}^{k_i} \hat{U}_{ij}^1(q_i|\Theta_{ij}^1)\ddot{q}_{r_{ij}} + \hat{U}_i^2(q_i, \dot{q}_i|\Theta_i^2)$$
$$-(\bar{D}_i(q_i)\ddot{q}_{r_i} + \bar{C}_i(q_i, \dot{q}_i)\dot{q}_{r_i} + \bar{G}_i(q_i)) - \hat{\tau}_{d_i} + \tau_{d_i} - K_{s_i}s_i - J_{\phi_i}^T(q_i)K_i\tilde{x}.$$

Substituting equations (7.5) and (7.9) results in

$$D_i(q_i)\dot{s}_i + C_i(q_i, \dot{q}_i)s_i - J^T_{\phi_i}(q_i)\tilde{f}_i = Y_i^*(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t))\tilde{\varphi}_i^* + \sum_{j=1}^{k_i} \hat{U}^1_{ij}(q_i|\Theta^1_{ij})\ddot{q}_{r_{ij}}$$

$$+\hat{U}^2_i(q_i, \dot{q}_i|\Theta^2_i) - (\bar{D}_i(q_i)\ddot{q}_{r_i} + \bar{C}_i(q_i, \dot{q}_i)\dot{q}_{r_i} + \bar{G}_i(q_i) + \bar{T}_{d_i}(q_i, \dot{q}_i))$$

$$-\bar{\varepsilon}_i(t) - K_{s_i}s_i - J^T_{\phi_i}(q_i)K_i\tilde{x}.$$

From equations (7.15), (7.16), (7.21), and (7.22), we then get

$$D_i(q_i)\dot{s}_i + C_i(q_i, \dot{q}_i)s_i - J^T_{\phi_i}(q_i)\tilde{f}_i = Y_i^*(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t))\tilde{\varphi}_i^*$$

$$+\sum_{j=1}^{k_i} \hat{U}^1_{ij}(q_i|\tilde{\Theta}^1_{ij})\ddot{q}_{r_{ij}} + \hat{U}^2_i(q_i, \dot{q}_i|\tilde{\Theta}^2_i) - K_{s_i}s_i - J^T_{\phi_i}(q_i)K_i\tilde{x} + \epsilon_i(t).$$

■

Remark 7.4.1 *If the fuzzy logic systems $\hat{U}^1_{ij}(q_i|\Theta^1_{ij})$, and $\hat{U}^2_i(q_i, \dot{q}_i|\Theta^2_i)$, are knowledge rich enough, that is if their input fuzzy labels uniformly and thoroughly span their whole respective input spaces and the number of their fuzzy rules is large enough to describe the systems dynamics, then from the universal approximation theorem, the minimum approximation errors w^1_{ij} and w^2_i, can be approximated to zero with any degree of accuracy [127, 128]. Since w^1_{ij} and w^2_i can be smaller than any machine precision, they can be considered as practically equal to zero.*

$$w^1_{ij} = 0, \ \text{and} \ \ w^2_i = 0,$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, k_i$. Hence, neglecting $\bar{\varepsilon}_i(t)$ (assumption 7.2.1), equation (7.24) yields

$$\epsilon_i(t) = 0. \tag{7.25}$$

Lemma 7.4.2 *If s_i, $\tilde{x}$, $\tilde{\varphi}_i^*$, $\tilde{\Theta}_{ij}^1$, and $\tilde{\Theta}_i^2$, are all bounded for $i = 1, \ldots, m$ and $j = 1, \ldots, k_i$, and the desired trajectory x_d is uniformly continuous, then s_i and $\tilde{x}$ are uniformly continuous.*

Proof: Premultiplying the error dynamics (7.23) by $J_{\phi_i}(q_i)D_i^{-1}(q_i)$, yields

$$\begin{aligned} J_{\phi_i}(q_i)\dot{s}_i - J_{\phi_i}(q_i)D_i^{-1}(q_i)J_{\phi_i}^T(q_i)\tilde{f}_i = \\ J_{\phi_i}(q_i)D_i^{-1}(q_i)\left[Y_i^*(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t))\tilde{\varphi}_i^* + \sum_{j=1}^{k_i}\hat{U}_{ij}^1(q_i|\tilde{\Theta}_{ij}^1)\ddot{q}_{r_{ij}} + \hat{U}_i^2(q_i, \dot{q}_i|\tilde{\Theta}_i^2)\right] \\ -J_{\phi_i}(q_i)D_i^{-1}(q_i)\left[C_i(q_i, \dot{q}_i) + K_{s_i}\right]s_i - J_{\phi_i}(q_i)D_i^{-1}(q_i)J_{\phi_i}^T(q_i)K_i\tilde{x} \\ +J_{\phi_i}(q_i)D_i^{-1}(q_i)\epsilon_i(t). \end{aligned} \tag{7.26}$$

Differentiating equation (4.18) with respect to time leads to

$$J_{\phi_i}(q_i)\dot{s}_i = \ddot{\tilde{x}} + \gamma_i\dot{\tilde{x}} - \dot{J}_{\phi_i}(q_i)s_i. \tag{7.27}$$

Using equations (7.27) and (7.25), and by letting $\gamma_i = \gamma$, equation (7.26) can be rewritten as

$$D_i'(q_i)\tilde{f}_i = \ddot{\tilde{x}} + \gamma\dot{\tilde{x}} + a_i\tilde{x} + b_i s_i - u_i. \tag{7.28}$$

where

$$D_i'(q_i) = J_{\phi_i}(q_i) D_i^{-1}(q_i) J_{\phi_i}^T(q_i) \tag{7.29}$$

$$a_i = D_i'(q_i) K_i$$

$$b_i = J_{\phi_i}(q_i) D_i^{-1}(q_i) (C_i(q_i, \dot{q}_i) + K_{s_i}) - \dot{J}_{\phi_i}(q_i) \tag{7.30}$$

$$u_i = J_{\phi_i}(q_i) D_i^{-1}(q_i) \left[Y_i^*(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, e_i(t)) \tilde{\varphi}_i^* + \sum_{j=1}^{k_i} \hat{U}_{ij}^1(q_i|\tilde{\Theta}_{ij}^1) \ddot{q}_{r_{ij}} \right.$$
$$\left. + \hat{U}_i^2(q_i, \dot{q}_i|\tilde{\Theta}_i^2) + \epsilon_i(t) \right]. \tag{7.31}$$

By computing $[D_1'(q_1)\tilde{f}_1 - D_i'(q_i)\tilde{f}_i]$ for $i = 2, \ldots, m$, and from equation (3.8), we obtain

$$R(q_1, \ldots, q_m)\tilde{f} = Q(t).$$

where $\tilde{f}^T = [\tilde{f}_1^T, \ldots, \tilde{f}_m^T]$, and $R(q_1, \ldots, q_m)$ and $Q(t)$ are defined as follows

$$R(q_1, \ldots, q_m) = \begin{bmatrix} D_1'(q_1) & -D_2'(q_2) & 0 & \cdots & 0 \\ D_1'(q_1) & 0 & -D_3'(q_3) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ D_1'(q_1) & 0 & 0 & \cdots & -D_m'(q_m) \\ I & I & I & \cdots & I \end{bmatrix}$$

$$Q(t) = \begin{bmatrix} (a_1 - a_2)\tilde{x} + (b_1 s_1 - b_2 s_2) - (u_2 - u_1) \\ (a_1 - a_3)\tilde{x} + (b_1 s_1 - b_3 s_3) - (u_3 - u_1) \\ \vdots \\ (a_1 - a_m)\tilde{x} + (b_1 s_1 - b_m s_m) - (u_m - u_1) \\ 0 \end{bmatrix}$$

From the structure of matrix $R(q_1, \ldots, q_m)$ and since $D_i(q_i)$, $i = 1, \ldots, m$, is positive definite, it can be deduced that $R(q_1, \ldots, q_m)$ has full rank. The matrices $J_{\phi_i}(q_i)$ and $D_i^{-1}(q_i)$ are bounded. Thus, $R(q_1, \ldots, q_m)$ and $R^{-1}(q_1, \ldots, q_m)$ are also bounded. The matrices $C_i(q_i, \dot{q}_i)$ and $\dot{J}_{\phi_i}(q_i)$ are bounded for a bounded joint velocity $\dot{q}_i$ (will be justified later in this proof). Besides, s_i, $\tilde{x}$, $\tilde{\varphi}_i^*$, $\tilde{\Theta}_{ij}^1$, and $\tilde{\Theta}_i^2$, for $j = 1, \ldots, k_i$, and $i = 1, \ldots, m$, are bounded. Hence, $Q(t)$ is bounded. Therefore, $\tilde{f} = R^{-1}(q_1, \ldots, q_m)Q(t)$ is bounded. From equation (7.23), the boundedness of $\dot{s}_i$ becomes straight forward. This proves that s_i is uniformly continuous. For a bounded s_i and $\tilde{x}$, it is directly implied by equation (4.18) that $\dot{\tilde{x}}$ is bounded. Then, $\tilde{x}$ is uniformly continuous. From (3.2), it can be deduced that a bounded $\dot{\tilde{x}}$ implies a bounded $\dot{\tilde{q}}_i$. Given the uniform continuousness of x_d it can be deduced from (3.2) that the desired joint velocity $\dot{q}_{d_i}$ is bounded. Thus, $\dot{q}_i$ is indeed bounded. This also justifies the boundedness of $C_i(q_i, \dot{q}_i)$ and $\dot{J}_{\phi_i}(q_i)$ claimed earlier in the proof.

■

Theorem 7.4.1 *Suppose that the gains γ_i are chosen to have the same value $\gamma > 0$ for all m manipulators. If the matrix gains $K_{s_i} > \eta_i$ and $K_i > 0$, $i = 1, \ldots, m$, then the proposed HIC with control law (7.14) gives rise to an asymptotic convergence of s_i and $\tilde{x}$ to zero.*

Proof: Consider the following Lyapunov function candidate

$$V = \sum_{i=1}^{m} V_i.$$

where

$$V_i = \frac{1}{2}\left\{ s_i^T D_i(q_i) s_i + \tilde{x}^T K_i \tilde{x} + \tilde{\varphi}_i^{*T} \Gamma_i \tilde{\varphi}_i^* + \sum_{j=1}^{k_i} \sum_{k=1}^{k_i} (\tilde{\Theta}_{ij}^1)_k^T \Gamma_{ij}^1 (\tilde{\Theta}_{ij}^1)_k \right.$$
$$\left. + \sum_{k=1}^{k_i} (\tilde{\Theta}_i^2)_k^T \Gamma_i^2 (\tilde{\Theta}_i^2)_k \right\}.$$

with $(\tilde{\Theta}_{ij}^1)_k$ and $(\tilde{\Theta}_i^2)_k$ denoting the kth column of matrices $\tilde{\Theta}_{ij}^1$ and $\tilde{\Theta}_i^2$, respectively. Since $D_i(q_i)$, K_i, Γ_i, Γ_{ij}^1, and Γ_i^2 are positive definite matrices, V_i, and hence V, are non-negative scalars. From the error dynamics (7.23) and equation (3.13),

$$\dot{V}_i = -s_i^T (K_{s_i} - \frac{1}{2}\dot{c}_i(t) Q_0(q_i, \dot{q}_i)) s_i - s_i^T J_{\phi_i}^T (q_i) K_i \tilde{x}$$
$$+ s_i^T J_{\phi_i}^T (q_i) \tilde{f}_i + \tilde{x}^T K_i \dot{\tilde{x}} + \dot{\tilde{\varphi}}_i^{*T} \Gamma_i \tilde{\varphi}_i^* + \sum_{j=1}^{k_i} \sum_{k=1}^{k_i} (\dot{\tilde{\Theta}}_{ij}^1)_k^T \Gamma_{ij}^1 (\tilde{\Theta}_{ij}^1)_k + \sum_{k=1}^{k_i} (\dot{\tilde{\Theta}}_i^2)_k^T \Gamma_i^2 (\tilde{\Theta}_i^2)_k$$
$$+ s_i^T \left[Y_i^*(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t)) \tilde{\varphi}_i^* + \sum_{j=1}^{k_i} \tilde{U}_{ij}^1 (q_i | \tilde{\Theta}_{ij}^1) \ddot{q}_{r_{ij}} + \tilde{U}_i^2 (q_i, \dot{q}_i | \tilde{\Theta}_i^2) + \epsilon_i(t) \right].$$

Using equations (7.21), and (7.22) and substituting equation (4.18), leads to

$$\dot{V}_i = -s_i^T (K_{s_i} - \frac{1}{2}\dot{c}_i(t) Q_0(q_i, \dot{q}_i)) s_i - \gamma_i \tilde{x}^T K_i \tilde{x} + (\dot{\tilde{x}} + \gamma_i \tilde{x})^T \tilde{f}_i$$
$$+ \dot{\tilde{\varphi}}_i^{*T} \Gamma_i \tilde{\varphi}_i^* + \sum_{j=1}^{k_i} \sum_{k=1}^{k_i} (\dot{\tilde{\Theta}}_{ij}^1)_k^T \Gamma_{ij}^1 (\tilde{\Theta}_{ij}^1)_k + \sum_{k=1}^{k_i} (\dot{\tilde{\Theta}}_i^2)_k^T \Gamma_i^2 (\tilde{\Theta}_i^2)_k$$
$$+ s_i^T \left[Y_i^*(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t)) \tilde{\varphi}_i^* + \sum_{j=1}^{k_i} (\tilde{\Theta}_{ij}^1)^T \xi_i^1 (q_i) \ddot{q}_{r_{ij}} + (\tilde{\Theta}_i^2)^T \xi_i^2 (q_i, \dot{q}_i) + \epsilon_i(t) \right]$$
$$= \dot{V}_{i1} + \dot{V}_{i2} + \dot{V}_{i3} + \dot{V}_{i4} + \dot{V}_{i5} + \dot{V}_{i6}.$$

where

$$\dot{V}_{i1} = -s_i^T(K_{s_i} - \frac{1}{2}\dot{c}_i(t)Q_0(q_i, \dot{q}_i))s_i - \gamma_i\tilde{x}^T K_i\tilde{x}$$

$$\dot{V}_{i2} = (\dot{\tilde{x}} + \gamma_i\tilde{x})^T\tilde{f}_i$$

$$\dot{V}_{i3} = \dot{\tilde{\varphi}}_i^{*T}\Gamma_i\tilde{\varphi}_i^* + s_i^T Y_i^*(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t))\tilde{\varphi}_i^* \tag{7.32}$$

$$\dot{V}_{i4} = \sum_{j=1}^{k_i}\sum_{k=1}^{k_i}(\dot{\tilde{\Theta}}_{ij}^1)_k^T\Gamma_{ij}^1(\tilde{\Theta}_{ij}^1)_k + s_i^T\sum_{j=1}^{k_i}(\tilde{\Theta}_{ij}^1)^T\xi_i^1(q_i)\ddot{q}_{r_{ij}} \tag{7.33}$$

$$\dot{V}_{i5} = \sum_{k=1}^{k_i}(\dot{\tilde{\Theta}}_i^2)_k^T\Gamma_i^2(\tilde{\Theta}_i^2)_k + s_i^T(\tilde{\Theta}_i^2)^T\xi_i^2(q_i, \dot{q}_i) \tag{7.34}$$

$$\dot{V}_{i6} = s_i^T\epsilon_i(t).$$

Setting γ_i to a unique value γ for all m manipulators, and using equation (3.14), yields

$$\dot{V}_{i1} \leq -s_i^T(K_{s_i} - \eta_i I_{k_i})s_i - \gamma\tilde{x}^T K_i\tilde{x}.$$

If the gain matrices K_{s_i} and K_i are chosen in such a way as to satisfy the following constraints

$$K_{s_i} > \eta_i I_{k_i}, \quad K_i > 0,$$

then

$$\dot{V}_{i1} \leq 0.$$

From the property of internal forces, we have $\sum_{i=1}^m f_i = 0$ and $\sum_{i=1}^m f_{d_i} = 0$. Thus, with γ_i set to a unique value γ for all m manipulators,

$$\sum_{i=1}^m \dot{V}_{i2} = 0.$$

Substituting equation (7.7) into (7.32) leads to

$$\dot{V}_{i3} = 0.$$

Similarly, substituting equations (7.11) and (7.12) into (7.33) and (7.34), respectively, results in

$$\dot{V}_{i4} = 0 \ . \quad \text{and} \ \dot{V}_{i5} = 0.$$

Equation (7.25) directly implies that

$$\dot{V}_{i6} = 0.$$

Therefore,

$$\dot{V}_i = \sum_{j=1}^{6} \dot{V}_{ij} \leq 0.$$

Thus, V_i is non-increasing, and hence s_i, $\tilde{x}$, $\tilde{\varphi}_i^*$, $\tilde{\Theta}_{ij}^1$, and $\tilde{\Theta}_i^2$, for $i = 1, \ldots, m$ and $j = 1, \ldots, k_i$, are bounded, since they appear in V_i. If the desired trajectory x_d is uniformly continuous, then lemma 7.4.2, implies that s_i and $\tilde{x}$ are uniformly continuous. Remember that $(\dot{c}_i(t)Q_0(q_i, \dot{q}_i))$ is uniformly continuous (assumption 3.4.1). Thus, from Barbalat's lemma,

$$\lim_{t\to\infty} \dot{V}_{i1} = 0.$$

Therefore,

$$\lim_{t\to\infty} s_i = 0. \quad \text{and} \quad \lim_{t\to\infty} \tilde{x} = 0.$$

∎

Remark 7.4.2 *It is possible to reduce the effect of the error term $\epsilon_i(t)$ and enhance the robustness of the HIC by adding an additional term to equation (7.13). The robust adaptive fuzzy control law would have the following form*

$$\tau_i^{(f)} = \sum_{j=1}^{k_i} \hat{U}_{ij}^1(q_i|\Theta_{ij}^1)\ddot{q}_{r_{ij}} + \hat{U}_i^2(q_i, \dot{q}_i|\Theta_i^2) - W_i\, sgn(s_i). \tag{7.35}$$

where

$$W_i = diag(\bar{w}_{i1}, \ldots, \bar{w}_{ik_i}),$$
$$\bar{w}_i \geq \sup_t \Big(\sum_{j=1}^{k_i} |w_{ij}^1 \ddot{q}_{r_{ij}}| + |w_i^2|\Big),$$

and $\bar{w}_{ik}$ is the kth element of vector $\bar{w}_i$, $i = 1, \ldots, m$. This will ensure that $\dot{V}_i$ is negative even when $\epsilon_i(t) \neq 0$. It is important to point out that, inspite of the robustness of the AFC with control law (7.35), it is not preferred from a practical point of view since it often leads to a chattering behavior of the manipulators. This makes the control of closed-chain robotic systems a very difficult task to achieve as it complicates the coordination between the manipulators.

Theorem 7.4.2 *The adaptive fuzzy controller (7.14) also leads to an asymptotic convergence of the force error $\tilde{f}_i$ to zero.*

Proof: Premultiplying equation (7.28) by $(D_i'(q_i))^{-1}$ results in

$$\tilde{f}_i = (D_i'(q_i))^{-1}(\ddot{\tilde{x}} + \gamma\dot{\tilde{x}}) + K_i\tilde{x} + (D_i'(q_i))^{-1}b_i s_i - (D_i'(q_i))^{-1}u_i. \tag{7.36}$$

where $D_i'(q_i)$, b_i, and u_i, are as defined in equations (7.29), (7.30), and (7.31), respectively. It is easy to see from equation (4.14) that matrix $D_i'(q_i)$ is invertible. Since, $\tilde{x}$ converges to zero, $\dot{\tilde{x}}$ and $\ddot{\tilde{x}}$ also converge to zero. Keeping in mind that s_i

converges to zero and from equations (3.8), (7.31), and (7.25),

$$\lim_{t\to\infty} \sum_{i=1}^{m} (D_i'(q_i))^{-1} J_{\phi_i}(q_i) D_i^{-1}(q_i) \Big[Y_i^*(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t)) \tilde{\varphi}_i^* + \sum_{j=1}^{k_i} \hat{U}_{ij}^1(q_i|\tilde{\Theta}_{ij}^1) \ddot{q}_{r_{ij}} + \hat{U}_i^2(q_i, \dot{q}_i|\tilde{\Theta}_i^2) \Big] = 0. \quad (7.37)$$

The joint velocity q_i, and hence the term $(D_i'(q_i))^{-1} J_{\phi_i}(q_i) D_i^{-1}(q_i)$, can be continuously changing in time and may assume an infinite number of possible values for $i = 1, \ldots, m$. Since equation (7.37) has to hold for all those values, and since the term $(D_i'(q_i))^{-1} J_{\phi_i}(q_i) D_i^{-1}(q_i)$ is bounded, the only solution to equation (7.37) is to have

$$\lim_{t\to\infty} \left[Y_i^*(q_i, \dot{q}_i, \dot{q}_{r_i}, \ddot{q}_{r_i}, c_i(t)) \tilde{\varphi}_i^* + \sum_{j=1}^{k_i} \hat{U}_{ij}^1(q_i|\tilde{\Theta}_{ij}^1) \ddot{q}_{r_{ij}} + \hat{U}_i^2(q_i, \dot{q}_i|\tilde{\Theta}_i^2) \right] = 0.$$

Then, from equation (7.36),

$$\lim_{t\to\infty} \tilde{f}_i = 0.$$

■

The enhanced position and internal force tracking capabilities of the proposed HIC in the face of parametric and unstructured uncertainties are illustrated in the following section.

7.5 Results and Discussion

In order to assess the HIC and to confirm the theoretical properties pertaining to its stability and tracking abilities under significant structured and unstructured uncertainties, the same three experiments of Chapter 6 are reproduced here. The initial estimated value of the payload's mass, $\hat{\varphi}_i^*$, $i = 1,2$, is chosen to be zero. This may be considered as an extreme case as in most practical situations at least a range of the possible values of such a parameter is known in advance. The point behind this choice is to better illustrate the high tracking capability of the HIC even in extreme cases. As for the AFC module, it is assumed to have no apriori knowledge of the system's dynamics and its membership functions are chosen to be the same as those used in Chapter 6. The initial values of $\hat{\tau}_{d_i}$, Θ_1^{ij}, and Θ_2^i, are set to zero, for $i = 1,2$ and $j = 1,\ldots,k_i$. The matrices $D_i^*(q_i)$, $C_i^*(q_i,\dot{q}_i)$, and $G_i^*(q_i)$, are chosen to be 50% of their respective nominal values. In other words,

$$D_i^*(q_i) = 0.5\, D_i(q_i),$$

$$C_i^*(q_i,\dot{q}_i) = 0.5\, C_i(q_i,\dot{q}_i),$$

$$G_i^*(q_i) = 0.5\, G_i(q_i).$$

It is important to point out that this percentage value is not much significant as the AFC module in the HIC always compensates for the dynamics modeling error regardless of its magnitude. The CAC module is set to operate at a bandwidth four times higher than that of the AFC module.

In the first experiment, only parametric uncertainties are assumed to exist in the robots dynamics. The HIC's tracking performance is shown in figure 7.4. The payload's position and the internal forces converge to their desired values after a certain parameters tuning phase. It is important to notice how the tracking performance of the HIC is close to that of the AFC developed in the previous chapter. In order to compare the tracking ability of the HIC with its conventional counterpart, the pure CAC (of Chapter 4), the tracking errors of the two controllers are superimposed as shown in figure 7.5. Although all the tracking errors decay to zero with both controllers, the HIC shows a slower convergence rate than the CAC in this case. As in the case of the pure AFC in Chapter 6, this is expected since the AFC module of the HIC starts the learning of the robots overall dynamics without any initial prior knowledge of it (i.e., Θ_1^{ij}, and Θ_2^{i}, are initially set to zero, for $i = 1, 2$ and $j = 1, \ldots, k_i$.) To compare the amount of effort generated by the HIC, the pure CAC (of Chapter 4), and the pure AFC (of Chapter 6), the same type of control effort performance index as the one used in the previous chapter is also used here. The effort performance indices of the three types of controllers are provided in table 7.1. As it can be seen, the HIC consumes less overall effort than the two other controllers. The computed torques in the two robots are illustrated in figure 7.6.

Table 7.1: Effort Indices of the pure CAC, pure AFC, and the HIC, in the Presence of Parametric Uncertainties Only ($\alpha = 0$.)

	$\beta_1^{(1)}$	$\beta_1^{(2)}$	$\beta_1^{(3)}$	β_1	$\beta_2^{(1)}$	$\beta_2^{(2)}$	$\beta_2^{(3)}$	β_2	β
Pure CAC	360.20	272.20	8.87	641.35	414.68	404.86	36.93	856.47	1,497.82
Pure AFC	357.01	247.80	9.66	614.48	415.20	318.38	18.29	751.87	1,366.35
HIC	309.84	248.07	9.36	567.27	409.95	284.79	21.28	716.02	1,283.29

The second experiment is meant to study the tracking behavior of the HIC under parametric and modeling uncertainties. The HIC's tracking performance is provided in figure 7.7. As it is expected, and unlike the pure CAC, the HIC shows a satisfactory performance even when, both, structured and unstructured uncertainties coexist in the system's dynamics. Figure 7.8 provides a comparison of the tracking errors of the HIC and the pure CAC. In this case, it is clear how the HIC outperforms its conventional counterpart, the pure CAC. Like in the pure AFC, this figure also shows that the tracking errors of the HIC do not converge exactly to zero but they converge rather to a certain envelope which is in the vicinity of zero. Again, we believe that this is mainly due to the fact that the effect of the external disturbance factor $\rho(t)$ becomes no longer negligible. The time-varying term $\rho(t)$ depends on none of the HIC's inputs, and hence increasing α violates a close to zero value of $\epsilon_i(t)$, which in turn invalidates equation (7.25). Figure 7.9 shows the HIC's computed torques for the two robots.

The third experiment is meant to compare the tracking performance of the HIC to that of the pure CAC under parametric uncertainties and varying degrees of modeling uncertainties. This is achieved by letting α span a spectrum of values in the interval $[0, 2.5]$. The traking errors of both controllers are plotted in figure 7.10. Again, the tracking superiority of the HIC over that of the pure CAC is very clear. The pure CAC shows very little tolerance to the increasing effect of the modeling uncertainty caused by the increase of α. On the other hand, the HIC proves to be more robust and it shows a higher flexibility in tolerating, both, structured and unstructured uncertainties. The HIC's tracking errors do not converge exactly to zero for relatively high values of α. This is mainly due to the higher impact of $\rho(t)$ on the system's dynamics for high values of α, as explained earlier. However, even in the presence of significant modeling uncertainties, the HIC's tracking ability is still satisfactory as its tracking errors still converge to within the vicinity of zero. At the end of this section, it is worth mentioning that in terms of CPU time requirements, the HIC provided an elevenfold improvement as compared to the AFC scheme proposed in Chapter 6. This marks HIC closer to CAC in terms of CPU time requirements while providing much better tracking and stability performances in the face of considerable structured and unstructured uncertainties.

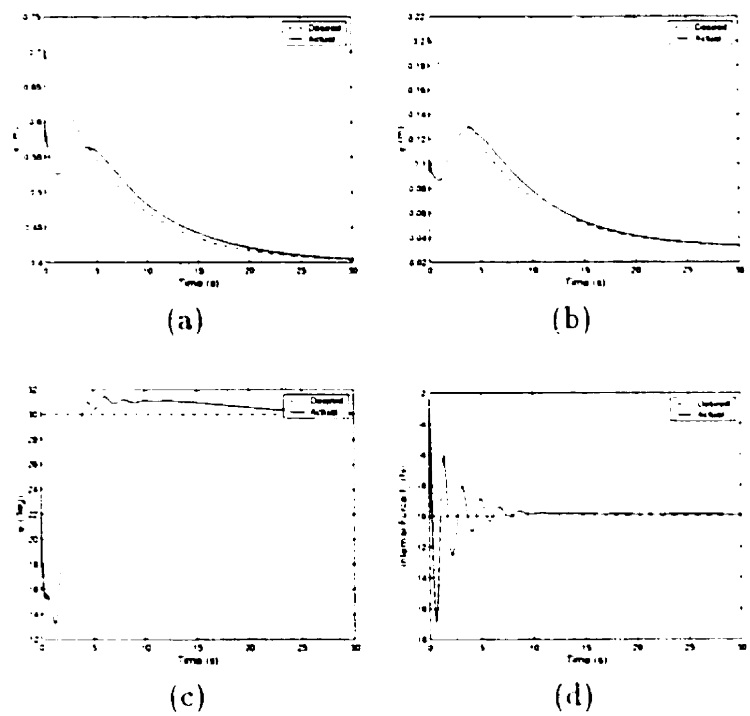


Figure 7.4: Tracking Performance of the HIC in the Presence of Parametric Uncertainties Only ($\alpha = 0$.)

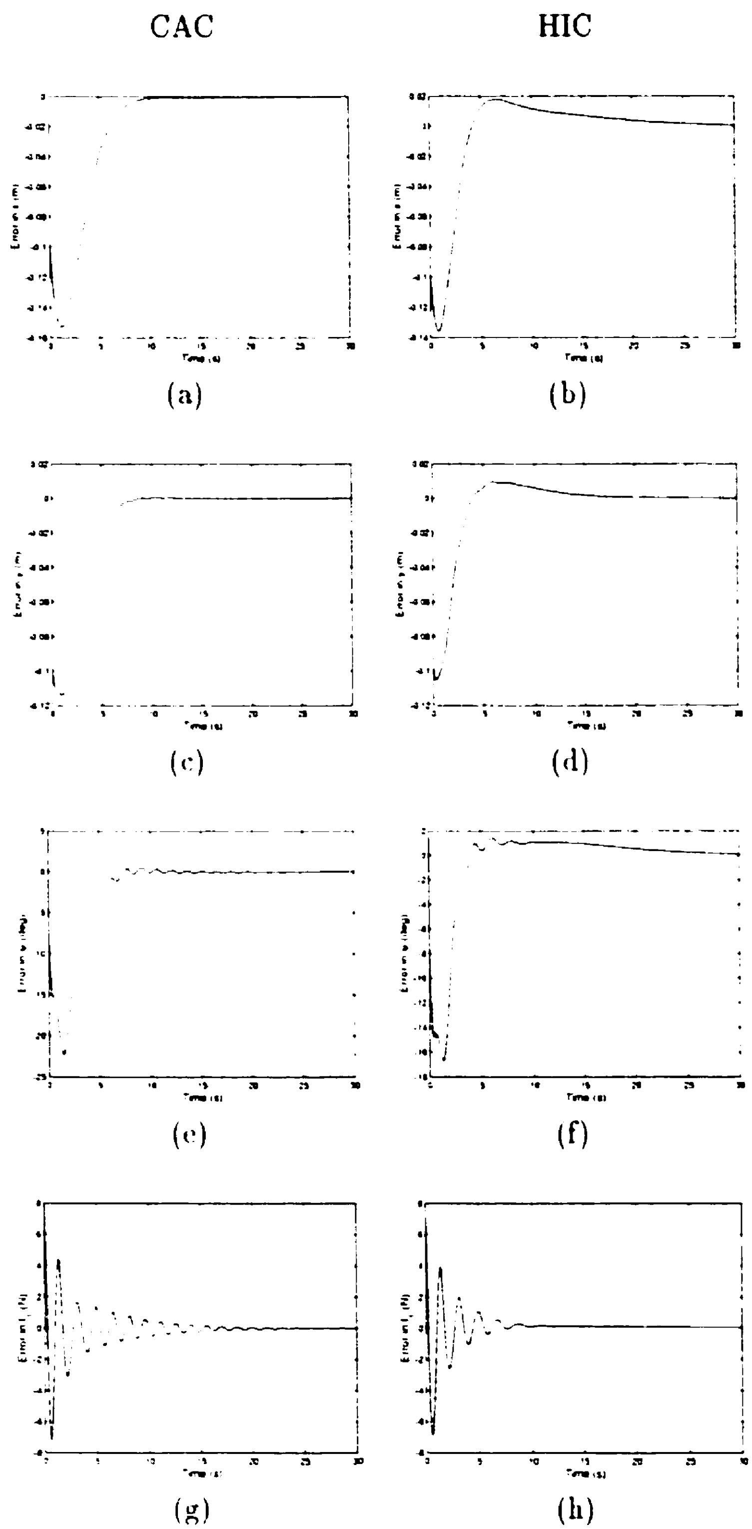


Figure 7.5: Tracking Errors of the pure CAC and the HIC in the Presence of Parametric Uncertainties Only ($\alpha = 0.$)

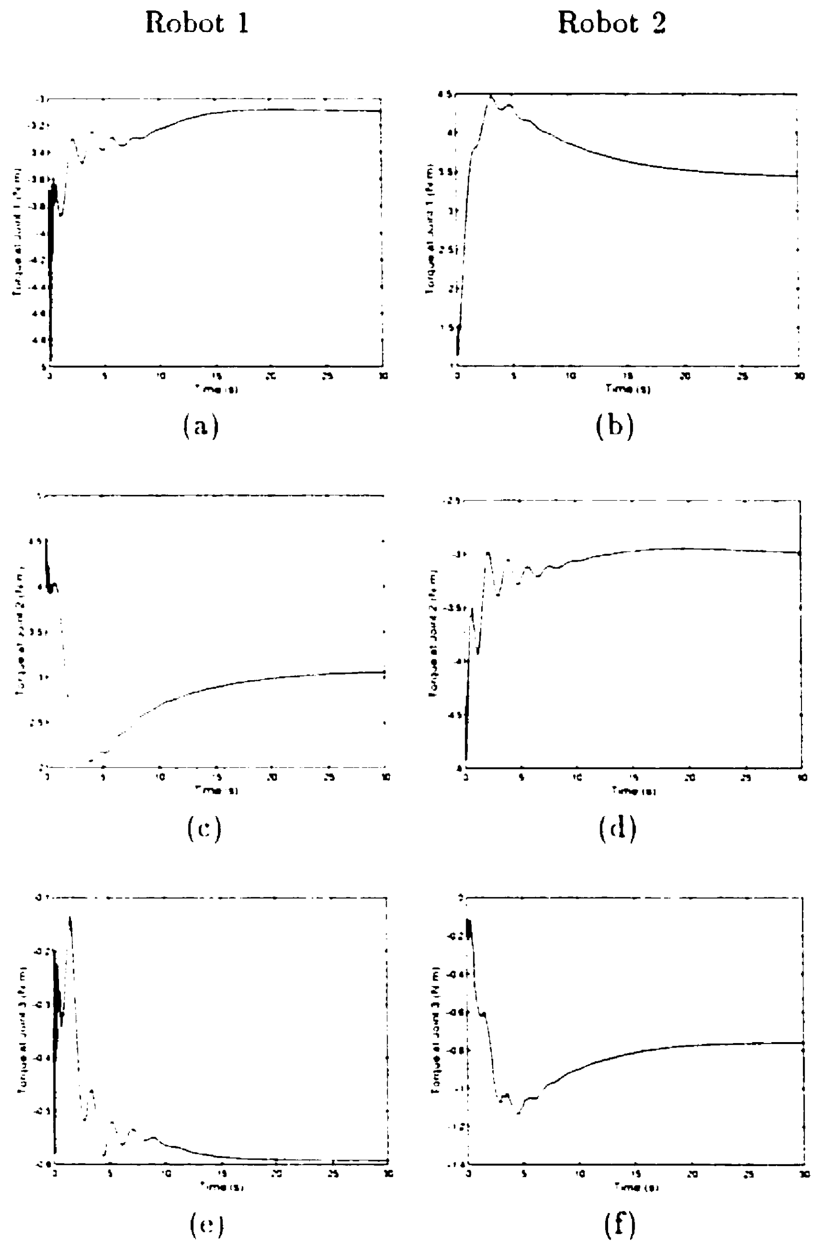


Figure 7.6: Computed Torques Using the HIC in the Presence of Parametric Uncertainties Only ($\alpha = 0$.)

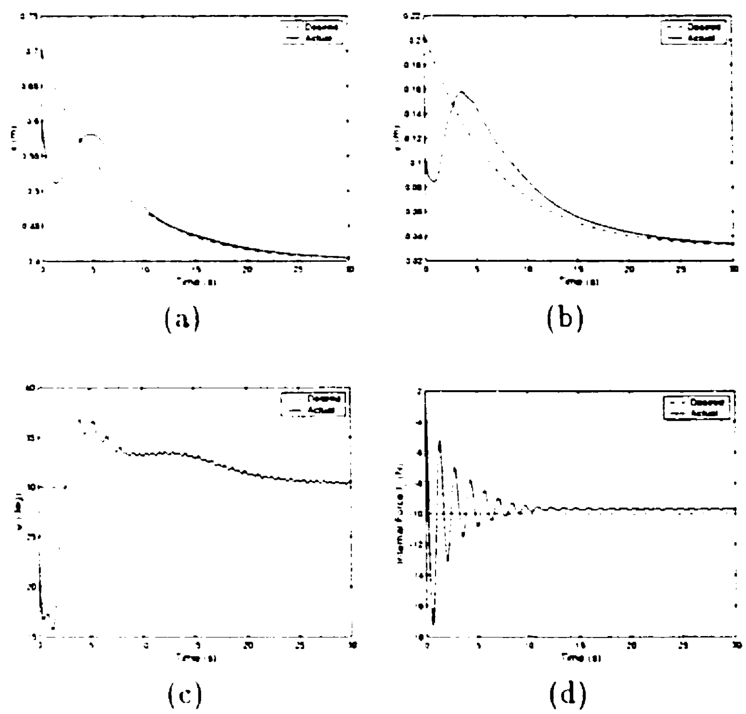


Figure 7.7: Tracking Performance of the HIC in the Presence of Parametric and Modeling Uncertainties ($\alpha = 1.$)

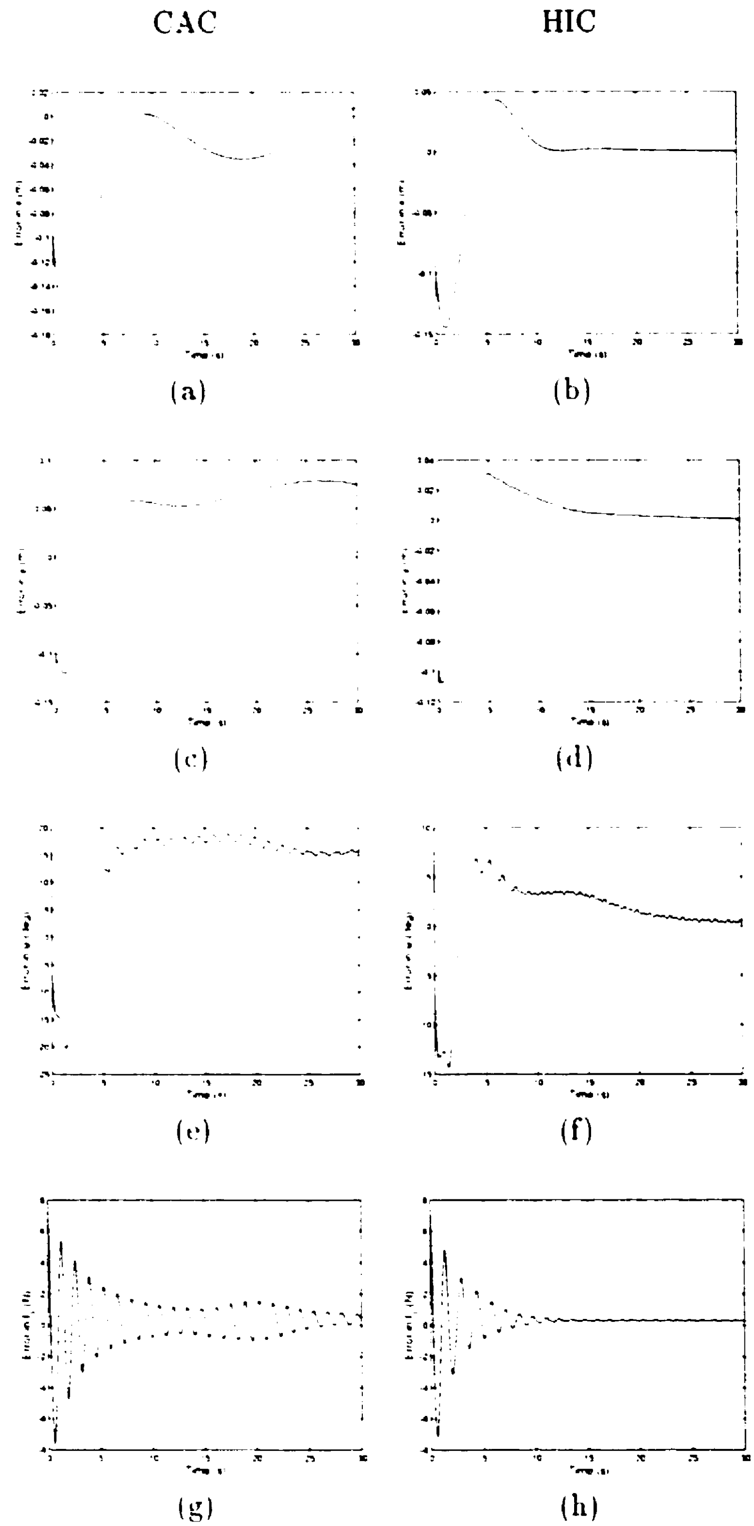


Figure 7.8: Tracking Errors of the pure CAC and the HIC in the Presence of Parametric and Modeling Uncertainties ($\alpha = 1.$)

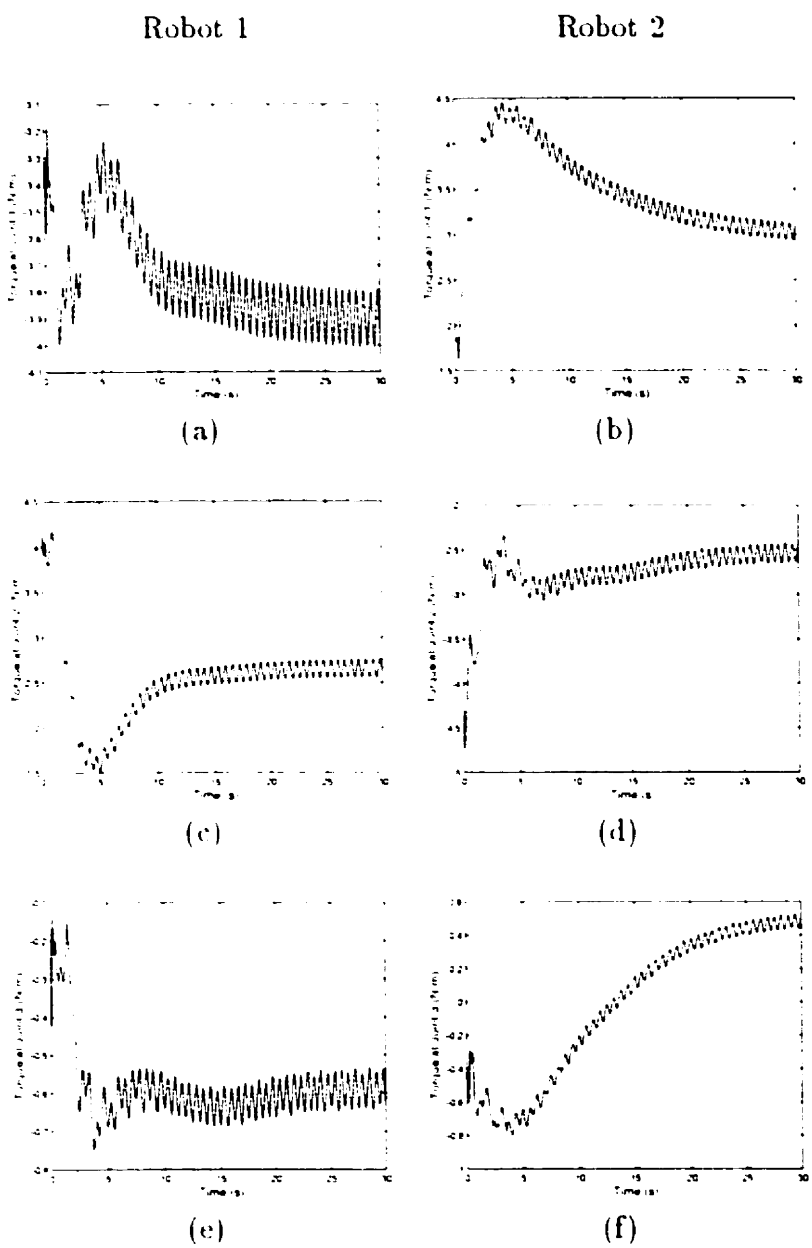


Figure 7.9: Computed Torques Using the HIC in the Presence of Parametric and Modeling Uncertainties ($\alpha = 1.$)

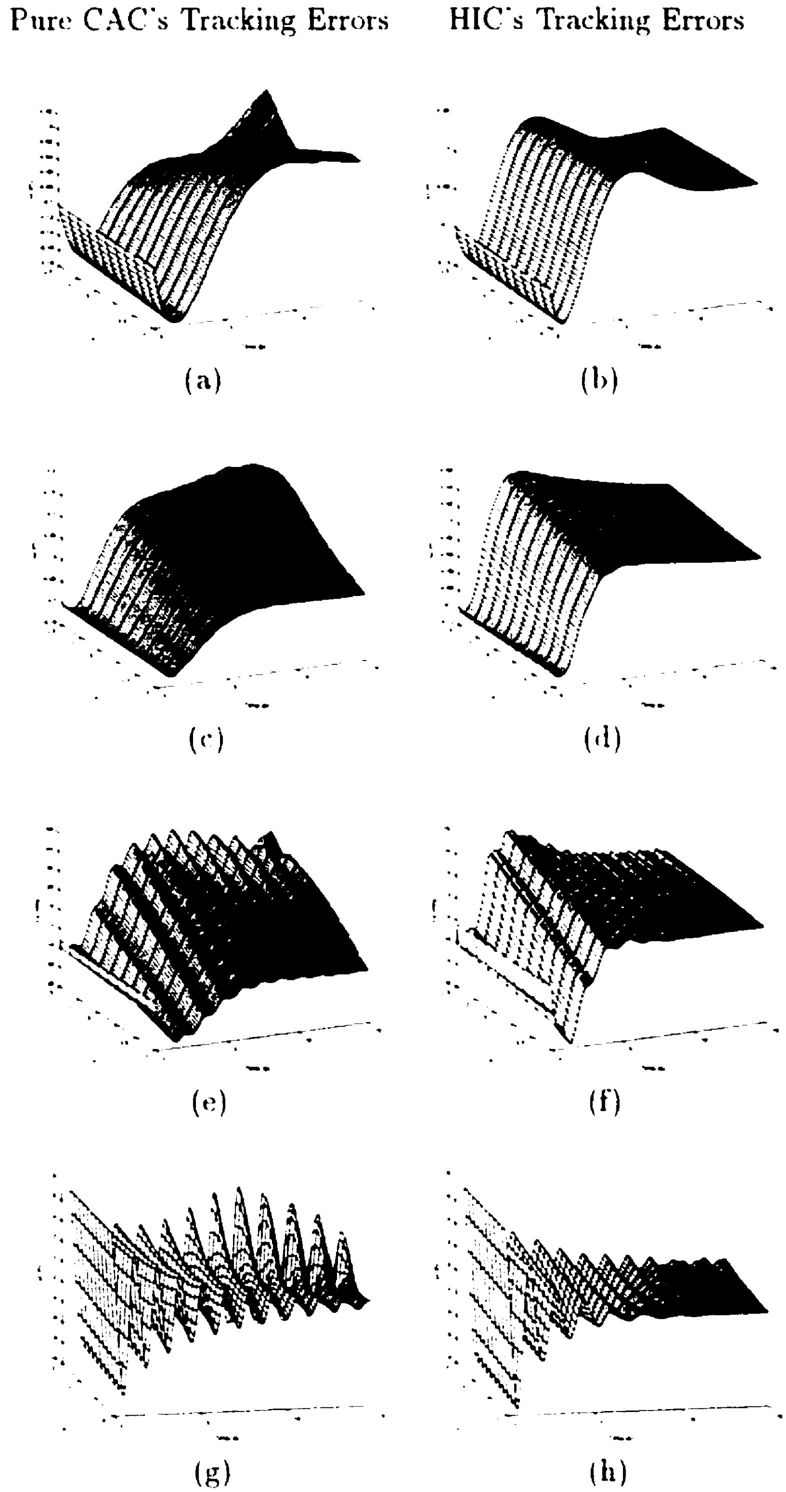


Figure 7.10: The Sensitivity of the pure CAC Versus that of the HIC under Parametric Uncertainties and Varying Intensities of Unstructured Uncertainties.

7.6 Conclusion

In this chapter, a decentralized adaptive HIC is proposed for the control of multiple strongly coupled manipulators handling a common object. The controller makes use of an innovative hierarchical structure that consists of two adaptive control modules: a conventional one and a MIMO AFC which makes use of an online adaptation scheme to fully approximate the overall system's dynamics starting from partial or no apriori knowledge of it. An innovative rule reduction technique is suggested to drastically decrease the computational complexity of the AFC module. Besides, making the AFC module operate at a lower bandwidth than the conventional adaptive control module of the HIC, reduces the computational burden even further. The proposed controller has been shown to be robust in the face of a substantial amount of parametric and modeling uncertainties and external disturbances with varying intensity levels. It has been proven that even under the existence of these uncertainties and external disturbances in the system's dynamics, the position and the internal forces tracking errors still converge to zero. The present controller not only provided powerful tracking capabilities but has been also shown to be less expensive in terms of control efforts and computational requirements. A hardware implementation of this controller could not only benefit from this type of robotic structure but also other type of structures of similar complexities.

Chapter 8

Conclusions

In this chapter we provide the reader with a brief summary of this research work along with comments we were able to draw from it. We conclude the chapter, by providing some suggestions on possible extensions and further improvements of this work.

8.1 Thesis Summary and Concluding Remarks

In this thesis, we tackled the very complex control problem of closed kinematic chain mechanisms formed by two or more robotic manipulators grasping a common object and simultaneously carrying it along a predefined trajectory. This is a much more complex problem than that of controlling a single robotic arm. The kinematic constraints present in cooperative manipulator systems contribute to the synchronized motion of the end-effectors involved as to track the predefined desired position and orientation of the payload without losing contact of it. Due to the strong coupling of the manipulators in the closed kinematic chain mechanisms and as a result of the direct contact of the end-effectors and the manipulated object, internal forces are generated in the mechanical system. Consequently, the robotic manipulators have to be dynamically coordinated as to control these internal forces to avoid any possible physical damage to the load or to the manipulators' wrists/end-effectors.

Most of the research work that has tackled this problem so far using conventional non-adaptive control techniques assumes a full knowledge of the system's dynamics [75, 76, 142, 143]. This is usually an unrealistic assumption as most complex systems, such as strongly coupled cooperative manipulators, are subject to different types of varying uncertainties. As it is shown in Chapter 4, a conventional non-adaptive control scheme, such as the feedback linearization, is severely sensitive to even minor dynamics imprecisions. Besides, it does not allow simultaneous tracking of the payload's position/orientation and that of the internal forces. To overcome this problem, several conventional adaptive control strategies have been proposed [47, 70, 84, 104, 105, 125, 126, 139]. Although most of them were able to achieve a satisfactory position/orientation tracking performance in the face of parametric (structured) uncertainties, their internal force tracking abilities were still very poor [6, 125, 135]. Recently, a widely cited conventional adaptive controller has been proposed to simultaneously control the position/orientation of the payload and the internal forces acting on it [70]. Although this controller is considered as a breakthrough in the area of cooperative manipulators control, it has been shown (in Chapter 4) that it can only achieve its control objectives when the system's dynamics is free from any modeling (unstructured) uncertainties, such as unmodeled external disturbances for instance. In other words, the controller is robust to only parametric uncertainties.

Modeling imperfection of complex systems, such as closed-chain robotic manipulators, is inevitable in many cases. Not only are conventional adaptive controllers intolerant to unstructured uncertainties, they also require an explicit linear parameterization of the modeled uncertain dynamics parameters to exist. Although this condition is guaranteed for every robotic dynamic system, the derivation of such a linear parameterization may be quite complex and cumbersome, especially with complex ill-defined systems. It is important to mention that the effect of the unstructured uncertainties on the controller's tracking performance and stability is ignored in most of the work on conventional adaptive control of robotic systems. In addition, it is to the best of our knowledge that this is still an open area of research in which no significant practical results have been achieved yet as far as strongly coupled cooperative robots are concerned. This makes the development of a robust control approach for the increasingly complex cooperative manipulator systems a necessary step to keep up with the increasingly demanding design requirements of such systems.

Our main approach to tackle these difficult, yet important, problems is based on using adaptive soft computing based controllers. This choice, which is still unique of its kind, is motivated by the excellent approximation capabilities of fuzzy logic systems and their tolerance to, both, structured and unstructured uncertainties. Moreover, providing them with a systematic online learning mechanism gives them an additional strength to automatically adapt themselves to time-varying operating conditions making them even more suitable for the control of ill-defined systems.

In Chapter 6, a direct model-reference adaptive fuzzy controller (AFC) is developed for this purpose. Starting from no prior knowledge of the system's dynamics, the AFC automatically adapts itself online to better approximate the system's overall dynamics and generate the appropriate torques as to perfectly track the payload's desired position/orientation trajectory and the internal force desired values, simultaneously. The designed AFC is shown to achieve these control goals even in the presence of significant parametric and modeling uncertainties. Its convergence and stability properties are also studied through a Lyapunov stability approach. However, inspite of the robustness of this AFC and its high tracking ability, it still suffers from a few shortcomings:

- Although a rule-reduction technique is used to decrease the total number of rules needed for the AFC, it might still cause a substantial computational burden to the system as the number of rules required might still be too large to be fired simultaneously within the closed loop control system.

- The AFC lacks the ability of incorporating the already acquired knowledge of the system's dynamics as it initially assumes no prior knowledge of it.

To overcome these drawbacks,a hybrid intelligent controller (HIC) is proposed in Chapter 7. This knowledge based controller takes advantage of an innovative hierarchical framework, which we designed exclusively for this purpose, to allow a conventional adaptive controller and an adaptive fuzzy one to operate simultaneously, but at different bandwidths, to achieve the common overall control objective while incorporating any (partial) information already known about the system's dynamics. Moreover, the HIC also makes use of another innovative rule-reduction scheme, introduced in this work for the first time, to significantly enhance the computational requirement of the controller. As a matter of fact, the HIC uses only 34.13% of the number of fuzzy rules used by the pure AFC developed in Chapter 6 and an elevenfold improvement in the CPU time consumption was observed. The HIC is also shown to require less control effort than the pure AFC and the pure CAC. In addition to the high control effort and computational efficiencies, the HIC is proven to maintain a similar high tracking ability and stability properties to those of the pure AFC. All these assets make the HIC ideal for a hardware implementation. It is worth mentioning that although the application of the AFC and the HIC presented in chapters 6 and 7 has been restricted to cooperative manipulators in this thesis, it can be easily extended to cover a wide range of electro-mechanical systems as well. It is important to point out that, to the best our knowledge, this research work is the first of its kind in the area of advanced control of cooperative manipulator systems. Yet, the results obtained have been very encouraging and would certainly open new opportunities for tackling robot control of complex structures in general. This being said, we still believe some improvements could be done to make the approaches being developed easily implementable in real world environments.

8.2 Future Research Directions

As with most research efforts, this dissertation has possibly raised as many questions as it has solved. We believe that the different ideas and techniques presented here should provide a strong foundation upon which researchers can build upon and extend. In the following, we present some potential future research directions to our current work.

We strongly believe that the number of fuzzy rules used by the HIC can be still reduced even further. A possibly large number of fuzzy rules have small significant firing strengths. An enhanced rule reduction technique might use this fact to fire only those rules the firing strengths of which are higher than a certain predefined threshold. It is worth mentioning that a similar technique is widely used in the area of numerical linear algebra to annihilate some of the nonzero elements caused by fill-ins when computing the inverse of large sparse matrices.

In this work, the membership functions used to fuzzify the inputs of the fuzzy logic systems are static. This is means that their parameters are not adjusted. A possible improvement would be to derive an adaptation law to automatically tune the parameters of these membership functions online as we did for the consequent parameters of the fuzzy logic systems. This extension might not be trivial though.

Moreover and along the same direction, the fuzzy logic systems used in Chapters 6 and 7 have the structure of a zero-order Sugeno system. As it is generally known, first-order Sugeno systems have higher function approximation capabilities than those of the zero-order. As a matter of fact, first-order Sugeno systems have been successfully applied in a large number of control applications, especially after the development of ANFIS [49]. As a future work, one can try to derive an adaptation mechanism to automatically adjust the consequent parameters of a first-order adaptive Sugeno system online, and then apply it to the control of single or multiple manipulator systems.

Another possible future research direction is to apply a type-2 fuzzy logic system to the control of cooperative robotic manipulators rather than the ordinary type of fuzzy logic systems. Type-2 fuzzy logic systems are generally credited to be more insensitive to parametric and modeling imprecisions than the ordinary fuzzy logic systems. Inspite of the rapidly growing interest in the applications of such systems [13, 67], their theoretical foundations are still considered in their infancy state. Nevertheless, we strongly believe that the research on the applications of these new powerful tools of computational intelligence to single or cooperative robotic manipulator systems might lead to novel results in this important field.

Appendix A

Fuzzy Logic Controllers

A.1 Fuzzy Sets

Fuzzy sets are the core of fuzzy logic based controllers. By definition, a fuzzy set is a set containing elements that have varying degrees of membership, unlike classical (or crisp) sets where members of a crisp set would not be members unless their membership was full in that set (i.e., their membership is assigned a value of 1). Elements of a fuzzy set are mapped to a universe of membership values using a function-theoretic form. The function maps elements of a fuzzy set into a real value belonging to the interval between 0 and 1.

Definition A.1.1 A *fuzzy set* is a set of ordered crisp pairs $A = \{(x, \mu_A(x)) : x \in X\}$, where X is a set of crisp values and is known as the *universe of discourse*, and $\mu_A(x)$ is a mapping from X to the unit interval $[0, 1]$ and is referred to as the *membership function*.

In the thesis, we denote the membership function of any fuzzy set S as $\mu_S(\cdot)$.

A.2 Fuzzy Set Operations

To make fuzzy sets useful, fuzzy set operations and their corresponding operators have been defined. Like in crisp sets, the most used fuzzy set operations are the union, the intersection and the complement. These operations are denoted by $\cup$, $\cap$ and $^-$ (bar), respectively. The union and the intersection operators are usually referred to as *t-conorms* and *t-norms*, respectively.

In the following, we will consider that A, B, C, and D, to be four fuzzy sets, and x is an element of their corresponding universe of discourse X.

Union The union of two fuzzy sets is a fuzzy set whose membership function depends on the t-conorm (also referred to as the *s-norm*) used. A t-conorm operating on the two membership functions $\mu_A(\cdot)$ and $\mu_B(\cdot)$, is simply a two-parameter function, s, of the form

$$\begin{aligned} s: \quad [0,1] \times [0,1] \quad &\longrightarrow \quad [0,1] \\ x \quad &\longmapsto \quad \mu_{A\cup B}(x) = s(\mu_A(x), \mu_B(x)) \end{aligned}$$

Any t-conorm function has to satisfy the following conditions [68]:

i. Boundary Conditions: $s(1,1) = 1$, and $s(\mu_A(x), 0) = s(0, \mu_A(x)) = \mu_A(x)$.

ii. Commutativity: $s(\mu_A(x), \mu_B(x)) = s(\mu_B(x), \mu_A(x))$.

iii. Motonocity: if $\mu_A(x) \leq \mu_C(x)$ and $\mu_B(x) \leq \mu_D(x)$

then $s(\mu_A(x), \mu_B(x)) \leq s(\mu_C(x), \mu_D(x))$.

iv. Associativity: $s(\mu_A(x), s(\mu_B(x), \mu_C(x))) = s(s(\mu_A(x), \mu_B(x)), \mu_C(x))$.

Inspite of the wide variety of t-conorms discused in the literature [18, 19, 24, 40, 100, 132, 136, 152], only a few of them are commonly used. The following are the definitions of four of the most popular t-conorms:

1. Union: $\mu_A \vee \mu_B = \max(\mu_A, \mu_B)$.
2. Algebraic sum: $\mu_A \hat{+} \mu_B = \mu_A + \mu_B - \mu_A \cdot \mu_B$.
3. Bounded sum: $\mu_A \oplus \mu_B = \min(1, \mu_A + \mu_B)$.
4. Drastic sum: $\mu_A \dot{\vee} \mu_B = \begin{cases} \mu_A, & \text{if } \mu_B = 0 \\ \mu_B, & \text{if } \mu_A = 0 \\ 1, & \text{if } \mu_A > 0 \text{ and } \mu_B > 0 \end{cases}$.

Intersection As in the case of the union, the intersection of two fuzzy sets is a fuzzy set whose membership function depends on the t-norm used. A t-norm operating on the two membership functions $\mu_A(\cdot)$ and $\mu_B(\cdot)$, is simply a two-parameter function, t, of the form

$$\begin{aligned} t: \quad [0,1] \times [0,1] &\longrightarrow [0,1] \\ x &\longmapsto \mu_{A\cap B}(x) = t(\mu_A(x), \mu_B(x)) \end{aligned}$$

Any t-norm function has to satisfy the following conditions:

i. Boundary Conditions: $t(0,0) = 0$, and $t(\mu_A(x), 1) = t(1, \mu_A(x)) = \mu_A(x)$.

ii. Commutativity: $t(\mu_A(x), \mu_B(x)) = t(\mu_B(x), \mu_A(x))$.

iii. Motonocity: if $\mu_A(x) \leq \mu_C(x)$ and $\mu_B(x) \leq \mu_D(x)$

then $t(\mu_A(x), \mu_B(x)) \leq t(\mu_C(x), \mu_D(x))$.

iv. *Associativity:* $t(\mu_A(x), t(\mu_B(x), \mu_C(x))) = t(t(\mu_A(x), \mu_B(x)), \mu_C(x))$.

Like in the t-conorms case, there is a wide range of t-norms that have been used and tested in the literature [18, 19, 24, 40, 100, 132, 136, 152], but only a few of them are commonly used. The following are the definitions of four of the most popular t-norms:

1. Intersection: $\mu_A \wedge \mu_B = \min(\mu_A, \mu_B)$.

2. Algebraic product: $\mu_A \cdot \mu_B = \mu_A \times \mu_B$.

3. Bounded product: $\mu_A \odot \mu_B = \max(0, \mu_A + \mu_B - 1)$.

4. Drastic product: $\mu_A \hat{\cdot} \mu_B = \begin{cases} \mu_A, & \text{if } \mu_B = 1 \\ \mu_B, & \text{if } \mu_A = 1 \\ 1, & \text{if } \mu_A < 1 \text{ and } \mu_B < 1 \end{cases}$.

Complement Another fundamental fuzzy set operation is the complement. The complement of a fuzzy set A is denoted by $\bar{A}$. Although several fuzzy complements have been defined in the literature, the most commonly used one is known as the *negation complement*. According to the negation complement, the membership function of the complement of A is defined as

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x).$$

For detailed description, one may consult [68].

A.3 Structure of Fuzzy Logic Controllers

The basic structure of an FLC is shown in figure A.1. A fuzzy controller is, generally,

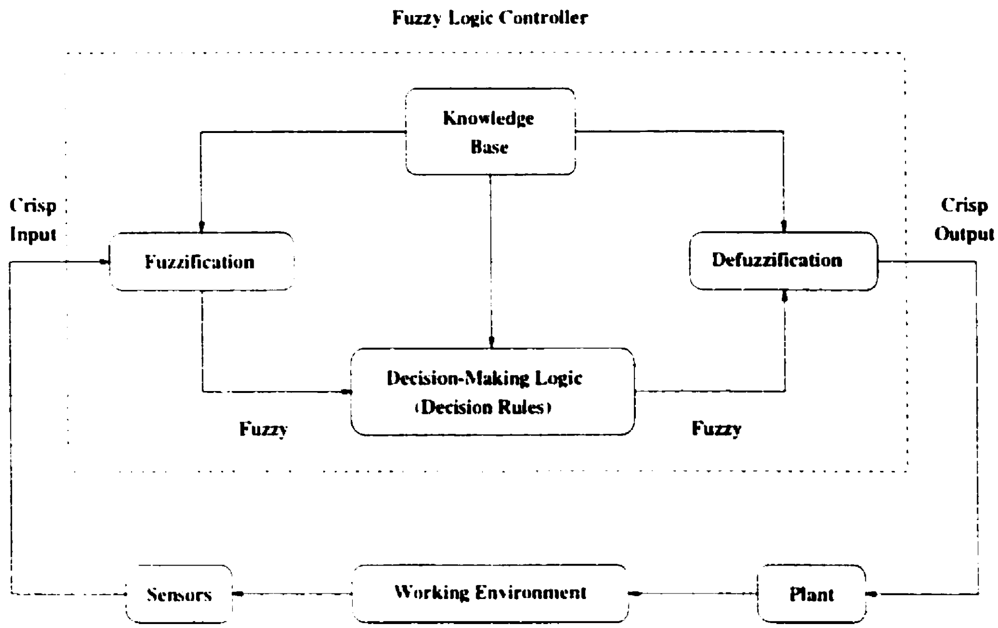


Figure A.1: The Structure of a Fuzzy Controller.

composed of three complementary components: a *fuzzifier*, an *inference engine*, and a *defuzzifier*.

Fuzzification First, the FLC fuzzifies its crisp valued input vector $x = (x_1, \ldots, x_n)^T \in U$ by mapping it into a fuzzy set in U. This is achieved by means of membership functions stored in the knowledge base. The fuzzifier is defined as a mapping from a crisp-valued input, from an observed input space, to labels of fuzzy sets in a specified input universe of discourse. The mapping is achieved through the membership functions that span the universe of discourse. The linguistic labels of the membership functions provide the controller with the power of computing with words rather than with numbers. All the information about the universe of discourse, the membership functions and their corresponding linguistic labels, is stored in the *knowledge base*.

The choice of the membership functions used to fuzzify the universe of discourse is crucial for the overall controller's accuracy and time efficiency [93, 129]. The most widely used membership functions are the *triangular*, *trapezoidal*, *Gaussian*, and *Bell* functions.

Inference Engine The inference engine simulates the rules stored in the *knowledge base*. The rule base, $R = \cup_{l=1}^{L} R^{(l)}$, is a union of L MIMO fuzzy rules (also called fuzzy control statements.) Each rule takes the following form

$$\begin{aligned} R^{(l)} &: \text{If } X_1 \text{ is } A_1^{(l)} \text{ and } \ldots \text{ and } X_n \text{ is } A_n^{(l)}, \\ &\quad \text{Then } Y_1 \text{ is } B_1^{(l)} \text{ and } \ldots \text{ and } Y_m \text{ is } B_m^{(l)}, \end{aligned} \tag{A.1}$$

or

$$\begin{aligned} R^{(l)} &: \text{If } X_1 \text{ is } A_1^{(l)} \text{ and } \ldots \text{ and } X_n \text{ is } A_n^{(l)}, \\ &\quad \text{Then } y_1 = f_1^{(l)}(x_1, x_2, \ldots, x_n) \text{ and } \ldots \text{ and } y_m = f_m^{(l)}(x_1, x_2, \ldots, x_n), \end{aligned} \tag{A.2}$$

where the X_i and Y_j, $i = 1, \ldots, n$ and $j = 1, \ldots, m$, with corresponding numeric values x_i and y_j, are input linguistic variables, $y = (y_1, \ldots, y_m)^T \in V$ is the output vector of the FLC. $A_i^{(l)}$, $i = 1, 2, \ldots, n$, $B_j^{(l)}$, $j = 1, 2, \ldots, m$, are fuzzy sets in U_i and V_j, respectively, and $f_i^{(l)}$, $i = 1, \ldots, m$, is any function in $x_1, x_2, \ldots, x_n$. Choosing the appropriate rules and membership functions is so important in any FLC as they characterize the input-output relation of the system.

The fuzzy inference engine can be regarded as the kernel of the fuzzy controller. It represents the decision making part of it, as it has the capability of simulating human decision making based on fuzzy concepts, and also of inferring fuzzy control actions employing fuzzy implication and fuzzy inference techniques.

Two typical implication operations are the *min* [77, 78] and the *product* [62] fuzzy implication operations. They are defined as

$$\textit{min:} \qquad \mu_{A\longrightarrow B}(x, y) = \min(\mu_A(x), \mu_B(y)).$$

$$\textit{product:} \qquad \mu_{A\longrightarrow B}(x, y) = \mu_A(x) \cdot \mu_B(y).$$

Generalized modus ponens plays an important role in fuzzy reasoning. A generalized form of the modus ponens can be written as follows:

Premise 1	*R*:	*IF* x is A *THEN* y is B
Premise 2		x is A'
Conclusion		y is B'

where A, A', B and B' are fuzzy predicates (fuzzy sets or relations). B' is then defined as

$$\begin{aligned} B' &= A' \circ R = A' \circ (A \longrightarrow B), \text{ and} \\ \mu_{B'}(\nu) &= \sup_{v} \{\mu_{A'}(v) * \mu_{A\longrightarrow B}(v, \nu)\}, \end{aligned}$$

where $\circ$ is the superstar composition operator [64, 149], and $*$ denotes the t-norm operator used.

Defuzzification The defuzzification process is a mapping from the output universe of discourse to a non-fuzzy (i.e.: crisp) output space. This process is necessary because, in real life, a crisp value is needed to actuate a control action. For instance, this value may represent the current, or voltage, input to an electric device; or the speed, or force, for an mechanical device.

Several defuzzification methods are discussed in the literature [23,64,137]. One of the most popular techniques is that which we have been using throughout our experiments. It is called the *center of area* (COA), or the center-average, defuzzification method. Suppose that the membership functions aggregation process, performed through the inferencing of the rules, resulted in an aggregated membership function $\mu_{C_j}(\cdot)$ for output y_j, $j = 1, \ldots, m$. Then, applying the COA defuzzification strategy to $\mu_{C_j}(\cdot)$ yields to a crisp output value

$$y_j = \frac{\int_z \mu_{C_j}(z)\, z\, dz}{\int_z \mu_{C_j}(z)\, dz}, \quad j = 1, \ldots, m. \tag{A.3}$$

A major drawback of the COA method is that it is expensive to compute and hence it may lead to rather slow inference cycles.

A.4 Types of Fuzzy Inference Systems

There are three basic types of fuzzy inference systems: the *Mamdani*, the *Sugeno-Takagi*, and the *Tsukomoto* [118] systems. They are basically the main components of a given fuzzy system to provide for the approximate decision signal. The first two types are the ones most commonly used in today's intelligent control systems.

The Mamdani Fuzzy Inference System The fuzzy rules in the Mamdani inference system are in the form given by equation (A.1). In this type, the qualified output of each fuzzy rule is obtained by computing its corresponding firing strength and the output membership function. After that, the overall fuzzy output is extracted by applying any t-norm and t-conorm operators to the qualified fuzzy outputs. To extract a crisp value from the aggregated output membership function, a defuzzification technique is applied. For instance, if the COA defuzzification method is used, the crisp output vector y of the FLC is determined by applying equation (A.3).

The Takagi-Sugeno Fuzzy Inference System The rules in the Takagi-Sugeno inference system [108] are in the form of equation (A.2). Since the consequent of each rule is a crisp value, there is no need for a defuzzification process in such a type of controllers. Hence, the overall output of the system is a weighted sum of the consequents of the rules, and it can be expressed as

$$y_j = \frac{\sum_{l=1}^{L} \alpha^{(l)} f_j^{(l)}(x_1, x_2, \ldots, x_n)}{\sum_{l=1}^{L} \alpha^{(l)}}, \quad j = 1, \ldots, m,$$

where l is the rule index and $\alpha^{(l)}$ is the truth value (or firing strength) of rule l. The expression of $\alpha^{(l)}$ is given by

$$\alpha^{(l)} = \prod_{i=1}^{n} \mu_{A_i^{(l)}}(x_i) = \mu_{A_1^{(l)}}(x_1) * \mu_{A_2^{(l)}}(x_2) * \cdots * \mu_{A_n^{(l)}}(x_n),$$

where $*$ stands for the t-norm operator.

Bibliography

[1] C. O. Alford and S. M. Belyeu. Coordinated control of two robot arms. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 468–473, March 1984.

[2] A. M. Annaswamy, F. P. Skantze, and A. Loh. Adaptive control of continuous time systems with convex/concave parametrization. *Automatica*, vol. 34, no. 1, pp. 33–49, 1998.

[3] P. J. Antsaklis and K. M. Passino, Eds. *An Introduction to Intelligent and Autonomous Control*. Kluwer Academic Publishers, MA, USA, 1993.

[4] E. Appleton and D. J. Williams. *Industrial Robot Applications*. Open University Press, 1987.

[5] K. J. Åström and B. Wittenmark. *Adaptive Control*. Addison-Wesley Publishing, 1989.

[6] M. Bergerman, Y. Xu, and Y. H. Liu. Robust control of cooperative underactuated manipulators. In *Proceedings of the 1998 IEEE/RSJ International*

Conference on Intelligent Robots and Systems, pp. 443–448, Victoria, BC, Canada, 1998.

[7] S. P. Chan. A neural network compensator for uncertainties in robotic assembly. *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 13, no. 2, pp. 127–141, 1995.

[8] Y. C. Chang and B. S. Chen. Robust tracking designs for both holonomic and nonholonomic constrained mechanical systems: adaptive fuzzy approach. *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 1, pp. 46–66, 2000.

[9] X. Cheung. On-line collision free path planning for service and assembly tasks by a two-arm robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1523–1528, 1995.

[10] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano. Task space dynamic analysis of multiarm system configurations. *International Journal of Robotics Research*, vol. 10, no. 6, December 1993.

[11] M. H. Choi, M. S. Ko, and B. H. Lee. Optimal load distribution for two cooperating robots using force ellipsoid. *Robotica*, vol. 11, pp. 61–72, 1993.

[12] M. K. Ciliz and M. Tomizuka. Modeling and compensation of frictional uncertainties in motion control: a neural network based approach. In *Proceedings of the American Control Conference*, vol. 5, pp. 3269–3273, 1995.

[13] C. Davis and P. Y. Peng. Reinforcement tuning of type ii fuzzy systems. In *Proceedings of the American Control Conference*, pp. 2315–2319, San Diego, CA, USA, 1999.

[14] A. De Luca and C. Manes. Hybrid force/position control for robots in contact with dynamic environments. In *Proceedings of Robot Control SYROCO 91*, pp. 377–382, Vienna, Austria, 1991.

[15] C. W. de Silva. *Intelligent Control Fuzzy Logic Applications*. CRC Press, 1995.

[16] A. Delgado, C. Kambhampati, and K. Warwick. Dynamic recurrent neural network for system identification and control. In *IEE Proceedings of Control Theory and Applications*, vol. 142, pp. 307–314, 1995.

[17] W. F. Dellinger and J. N. Anderson. Interactive force dynamics of two robotic manipulators grasping a nonrigid object. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2205–2210, 1992.

[18] J. Dombi. A general class of fuzzy operators, the demorgan class of fuzzy operators and fuzziness measures induced by fuzzy operators. *Fuzzy Sets and Systems*, vol. 8, pp. 149–163, 1982.

[19] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. New York: Academic Press, 1980.

[20] J. Duffy. The fallacy of modern hybrid control theory that is based on 'orthogonal complements' of twist and wrench spaces. *Journal of Robotic Systems*, vol. 7, no. 2, pp. 139-144, 1990.

[21] M. Dunne. An advanced assembly robot. *Industrial Robots*, vol. 2, pp. 249-262, 1979.

[22] R. Fierro and F. L. Lewis. Control of nonholonomic mobile robot using neural networks. *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 589–600, 1998.

[23] D. P. Filev and R. R. Yager. A generalized defuzzification method via BAD distributions. *International Journal of Intelligent Systems*, vol. 6, no. 7, pp. 687-697, 1991.

[24] M. J. Frank. On the simultaneous associativity of $f(x, y)$ and $x + y - f(x, y)$. *Aequationes Math.*, vol. 19, pp. 194-226, 1979.

[25] E. Freund. Decoupling and pole assignment in nonlinear systems. *Electronics Letters*, vol. 9, no. 16, pp. 373-374, 1973.

[26] E. Freund. The structure of decoupled non-linear systems. *International Journal of Control*, vol. 21, no. 3, pp. 443-450, 1975.

[27] E. Freund. Fast nonlinear control with arbitrary pole placement for industrial robots and manipulators. *International Journal of Robotic Research*, , no. 1, pp. 65–78, 1982.

[28] E. Freund and H. Hoyer. Pathfinding in multi-robot systems: solution and applications. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 103–111, 1986.

[29] E. Freund and M. Syrbe. Control of industrial robots by means of microprocessors. In *Lecture Notes in Control and Information Sciences*, pp. 167–185. Springer-Verlag, New York, 1977.

[30] S. Fujii and S. Kurono. Coordinated computer control of a pair of manipulators. In *Proceedings of the fourth World Congress of the International Federation for Theory of Machines and Mechanisms (IFToMM)*, pp. 411–417, 1975.

[31] S. S. Ge and C. C. Hang. Direct adaptive neural network control of robots. *International Journal of Systems Science*, vol. 27, no. 6, pp. 533–542, 1996.

[32] M. J. Grimble. *Model Reference Adaptive Control.* Prentice Hall, 1992.

[33] W. Gueaieb, F. Karray, and S. Al-Sharhan. A robust adaptive fuzzy position/force control scheme for cooperative manipulators. *IEEE Transactions on Control Systems Technology.* (Submitted).

[34] W. Gueaieb, F. Karray, and S. Al-Sharhan. A robust hybrid intelligent position/force control scheme for cooperative manipulators. *IEEE Transactions on Robotics and Automation.* (Submitted).

[35] W. Gueaieb, F. Karray, and S. Al-Sharhan. Computational intelligence based approach for the joint trajectory generation of cooperative robotic systems.

In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 3687 3692, Nashville, TN, USA, 2000.

[36] W. Gueaieb, F. Karray, and S. Al-Sharhan. An adaptive fuzzy control approach for cooperative manipulators. In *Proceedings of the 2001 International Symposium on Intelligent Control (ISIC'01)*, 2001. (Submitted).

[37] W. Gueaieb, F. Karray, and S. Al-Sharhan. A robust hybrid intelligent position/force control scheme for cooperative manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington DC, USA, 2002. (Submitted).

[38] W. Gueaieb, F. Karray, and A. K. C. Wong. Computational intelligence based approach for the joint trajectory generation of cooperative robotic systems. *Journal of Systems Analysis, Modeling and Simulation*. (To appear).

[39] W.G. Guo, S. Komada, M. Ishida, and T. Hori. Coordinated motion of two redundant manipulators by simple hybrid position/force controller with disturbance observer. In *Proceedings of the IEEE International Conference on Industrial Technology (ICIT'96)*, pp. 493–497, 1996.

[40] H. Hamacher. Über logische verknupfungen unscharfer aussagen und deren zugehörige bewertungsfunktionen. In R. Trappl, G. J. Klir, and L. Ricciardi, Eds., *Progress in Cybernetics and Systems Research*, pp. 276–288, Washington DC, USA, 1978.

[41] A. Hemami. Kinematics of two-arm robots. *IEEE Journal Robotics and Automation*, vol. RA-2, no. 4, pp. 225–228, 1986.

[42] H. Hemami and B. Wyman. Indirect control of the force of constraint in dynamic system. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, vol. 101, no. 4, pp. 355–360, 1979.

[43] B. Hodges. *Industrial Robotics*. Newnes, 1992.

[44] J. Hoel. Some variations of Lee's algorithm. *IEEE Transactions on Computers*, vol. 25, no. 1, pp. 19–24, 1976.

[45] N. Hogan. Impedance control: An approach to manipulation, part 1 - theory, part 2 - implementation, part 3 - application. *Journal of Dynamic Systems, Measurement and Control*, vol. 107, pp. 1–24, 1985.

[46] F. Y. Hsu and L. C. Fu. Intelligent robot deburring using adaptive fuzzy hybrid position/force control. *IEEE Transactions on Robotics and Automation*, vol. 16, no. 4, pp. 325–335, 2000.

[47] Y. R. Hu and A. A. Goldenberg. An adaptive approach to motion and force control of multiple coordinated robots. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, vol. 115, no. 1, pp. 60–69, 1993.

[48] V. D. Hunt. *Understanding Robotics*. Academic Press, 1990.

[49] J. S. R. Jang. ANFIS: Adaptive-Network-based Fuzzy Inference System. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665-685, 1993.

[50] J. S. R. Jang, C. T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Prentice Hall, 1997.

[51] K. Jezernik, B. Curk, and J. Harnik. Observer-based sliding mode control of a robotic manipulator. *Robotica*, vol. 12, pp. 443 448, 1994.

[52] M. Kahn and B. Roth. The near-minimum-time control of open-loop articulated kinematic chains. *Transactions ASME, Journal of Systems Measurement and Control*, vol. 93, pp. 164-172, 1972.

[53] F. Karray, W. Gueaieb, and S. Al-Sharhan. Soft computing techniques as applied to expert tuning of PID controllers. *IEEE Transactions on Systems, Man and Cybernetics*. (To appear).

[54] F. Karray, W. Gueaieb, and S. Al-Sharhan. Soft computing techniques as applied to expert tuning of PID controllers. In *Proceedings of the 2000 IEEE International Symposium on Intelligent Control*, pp. 91-96, 2000.

[55] H. Kazerooni, T. B. Sheridan, and P. K. Houpt. Robust compliant motion for manipulators, part 1 - the fundamental concepts of compliant motion. *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 2, pp. 83-92, 1986.

[56] A. J. Koivo. Adaptive position-velocity-force control of two manipulators. In *Proceedings of the 34th IEEE Conference on Decision and Control*, pp. 334-337, Lauderdale, FL, 1985.

[57] S. Komada, T. Kimura, M. Ishida, and T. Hori. Robust position control of manipulators based on disturbance observer and inertia identifier in task space. In *Proceedings of 4th IEEE International Workshop on Advanced Motion Control - AMC '96 - MIE*, vol. 1, pp. 225–230, 1996.

[58] S. Komada, T. Miyakami, M. Ishida, and T. Hori. Simple control strategy of redundant manipulators by disturbance observer. In *Proceedings of the 1996 IEEE IECON. 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, vol. 3, pp. 1884–1889, 1996.

[59] S. Komada and K. Ohnishi. Force feedback control of robot manipulator by the acceleration tracing orientation method. *IEEE Transactions on Industrial Electronics*, vol. 37, no. 4, pp. 6–12, 1990.

[60] E. Kwan and M. Liu. An adaptive fuzzy approach for robot manipulator tracking. In *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA '99*, pp. 53–58, 1999.

[61] W. Kwon and B. H. Lee. Optimal force distribution of multiple cooperatng robots using nonlinear programming dual method. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2408–2413, 1996.

[62] P. M. Larsen. Industrial applications of fuzzy logic control. *International Journal of Man-Machine Studies*, vol. 12, no. 1, pp. 3–10, 1980.

[63] B. H. Lee and C. S. G. Lee. Collision-free motion planning of two robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 21–32, 1987.

[64] C. C. Lee. Fuzzy logic in control systems: Fuzzy logic controller. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 2, pp. 404–435, 1990. Parts I and II.

[65] F. L. Lewis, K. Liu, and A. Yeşildirek. Neural net robot controller with guaranteed tracking performance. *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 703–715, 1995.

[66] Q. Li, A. N. Poo, C. L. Teo, and C. M. Lim. Developing a neurocompensator for the adaptive control of robots. In *IEE Proceedings of Control Theory and Applications*, vol. 142, pp. 562–568, 1995.

[67] Q. Liang and J. M. Mendel. Interval type-2 fuzzy logic systems: Theory and design. *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 535–550, 2000.

[68] C. T. Lin and C. S. G. Lee. *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Prentice Hall, 1996.

[69] Y. H. Liu and S. Arimoto. Distributively controlling two robots handling an object in the task space without any communication. *IEEE Transactions on Automatic Control*, vol. 41, no. 8, pp. 1193–1199, 1996.

[70] Y. H. Liu and S. Arimoto. Decentralized adaptive and nonadaptive position/force controllers for redundant manipulators in cooperations. *International Journal of Robotics Research*, vol. 17, no. 3, pp. 232–247, 1998.

[71] Y. H. Liu, S. Arimoto, V. Parra-Vega, and K. Kitagaki. Adaptive distributed cooperation controller for multiple manipulators. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 1, pp. 489–494, 1995.

[72] Y. H. Liu, S. S. Arimoto, and K. Kitagaki. Adaptive control for holonomically constrained robots: time-invariant and time-variant cases. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 905–912, 1995.

[73] T. Lozano-Pérez and M. Wesley. An algorithm for planning collision free paths among polyhedral obstacles. *Communications ACM*, vol. 22, no. 10, pp. 560–570, 1979.

[74] J. Y. S. Luh and Y. F. Zheng. Constrained relations between two coordinated industrial robots for motion control. *International Journal of Robotics Research*, vol. 6, no. 3, pp. 60–70, 1987.

[75] Z. Luo and M. Ito. Control design of robot for compliant manipulation on dynamic environments. *IEEE Transactions on Robotics and Automation*, vol. 9, no. 3, pp. 286–296, 1993.

[76] Z. Luo, Y. Uematsu, K. Ito, A. Kato, and M. Ito. On cooperative manipulation of dynamic object. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, vol. 2, pp. 1064–71, 1994.

[77] E. H. Mamdani. Advances in the linguistic synthesis of fuzzy controllers. *International Journal of Man-Machine Studies*, vol. 8, no. 6, pp. 669–678, 1976.

[78] E. H. Mamdani. Application of fuzzy logic to approximate reasoning using linguistic systems. *IEEE Transactions on Computers*, vol. C-26, pp. 1182–1191, 1977.

[79] R. V. Mayorga and A. K. C. Wong. A robust method for the concurrent motion planning of multi-manipulators systems. *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 19, no. 1, pp. 73–88, 1997.

[80] R. K. Mudi and N. R. Pal. A self-tuning fuzzy PI controller. *Fuzzy Sets and Systems*, vol. 115, no. 2, pp. 327–338, 2000.

[81] M. A. Nahon and J. Angeles. Real-time force optimization in parallel kinematic chains under inequality constraints. *IEEE Transactions on Robotics and Automation*, vol. 8, no. 4, pp. 439–450, 1992.

[82] Y. Nakamura, K. Nagai, and T. Yoshikawa. Mechanics of coordinative manipulation by multiple robotic mechanisms. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 991–998, Raleigh, NC, 1987.

[83] M. Nakao, K. Ohnishi, and K. Miyachi. A robust decentralized joint control based on interference estimation. In *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 326–31, Raleigh, NC, 1987.

[84] T. Naniwa, S. Arimoto, and V. Parra-Vega. A model-based adaptive control scheme for coordinated control of multiple manipulators. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, vol. 1, pp. 695–702, Munich, Germany, 1994.

[85] K. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[86] D. E. Orin and S. Y. Oh. Control of force distribution in robotic mechanisms containing closed kinematic chains. *Transactions of the ASME Journal of Dynamic Systems Measurement and Control*, vol. 103, no. 2, pp. 134–141, 1981.

[87] G. Oriolo, G. Ulivi, and M. Vendittelli. Real-time map building and navigation for autonomous robots in unknown environments. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 28, no. 3, pp. 316–333, 1998.

[88] E. Paljug, X. Yun, and V. Kumar. Control of rolling contacts in multiple robotic manipulation. In *Proceedings the International Conference on Advanced Robotics*, pp. 591–596, 1991.

[89] E. Paljung and X. Yun. Experimental study of two robot arms manipulating large objects. *IEEE Transactions on Control Systems Technology*, vol. 64, pp. 21–30, 1994.

[90] J. C. Parrish and D. L. Akin. The ranger telerobotic flight experiment: Mission, technologies, and programmatics. In Laura A. Demsetz, Ed., *Robotics For Challenging Environments*, pp. 136–142. American Society of Civil Engineers, 1996. Proceedings of RCE II, the Second Conference.

[91] K. M. Passino and S. Yurkovich. *Fuzzy Control*. Addison-Wesley, 1998.

[92] R. P. Paul. Problems and research issues associated with the hybrid control of force and displacement. In *Proceedings the IEEE International Conference on Robotics and Automation*, pp. 1966–1971, 1987.

[93] W. Pedrycz. Why triangular membership functions? *Fuzzy Sets and Systems*, vol. 3, no. 2, pp. 177–188, 1995.

[94] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 102, pp. 126–133, 1981.

[95] A. K. Ramadorai, T. J. Tarn, and A. K. Bejczy. Task definition, decoupling and redundancy resolution by nonlinear feedback in multi-robot object handling. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pp. 467–474, 1992.

[96] P. Ramaswamy, R. M. Edwards, and K. Y. Lee. An automatic tuning method of a fuzzy logic controller for nuclear reactors. *IEEE Transactions on Nuclear Science*, vol. 40, no. 4, pp. 1253–1262, 1993.

[97] A. Rojko and K. Jezernik. Disturbance rejection by pi estimator in position robot control. In *Proceedings of ISIE '99. IEEE International Symposium on Industrial Electronics*, vol. 3, pp. 1056–1061, 1999.

[98] W. J. Rugh. Analytical framework for gain scheduling. *IEEE Control Systems Magazine*, vol. 11, no. 1, pp. 79–84, 1991.

[99] A. Saffiotti. Robot navigation under approximate self-localization. In F. Pin, M. Jamshidi, and P. Dauchez, Eds., *Robotics and Manufacturing vol. 6 – Proceedings of the 6th International Symposium (ISRAM)*, pp. 589–594, ASME Press, New York, NY, USA, 1996.

[100] B. Schweiser and A. Sklar. Associative functions and statistical triangle inequalities. *Publ. Math.*, vol. 8, pp. 169–186, 1961.

[101] X. Shi and P. J. A. Lever. Fuzzy behavior integration and action fusion for robotic excavation. *IEEE Transactions on Industrial Electronics*, vol. 43, pp. 39–402, 1996.

[102] X. Shi, P. J. A. Lever, and F. Y. Wang. Experimental robotic excavation with fuzzy logic and neural networks. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 957–962, 1996.

[103] T. Shibata and T. Fukuda. Coordination in evolutionary multi-agent-robotic system using fuzzy and genetic algorithm. *Control Engineering Practice*. vol. 2. no. 1. pp. 103–111. 1994.

[104] J. E. Slotine and W. Li. On the adaptive control of robot manipulators. *The International Journal of Robotic Research*. vol. 6. no. 3. pp. 49–59. 1987.

[105] C. Sue and Y. Stepanenko. Adaptive sliding mode coordinated control of multiple robot arms attached to a constrained object. *IEEE Transactions on Systems, Man and Cybernetics*. vol. 25. pp. 871–878. 1995.

[106] H. Suh and K. G. Shin. Coordination of dual robot arms using kinematic redundancy. In *Proceedings of IEEE International Conference on Robotics and Automation*. pp. 504–509. 1998.

[107] F. C. Sun. Z. Q. Sun. and G. Feng. An adaptive fuzzy controller based on sliding mode for robot manipulators. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*. vol. 29. no. 5. pp. 661–667. 1999.

[108] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modelling and control. *IEEE Transactions on Systems, Man, and Cybernetics*. vol. 15. no. 1. pp. 116–132. 1985.

[109] M. Takahashi. Biomedical applications of fuzzy logic controllers. In *Proceedings of the International Conference on Fuzzy Logic and Neural Networks*. pp. 553–556. Japan. 1990.

[110] G. Tao. On robust adaptive control of robot manipulators. *Automatica*, vol. 28, no. 4, pp. 803–807, 1992.

[111] T. J. Tarn, A. K. Bejczy, and X. Yun. Design of dymanic control of two cooperating robot arms: Closed chain formulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 7–13, 1987.

[112] T. J. Tarn, A. K. Bejczy, and X. Yun. New nonlinear control algorithms for multiple robot arms. *IEEE Transactions on Aerospace and Electronic Systems*, pp. 571–581, 1988.

[113] P. Tomei. Robust adaptive control of robots with arbitrary transient performance and disturbance attenuation. *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 654–658, 1999.

[114] P. Tomei. Robust adaptive friction compensation for tracking control of robot manipulators. *IEEE Transactions on Automatic Control*, vol. 45, no. 11, pp. 2164–2169, 2000.

[115] S. Tong, J. Tang, and T. Wang. Fuzzy adaptive control of multivariable nonlinear systems. *Fuzzy Sets and Systems*, vol. 111, no. 2, pp. 153–167, 2000.

[116] W. B. Tong, S. Y. T. Lang, and S. K. Tso. Conjoint operation of multiple mobile robots in an unknown environment using fuzzy logic. In *Proceedings of the American Control Conference (ACC)*, 1999.

[117] A. V. Topalov, J. H. Kim, and T. P. Proychev. Fuzzy-net control of non-holonomic mobile robot using evolutionary feedback-error-learning. *Journal of Robotics and Autonomous Systems*, vol. 23, pp. 187–200, 1998.

[118] Y. Tsukamoto. An approach to fuzzy reasoning method. In M. M. Gupta, R. K. Ragade, and R. R. Yager, Eds., *Advances in Fuzzy Set Theory and Applications*, pp. 137–149, North-Holland, Amsterdam, Holland, 1979.

[119] E. Tunstel, H. Danny, T. Lippincott, and M. Jamshidi. Autonomous navigation using an adaptive hierarchy of multiple fuzzy behaviors. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 276–281, Monterey, CA, USA, 1997.

[120] P. Tyridal. New ideas in multi-task real-time control systems for industrial robots. In *Proceedings of the 10th International Symposium on Industrial Robots*, pp. 659–670, Milan, 1980.

[121] C. S. Tzafestas. Path planning and control of a cooperative three-robot system manipulating large objects. *Journal of Intelligent and Robotic Systems*, vol. 22, pp. 99–116, 1998.

[122] M. A. Unseren. Rigid body dynamics and decoupled control architecture for two strongly interacting manipulators. *Robotica*, pp. 421–430, 1991.

[123] C. Von Altrock, B. Krause, and H. J. Zimmermann. Advanced fuzzy logic control of a modern car in extreme situations. *Fuzzy Sets and Systems*, vol. 48, pp. 41–52, 1992.

[124] M. Vukobratovic. How to control robots interacting with dynamic environment. *Journal of Intelligent and Robotic Systems*, vol. 19, pp. 119–152, 1997.

[125] M. Vukobratovic and A. Tuneski. Contribution to the adaptive control of multiple compliant manipulation of dynamic environments. *Robotica*, vol. 17, pp. 97–109, 1999.

[126] J. Wang, S. J. Dodds, and W. N. Bailey. Co-ordinated control of multiple robotic manipulators handling a common object-theory and experiments. *IEE Proceedings-Control Theory and Applications*, vol. 144, no. 1, pp. 73–86, 1997.

[127] L. X. Wang. Stable adaptive fuzzy control of nonlinear systems. *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 146–155, 1993.

[128] L. X. Wang. *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. PTR Prentice Hall, 1994.

[129] L. X. Wang. *A Course in Fuzzy Systems and Control*. Prentice Hall, 1997.

[130] S. D. Wang and C. K. Lin. Adaptive tuning of the fuzzy controller for robots. *Fuzzy Sets and Systems*, vol. 110, pp. 351–363, 2000.

[131] T. J. Wen and K. Kreutz. Motion and force control for multiple cooperative manipulators. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1246–1251, Phoenix, AZ, USA, 1989.

[132] B. Werners. An interactive fuzzy programming system. *Fuzzy Sets and Systems*, vol. 23, pp. 131–147, 1987.

[133] D. E. Whitney. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions MMS-10*, pp. 47-53, 1969.

[134] Z. W. Woo, H. Y. Chung, and J. J. Lin. A PID type fuzzy controller with self-tuning scaling factors. *Fuzzy Sets and Systems*, vol. 115, no. 2, pp. 321–326, 2000.

[135] L. C. Woon, S. S. Ge, X. Q. Chen, and C. Zhang. Adaptive neural network control of coordinated manipulators. *Journal of Robotic Systems*, vol. 16, no. 4, pp. 195–211, 1999.

[136] R. R. Yager. On a general class of fuzzy connectives. *Fuzzy Sets and Systems*, vol. 4, pp. 235–242, 1980.

[137] R. R. Yager. Fuzzy sets and approximate reasoning in decision and control. In *Proceedings of IEEE International Conference on Fuzzy Systems*, pp. 415–428, 1992.

[138] H. C. Yan and A. J. Koivo. Self-motions for two robot closed chain system. In *Proceedings of IEEE International Symposium on Intelligent Control*, pp. 147–152, 1994.

[139] B. Yao and M. Tomizuka. Adaptive coordinated control of multiple manipulators handling a constrained object. In *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 624–629, 1993.

[140] B. K. Yoo and W. C. Ham. Adaptive control of robot manipulator using fuzzy compensator. *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 2, pp. 186–199, 2000.

[141] T. Yoshikawa, T. Sugie, and M. Tanaka. Dynamic hybrid position/force control of robot manipulators - controller design and experiment. *IEEE Journal of Robotics and Automation*, vol. 4, no. 6, pp. 699–705, 1988.

[142] T. Yukawa, M. Uchiyama, and H. Inooka. Handling of a constrained flexible object by a robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 324–329, 1995.

[143] T. Yukawa, M. Uchiyama, D. N. Nenchev, and H. Inooka. Stability of control system in handling of a flexible object by rigid arm robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2332–2339, 1996.

[144] X. Yun. Nonlinear feedback control of two manipulators in presence of environment constraints. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1252–1257, Phoenix, AZ, USA, 1989.

[145] X. Yun. Coordination of two-arm pushing. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 182–187, 1991.

[146] X. Yun. Object handling using two arms without grasping. *International Journal Robotics Research*, vol. 12, no. 1, pp. 99–106, February 1993.

[147] X. Yun, V. Kumar, N. Sarkar, and E. Paljug. Control of multiple arms with rolling constraints. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2193–2198, 1992.

[148] L. Zadeh. Fuzzy sets. *Information Control*, vol. 8, pp. 338–353, 1965.

[149] L. Zadeh. Outline of a new approach to analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-3, pp. 28–44, 1973.

[150] Y. E. Zheng and J. Y. S. Luh. Optimized load distribution for two industrial robots handling a single object. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 344–349, 1988.

[151] Y. H. Zheng and J. Y. S. Luh. Joint torques for control of two coordinated moving robots. In *IEEE Conference on Robotics and Automation*, pp. 1375–1380, 1986.

[152] H. J. Zimmermann and P. Zysno. Latent connectives in human decision making. *Fuzzy Sets and Systems*, vol. 4, pp. 37–51, 1980.