UNIVERSITÉ DU QUÉBEC

THÈSE PRÉSENTÉE À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DU DOCTORAT EN GÉNIE ÉLECTRIQUE

PAR
HICHAM CHAOUI

COMMANDE ADAPTATIVE DE SYSTÈMES À DYNAMIQUE COMPLEXE BASÉE SUR L'INTELLIGENCE ARTIFICIELLE

SEPTEMBRE 2011

UNIVERSITY OF QUEBEC

THESIS SUBMITTED TO
UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING

BY
HICHAM CHAOUI

SOFT-COMPUTING BASED INTELLIGENT ADAPTIVE CONTROL DESIGN OF COMPLEX DYNAMIC SYSTEMS

SEPTEMBER 2011

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

DOCTORAT EN GÉNIE ÉLECTRIQUE (PH.D.)

Programme offert par L'Université du Québec à Trois-Rivières

Soft-Computing Based Intelligent Adaptive Control Design of Complex Dynamic Systems

PAR

HICHAM CHAOUI

Pierre Sicard, directeur de recherche	Université du Québec à Trois-Rivières
Ahmed Chériti, président du jury	Université du Québec à Trois-Rivières
Adel-Omar Dahmane, évaluateur	Université du Québec à Trois-Rivières
Philippe Lautier, évaluateur externe	Vestas Wind Systems

Thèse soutenue le 10 juin 2011

"If we knew what we were doing it wouldn't be research, would it ?"

Albert Einstein.

Einstein showed in the early years of the 20th century that time is a relative concept, and hence, it is subjected to change according to the environment (time is dependent on mass and velocity). This fact has been proved with Einstein's theory of relativity.

بِسمِ اللهِ الرَّحمٰنِ الرَّحِيم

وَ إِنَّ يَومَاً عِندَ رَبِكَ كَأَلفِ سَنَةٍ مِمَا تَعُدُون

سُورَةُ السَجدَة (٣٢)، أَيَة ٥.

In the name of God. Most Gracious, Most Merciful.

"A day with your Lord is equivalent to a thousand years in the way you count"

Sourate Sajda (32), Verse 5.

بِسمِ اللهِ الرَّحمٰنِ الرَّحِيم

وَ قُل رَبِ زِدنِي عِلمَاً

سُورَةُ طَهَ (٢٠)، أَيَة ١١٤.

In the name of God, Most Gracious, Most Merciful.

"O my Lord, advance me in knowledge"

Sourate Tâ-Hâ (20), Verse 114.

Résumé

En dépit des récentes avancées dans le domaine du contrôle des systèmes non-linéaires, les techniques classiques de contrôle dépendent en grande partie sur des modèles mathématiques précis du système à contrôler pour fournir des performances satisfaisantes. En pratique, à cause de non-linéarités élevées, dériver un modèle mathématique décrivant avec précision la dynamique du système à contrôler peut s'avérer une tâche très difficile. Bien que des stratégies de contrôle comme la commande adaptative et le mode de glissement compensent pour les incertitudes paramétriques, ces méthodes sont encore vulnérables en présence d'incertitudes de modélisation, aussi appelées incertitudes non structurées. D'autre part, les contrôleurs à base d'intelligence artificielle n'ont pas une telle limitation puisqu'ils ne dépendent pas d'une représentation mathématique du système à contrôler. Malgré les récents résultats, les contrôleurs à base de réseaux de neurones demeurent incapables d'intégrer de l'expertise sous forme de règles et les contrôleurs à base de la logique floue sont incapables d'incorporer des connaissances déjà acquises sur la dynamique du système.

Basé sur la motivation ci-dessus, cette thèse a pour but de contribuer à l'évolution récente et les mérites de ces outils par le développement de nouvelles structures de commande adaptative. Les contrôleurs proposés supposent que la dynamique des systèmes est incertaine ou inconnue pour obtenir une robustesse à la fois des incertitudes structurées et non structurées de grandeurs et natures différentes. Les contrôleurs classiques offrent une faible performance en présence de ces sortes d'incertitudes. Contrairement à ces approches, les contrôleurs proposés sont basées sur l'intelligence artificielle, qui n'ont pas de telles limitations, grâce à leur capacité d'apprentissage et de généralisation. Cependant, ces outils sont souvent basés sur des heuristiques et le réglage peut ne pas être évident. En outre, de nombreux contrôleurs intelligents soufrent de manque de preuves de stabilité dans les différentes applications de commande. Dans cette thèse, les architectures de contrôle proposées sont conçues en utilisant des techniques d'adaptation à base de Lyapunov au lieu des méthodes conventionnelles d'adaptation heuristique. Ainsi, la stabilité est garantie contrairement à beaucoup de systèmes de contrôle basés sur l'intelligence artificielle. Un résumé substantiel est disponible en annexe.

Abstract

Nonlinear dynamical systems face numerous challenges that need to be addressed such as, severe nonlinearities, varying operating conditions, structured and unstructured dynamical uncertainties, and external disturbances. In spite of the recent advances in the area of nonlinear control systems, conventional control techniques depend heavily on precise mathematical system models to provide satisfactory performance. In real life, and due to high nonlinearities, deriving a precise model could be a difficult undertaking. Although conventional nonlinear control strategies, such as adaptive and sliding mode controllers, compensate for parametric uncertainties, they are still vulnerable in the presence of unstructured modeling uncertainties. On the other hand, computational intelligence based controllers do not have such a limitation, thanks to their mathematical model dependence free characteristic. Despite of the recent results, neural network-based controllers remain incapable of incorporating any human-like expertise and fuzzy logic-based controllers are unable to incorporate any learning already acquired about the dynamics of the system in hand.

Driven by the aforementioned motivation, this thesis is meant to contribute to the latest developments and merits of such tools by novel adaptive control methodologies developments. The proposed controllers assume uncertain/unknown systems dynamics to achieve robustness to both structured and unstructured uncertainties of higher and different magnitudes. Conventional control structures offer poor performance in the presence of these kinds of uncertainties. Unlike these approaches, the proposed controllers are based on soft-computing tools, which do not have such limitations, thanks to their learning and generalization capabilities. However, these tools are often based on heuristics and tuning may not be trivial. Furthermore, many soft-computing based controllers lack stability proofs in various control applications. In this thesis, the proposed control architectures are designed using Lyapunov-based adaptation techniques instead of conventional heuristic tuning methods. Thus, the stability of the proposed controllers is guaranteed unlike many computational intelligence-based control schemes.

Acknowledgements

First and foremost, words cannot express the gratitude I owe the Lord for his guidance, relief in difficult moments, and success. In fact, if I try to count all His bounties, I won't be able to number them.

I also would like to thank the thesis advisor, Dr. Pierre Sicard, for his support and insight throughout my doctoral studies. Many thanks for being human in the way he treats his students, the way he encourages and respects them. I also feel so grateful for his support during both my master degrees.

I wish to thank the reviewers for their thorough review, very relevant comments and for their suggestions to improve the quality of this thesis.

Special thanks to Dr. Alfred Ng and Dr. James Lee from the Canadian Space Agency (CSA) for their time and support. Thanks for sharing with us the difficulties of spacecraft formation missions, as well as the free-flyer experimental data to validate our approach.

I thank my parents, Mr. Mohamed Chaoui and Mrs. Hania Daoudi, as well as my family members for their support throughout all my studies. Mom, your dream now becomes true. My parents have made many personal sacrifices to provide me the best possible education, it is impossible to put words on their contribution.

Finally I would like to thank my wonderful sister, Samira, and my lovely wife, Laurie, for their patience and encouragements over the years.

Thanks to all those who have contributed to this work.

To all engineers working in the field of renewable energy to make our planet greener.

To all researchers who love contributing to soft-computing techniques.

To all control systems researchers worldwide.

To all my friends around the world.

To all my family members.

To all whom it may concern.

To you all : believe in your dreams and do NOT let anyone tell you that you cannot do it.

Table des matières

Liste des tableaux

Table des figures

List of Acronyms

ANN	Artificial neural network
FLS	Fuzzy logic system
FLC	Fuzzy logic controller
MEMS	Microelectromechanical systems
MLP	MultiLayer perceptron
RBF	Radial basis function
FOU	Footprint of uncertainty
EA	Eevolutionary algorithm
GA	Genetic algorithm
ACO	Ant colony optimization
LS	Least squares
RLS	Recursive least squares
LMS	Least mean squares
MRAS	Model reference adaptive system
MRAC	Model reference adaptive control
CSA	Canadian space agency
JAXA	Japanese aerospace exploration agency
SIMO	Single input multiple output
MIMO	Multiple input multiple output
PI	Proportional integral

PD	Proportional derivative
PID	Proportional integral derivative
FF	Feedforward
FBK	Feedfback
PMSM	Permanent magnet synchronous machine
EKF	Extended Kalman filter
DTC	Direct torque control
MPC	Model predictive control
PWM	Pulse width modulation
SVPWM	Space vector pulse width modulation
SOC	State of charge
OCV	Open circuit voltage
BMS	Battery management system
FNN	Fuzzy neural network
FNSM	Fuzzy neural sliding mode

Chapitre 1

Introduction

1.1 Motivation

In the literature, control law design approaches for nonlinear systems can be divided into three categories. The first category consists of a nonlinear systems linearization around an operating point of the states [1]. In this case, classical linear control laws are applied for the approximated system. In spite of the control laws simplicity, the control system performance is not guaranteed for the overall system. The second category deals with nonlinear controllers design based on nonlinear systems dynamics. In this category, the characteristics of nonlinear systems are preserved. However, the design approach difficulties arise with the complexity of the nonlinear systems dynamics [2]. Furthermore, these approaches assume a precise mathematical system model and tend to work quite well in theory. But, their performance degrades in the presence of varying operating conditions, structured and unstructured dynamical uncertainties, and external disturbances. In real life applications, deriving a precise mathematical model for complex industrial processes might be a difficult task to undertake. In addition, other factors might be unpredictable, such as noise, temperature, and parameters variation. Hence, the system's dynamics cannot be efficiently based on presumably accurate mathematical models. The third category implements nonlinear controllers based on computational intelligence tools, such as artificial neural networks (ANNs) and fuzzy logic systems (FLSs) [3–8]. These techniques have been credited in various applications as powerful tools capable of providing robust approximation for mathematically ill-defined systems that may be subjected to structured and unstructured uncertainties [9, 10]. The universal approximation theorem has been the main driving force behind the increasing popularity of such methods as it shows that they are theoretically capable of uniformly approximating any continuous real function to any degree of accuracy. Various artificial neural network and fuzzy logic models have been proposed to solve many complex problems, which have led to a satisfactory performance [11–14], providing an alternative to conventional control techniques. To clearly show the power of neural networks, next section presents an application of neural networks in a learning identification strategy for spacecraft formation flying. This technique is used for thrusters' dynamics approximation of a nanosatellite, also called free-flyer. Its effectiveness is investigated using experimental data of a single thruster.

1.2 Spacecraft Formation Flying

Spacecraft formation flying is defined as two or more spacecrafts flying autonomously in a coordinated fashion and adapting their relative position and orientation to form a formation. It has been identified as an enabling technology with several benefits for space missions and recently became an important field of research in the space industry. Formation flying offers better performance, cost reduction, high failure tolerance, extended mission life, more adaptability to changing mission goals as well as larger spatial coverage. It is expected that formation flying satellites will be more reliable than their single satellite counterparts. However, this technology faces many challenges in coordination and formation trajectory generation. Environmental disturbances such as gravitational perturbation, atmospheric drag, solar radiation pressure and electromagnetic forces make guidance, navigation, and control tasks become more complex for larger formations. Hence, an efficient control algorithm easy to implement on board is required to fulfill the control requirements of a mission.

Nanosatellites require a miniature propulsion system with high propulsion capabilities.

Many researchers investigated propulsion technologies to identify those that maintain high performance at small scale. However, independently of the size, thrusters emit caustic propellant exhaust that contaminates neighboring spacecraft. This contamination problem is amplified as the formation spacing is reduced. Canadian Space Agency (CSA) initiated, as an international collaboration with the Japanese Aerospace Exploration Agency (JAXA), a feasibility study of JC2Sat-FF nanosatellite mission. It consists of the use of aerodynamic drag to maintain spacecraft formation of two nanosatellites. The main advantage is that there is no contamination from propellant exhaust and the missions' lifetime will not be limited by the amount of fuel on board. However, response times to changes in the formation are slow, the relative positions cannot be controlled to a very high accuracy and some types of formation are not possible. A theoretical, algorithmic and experimental study of spacecraft formation flying is conducted at the CSA, where, a test-bed has been developed to emulate a typical formation flying scenario for experimental validation using hardware-in-the-loop technology.

Recently, thrusters have received a thorough attention and have been used in many applications such as, high performance surface vessels and underwater robotics applications. The nonlinearities within thrusters make the modeling of their dynamics a difficult task to undertake. This has led to the use of a simplified thruster model. However, control systems based on this model as well as conventional control techniques often result in poor performance because of the nonlinear and time-varying thruster dynamics. Therefore, an advanced control system with learning capabilities is required to adapt to changes in the thruster dynamics.

1.2.1 Modeling

The free-flyer of the formation flying testbed [15] is shown in Fig 1.1. The testbed is used to emulate a typical formation flying scenario with the frictionless nature of space. In space, spacecrafts are free to move in 6 degrees of freedom (3 translations, 3 rotations) and obey the orbital dynamics. However, on the testbed, the free-flyer can only move in 3 degrees of freedom (2 translations, 1 rotation).

The dynamic equations of motion in two dimensions of the i^{th} free-flyer in the formation are

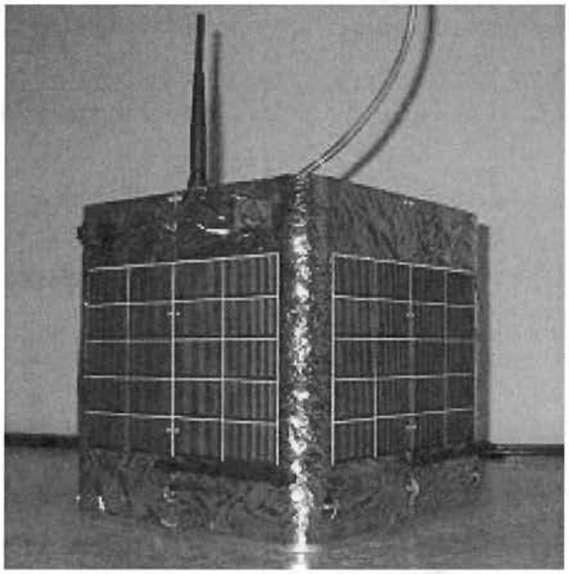


FIGURE 1.1 – Free-flyer testbed at the CSA

given by : (the coordinate frames are defined in [15]).

$$m_i\ddot{x}_i = F^c_{xi} + F^d_{xi} \tag{1.1a}$$

$$m_i\ddot{y}_i = F^c_{yi} + F^d_{yi} \tag{1.1b}$$

$$I_i\ddot{\alpha}_i = \tau^c_{zi} + \tau^d_{zi} \tag{1.1c}$$

where,

$m_i \in \mathbb{R}^n$: mass of the i^{th} satellite

$I_i \in \mathbb{R}^n$: moment of inertia of the i^{th} satellite about z-axis

$(x_i, y_i) \in \mathbb{R}^n$: location of the mass center of the i^{th} satellite in the inertial frame

$\alpha_i \in \mathbb{R}^n$: rotation angle of the i^{th} satellite about z-axis

$(F^d, \tau^d) \in \mathbb{R}^n$: disturbance forces and disturbance torque

$(F^c, \tau^c) \in \mathbb{R}^n$: control forces and control torque

Assumption 1 *We assume a symmetric mass distribution along z-axis.*

1.2.2 Neural Network Flow Control Valves Identification

This section describes thruster dynamics' learning identification strategy using neural networks for a free-flyer. Its effectiveness is investigated using experimental data of a single thruster. The free-flyer has eight flow control valves fed by an air distributor to produce thrust. Thus, air pressure drops when opening multiple valves at the same time such that only four valves are allowed to open simultaneously. Moreover, air distributor also provides air to the base, to form an air bearing as described in Fig 1.2.

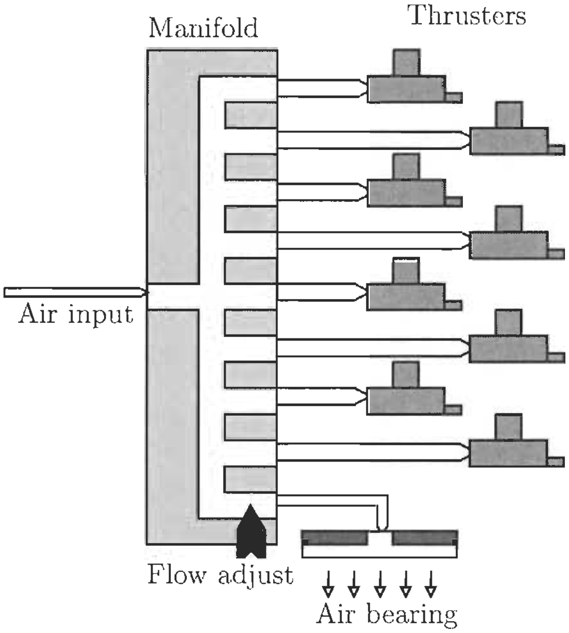


FIGURE 1.2 – Compressed air distribution diagram

The valves are installed on the base of the free-flyer to provide thrusts to move on $\pm x$ and $\pm y$ directions, as well as torques to rotate around $\pm z$ axis. A thrust calibration study of the flow control valves was conducted in [16]. In this study, a relationship between the digital inputs of the valves N_j and the produced thrust F_j was found through experiment for different air pressures ($P = 20, 40, 60, 80$ psi), $j = 1, \ldots, 8$.

For a given inlet air pressure P, the experimental measurements of the thrust F_j are taken for different incrementally increasing and decreasing input N_j to capture the hysteresis nature of the thrusters. It is worth pointing out that step inputs have similar output thrust as the incremental inputs. The experimental measurements are depicted in Fig 1.3 and thrusters diagram in Fig 1.4.

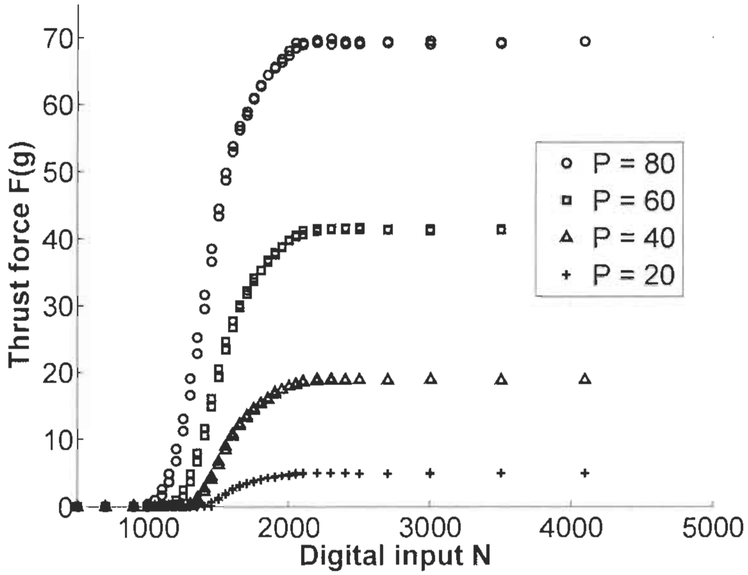


FIGURE 1.3 – Thruster experimental measurements for different air pressures

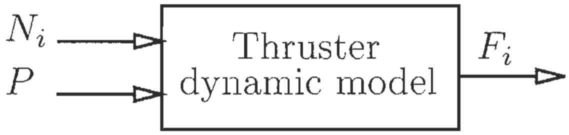


FIGURE 1.4 – Thrusters diagram of a free-flyer

Assumption 2 *We assume that all thrusters are identical.*

A feedforward multilayer perceptron (MLP) was used to learn thrusters' dynamics using a variable learning rate back-propagation algorithm [17]. Experimental data for ($P = 20, 60, 80$ psi) were presented to the network for training and experimental data for $P = 40$ psi were left for validation. The feedforward multilayer perceptron has two inputs, two hidden layers and one output. The first hidden layer has six nodes and the second has eight nodes. The activation functions used for the hidden and output layers are the sigmoid and linear functions, respectively.

1.2.3 Simulation and Experimental Validation

Fig. 1.5 shows the training results, the validation results are depicted in Fig. 1.6, and Fig 1.7 shows the MLP thrust force for $P \in [20, 80]$.

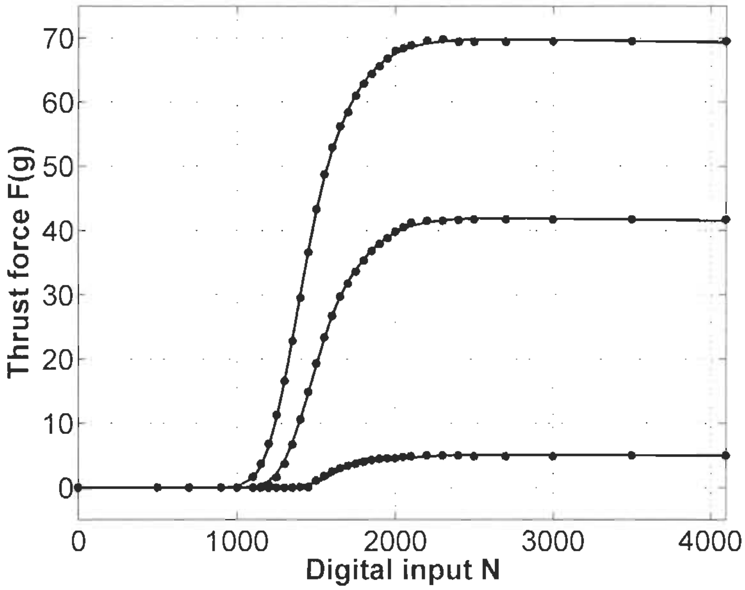


FIGURE 1.5 – MLP vs. experimental data in training phase

The following presents an ANN-based modeling strategy for a spacecraft formation flying. The ANN takes advantage of the merits of the soft-computing technique described in chapter 2 to approximate thrusters' dynamics using experimental data. The performance of the resultant strategy shows high approximation accuracy, which confirms the creditentials of such tools in providing a robust approximation for mathematically ill-defined systems. They are, therefore, powerful tools for compensating for even higher degrees of uncertainties. This fact motivates

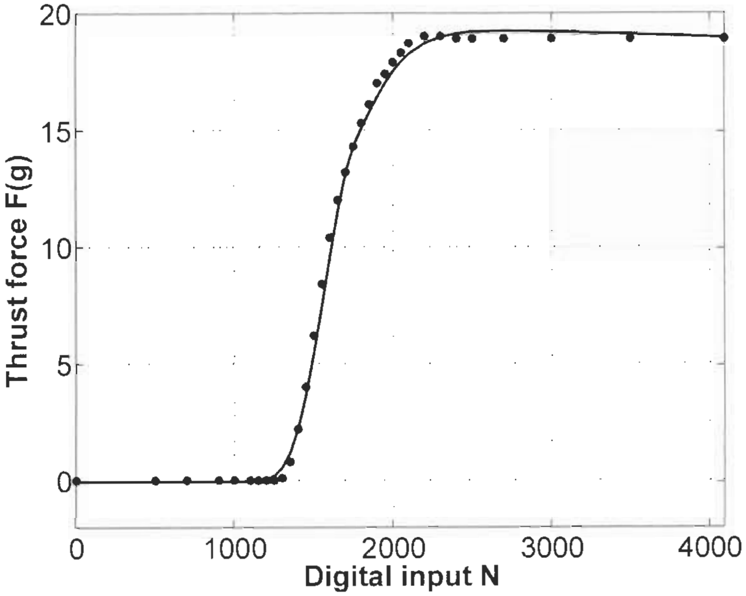


FIGURE 1.6 – MLP vs. experimental data for validation

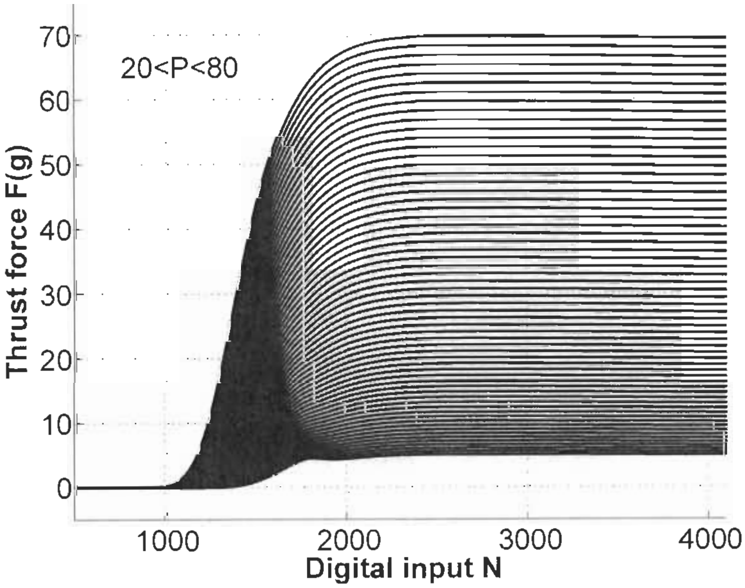


FIGURE 1.7 – Thruster model ($P \in [20, 80]$)

the design intelligent adaptive control techniques for complex dynamic systems. Since the stability of neural networks cannot be guaranteed with conventinal learning algorithms such as backpropogation, new Lyapunov-based adaptive learning mechanisms are proposed in this thesis.

1.3 Contributions

Nonlinear dynamic systems are governed by complex dynamics and hence are inevitably subject to the ubiquitous presence of high, particularly unstructured, modeling nonlinearities. The presence of such nonlinearities significantly changes the dynamics of nonlinear systems [14]. So, modeling a system's dynamics based on presumably accurate mathematical models cannot be applied efficiently in this case. This raises the urgency to consider alternative approaches for the control of this type of systems to keep up with their increasingly demanding design requirements. The major contributions of this thesis are the design of control structures, adaptation laws and stability proofs for adaptive soft-computing control strategies to achieve :

- Learning and approximation of *a priori* unknown nonlinear systems.
  - Thrusters dynamics identification for spacecraft formation flying missions.
  - High uncertainties approximation for inverted pendulums control problem.
  - Hysteresis compensation for piezoelectric actuators in Microelectromechanical systems (MEMS).
- Stability analysis for soft-computing techniques.
  - Lyapunov-based adaptive learning mechanism for neural networks.
  - Stable adaptation technique for fuzzy systems in the sense of Lyapunov.
- Lyapunov stable adaptive control strategies for electromechanical systems such as robotic manipulators.
  - Adaptive friction compensation to cope with its nonlinearities.
  - Disturbance attenuation to achieve robustness to unstructured uncertainties.
  - Elasticity effects' reduction to improve internal stability for flexible-joint manipula-

tors.
    - Parameters adaptation to obtain robustness to parametric uncertainties.
- High performance and robustness in PMSM drives.
    - Friction and load torque estimation for high tracking accuracy.
    - Adaptive control techniques for robustness to parametric uncertainties.
    - Soft-computing based approaches for high tolerance to unstructured uncertainties.
    - Speed estimation strategies.
- High efficiency for energy production systems.
    - Accurate state of charge estimation for battery management systems.
    - Soft-computing based adaptive control for power converters.
    - Adaptive DC bus voltage control to improve energy transfer efficiency.

1.4 Thesis Outline

The thesis is organized as follows. Chapters 2 to 4 present the background of artificial neural networks theory, fuzzy logic systems, neuro-fuzzy systems along with genetic and ant colony algorithms as soft-computing techniques, and nonlinear control systems. Chapter 5 presents Lyapunov-based adaptation strategies for neural networks and fuzzy systems. These techniques are applied to the well-known inverted pendulum problem. The ANN Lyapunov-based adaptation technique is also applied to piezoelectric actuators. In chapter 6, modeling of robotic manipulators with friction and joint elasticity is presented as well as many advanced control strategies with their results. In chapter 7, we present several speed control and estimation strategies for permanent magnet synchronous machines along with their stability proofs and results. Advanced adaptive techniques are presented in chapter 8 for energy production systems. We conclude in chapter 9 with some remarks and suggestions for further studies.

Chapitre 2

Artificial Neural Networks (ANNs)

2.1 Introduction

An artificial neural network (ANN) is a computational model that processes information using interconnected artificial neurons. As such, it mimics the structure and functional characteristics of biological neural networks of a brain in the way information is interpreted, the ability to learn, generalize and adapt to new situations. The ANN's inherent parallel nature increases information processing speed by distributing the calculations among many neurons. This chapter focuses on popular ANN models.

2.2 Learning Paradigms

An ANN, in its simplest form, has a set of inputs, connection weights, and outputs. The inputs are multiplied by the weight and are then summed before applying an activation function to the summed value. This activation level becomes the neuron's output and can be either an input for other neurons, or an output for the network. The learning is achieved by adjusting the weights to minimize the difference between the ANN and desired (training data) outputs. ANN can also be viewed as an adaptive system that changes its structure during the learning phase based on a set of training data. It can be used to model complex relationships between inputs and outputs or to find patterns in data.

There are two approaches to train a neural network, supervised and unsupervised learning. Supervised learning involves a mechanism, also called teacher, to provide the network with the desired output or the network's performance, such as a cost function. On the other hand, unsupervised learning does not have a function to be learned by the network but rather a continuous interaction with the environment.

2.2.1 Supervised Learning

In supervised learning, input-output training sets are provided to the network, which processes the information and computes the errors based on its resulting outputs. These errors are then propagated back through the network to adjust its weights. This process continues as the connection weights are ever refined until convergence. It is noteworthy that training sets with not enough information may not lead to convergence and too many layers and nodes make the network memorize the data instead of learning from it. The supervised learning scheme is depicted in figure. 2.1.

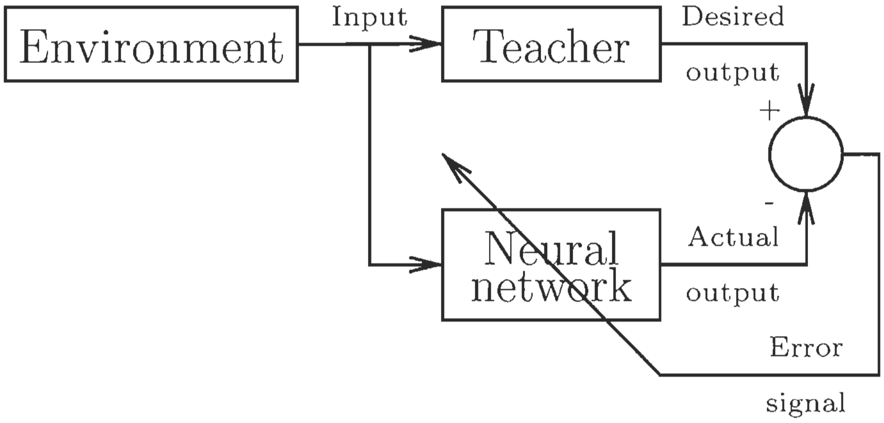


FIGURE 2.1 – Supervised learning scheme

2.2.2 Unsupervised Learning

In unsupervised learning, the network is provided with only input training sets. In real life, exact training sets do not exist for all situations. Hence, this type of learning aims to provide the network with the ability to adapt to the environment and learn from it, such as self-organizing map and reinforcement learning. The supervised learning scheme is depicted in figure. 2.2. As training data is not available, the network learns through a minimization of a cost function also called "critic".

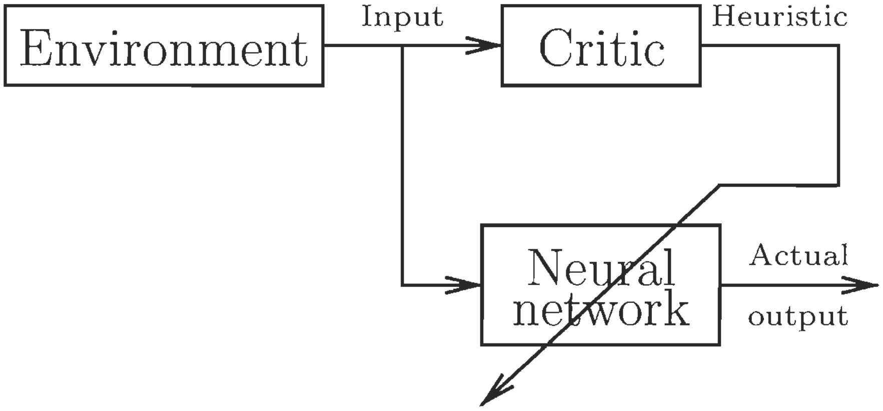


FIGURE 2.2 – Unsupervised learning scheme

2.3 Perceptron

The perceptron was first introduced by Rosenblatt in 1958 along with the mechanism to adjust its weights [18]. It is a very simple binary neural network with a single neuron. The perceptron uses supervised learning process to solve basic logical operations like AND or OR. However, it cannot solve more complicated logical operations, like the XOR problem. It is often used for pattern classification purposes. Figure. 2.3 shows a typical perceptron structure and its supervised training algorithm is described in algorithm 1.

Example : (OR problem)

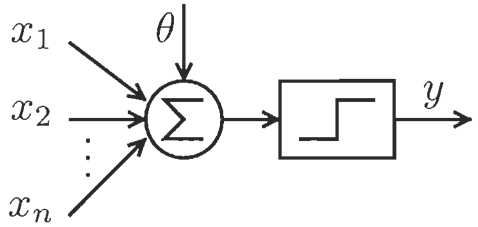


FIGURE 2.3 – Perceptron structure

Algorithm 1: Perceptron (Delta rule)

```
begin
    – Initialize the weight vector $w_i$ and the threshold $\theta$ to small random values.
    repeat
        repeat
            – Present an input vector $x(k) = [x_1, x_2, \dots, x_n]$ and a desired output $d(k)$ at instant $k$.
            – Compute the perceptron output $y(k) = f(\sum_{i=1}^{n}(x_i\, w_i) - \theta)$.
            $f(\diamond) = \begin{cases} 1, & \text{if } \diamond > \theta \\ 0, & \text{otherwise} \end{cases}$
            – Adapt the weight vector by the following :
            $e(k) = d(k) - y(k)$, and $\eta < 1$ is the learning rate.
            $w_i(k+1) = w_i(k) + \eta e(k) x_i(k)$
        until all training instances are finished.
    until convergence or satisfactory performance is reached.
end
```

TABLE 2.1 – OR problem

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

Consider a two-input perceptron to solve the OR problem in table 2.1.

Initialize the weights and the threshold to small random values :

$w_1 = 0.3$, $w_2 = 0.1$, and $\theta = 0.2$.

Set the learning rate to : $\eta = 0.15$.

Present the network with random patterns :

- First pattern : (pattern 3 = [1 0, 1])

  Compute the output $y(k) = f(\sum_{i=1}^{n}(x_i\ w_i) - \theta) = f(0.3 - 0.2) = f(0.1) = 0$.

  Compute the error $e(k) = d(k) - y(k) = 1 - 0 = 1$.

  Since $e(k) \neq 0$, adapt the weights :

  $w_1(k+1) = w_1(k) + \eta e(k) x_1(k) = 0.3 + 0.15 = 0.45$

  $w_2(k+1) = w_2(k) + \eta e(k) x_2(k) = 0.1 + 0 = 0.1$

- Second pattern : (pattern 2 = [0 1, 1])

  $y(k) = f(0.1 - 0.2) = f(-0.1) = 0$.

  $e(k) = 1 - 0 = 1$.

  $w_1(k+1) = 0.45 + 0 = 0.45$

  $w_2(k+1) = 0.1 + 0.15 = 0.25$

- Third pattern : (pattern 4 = [1 1, 1])

  $y(k) = f(0.45 + 0.25 - 0.2) = f(0.5) = 1$.

  $e(k) = 1 - 1 = 0$.

- Fourth pattern : (pattern 1 = [0 0, 0])

  $y(k) = f(-0.2) = f(-0.2) = 0$.

  $e(k) = 0 - 0 = 0$.

Recall, $y(k) = f(\sum_{i=1}^{n}(x_i\ w_i) - \theta) = (x_1\ w_1 + x_2\ w_2 - \theta)$. In this case, the separation between the two classes is a boundary in 2D space, i.e., a straight line, described by

$$x_1\ w_1 + x_2\ w_2 - \theta = 0$$

Therefore,

$$x_2 = -\frac{w_1}{w_2}x_1 + \frac{\theta}{w_2}$$

It is noteworthy that the slope of the line is determined by the weights and the threshold determines the offset. Thus, the perceptron represents a linear discriminant function. A geometrical representation before and after training is given in Fig. 2.4.

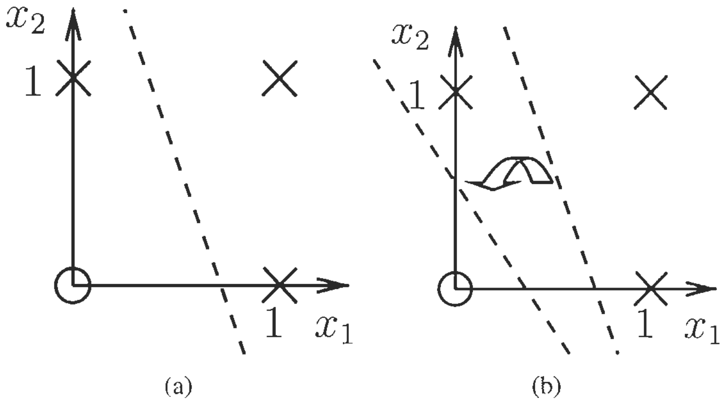


FIGURE 2.4 – Classification example : (a) before training ; and (b) after training.

2.4 Multilayer Perceptron (MLP)

The multi-layer-perceptron (MLP) was first introduced by Minsky and Papert in 1969 [18]. It is an extended version of the single layer perceptron as it may have one or more hidden layers. Whilst a perceptron forms a half-plane decision region, a MLP forms arbitrarily complex decision regions to separate various input patterns. Due to its extended structure, a MLP is able to solve every logical operation, including the XOR problem. Figure. 2.5 shows a typical MLP structure.

2.4.1 Back-propagation

Consider a single hidden layer neural network with n inputs, p hidden nodes, and m outputs as shown in Fig. 2.6.

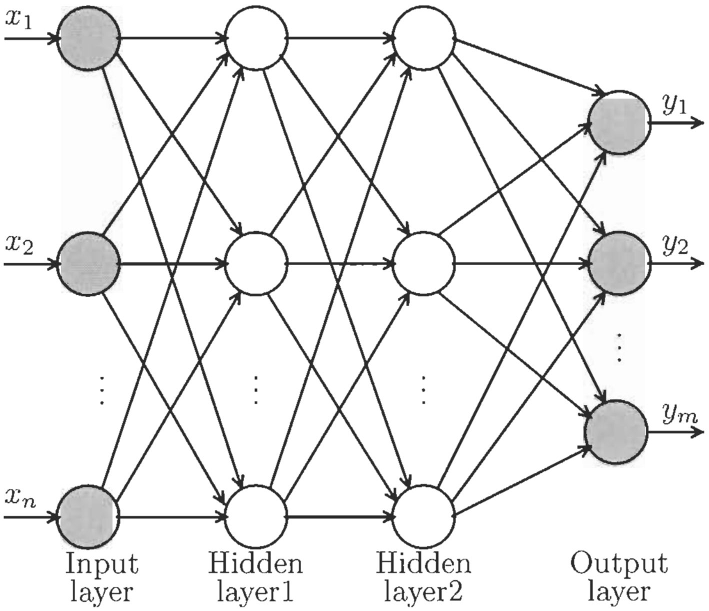


FIGURE 2.5 – Multilayer perceptron structure

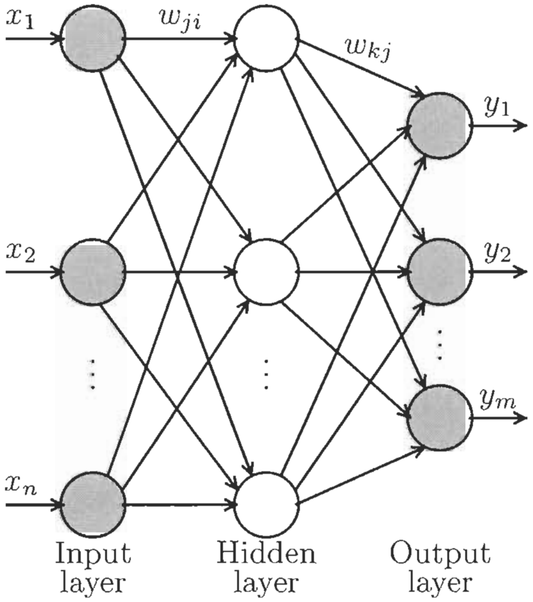


FIGURE 2.6 – Single hidden layer multilayer perceptron

We can describe the input-output relationship by

$$y_k(t) = \varphi_k(v_k(t)) \tag{2.1}$$

with,

$$v_k(t) = \sum_{j=1}^{p} w_{kj}(t)\, o_j(t) \tag{2.2}$$

$$o_j(t) = \varphi_j(u_j(t)) \tag{2.3}$$

$$u_j(t) = \sum_{i=1}^{n} w_{ji}(t)\, x_i(t) \tag{2.4}$$

where, $x_i(t)$ and $y_k(t)$ are the input and output vectors of the neural network at instant t, respectively.

The back-propagation is a supervised learning algorithm mainly used by MLPs to minimize a cost function that can be described as

$$\xi(t) = \frac{1}{2}\sum_{k=1}^{m} e_k^2(t) \tag{2.5}$$

with,

$$e_k(t) = d_k(t) - y_k(t) \tag{2.6}$$

where, $e_k(t)$ is the neural network's error at instant t and $d_k(t)$ is the desired neural network's output at instant t.

The back-propagation algorithm adapts the weights using a gradient descent technique, where the gradient is computed by

$$\frac{\partial \xi(t)}{\partial w_{kj}(t)} = \frac{\partial \xi(t)}{\partial e_k(t)} \frac{\partial e_k(t)}{\partial y_k(t)} \frac{\partial y_k(t)}{\partial v_k(t)} \frac{\partial v_k(t)}{\partial w_{kj}(t)} \tag{2.7}$$

$$\frac{\partial \xi(t)}{\partial w_{ji}(t)} = \frac{\partial \xi(t)}{\partial o_j(t)} \frac{\partial o_j(t)}{\partial u_j(t)} \frac{\partial u_j(t)}{\partial w_{ji}(t)} \tag{2.8}$$

Differentiating (2.5) with respect to $e_k(t)$ yields,

$$\frac{\partial \xi(t)}{\partial e_k(t)} = e_k(t) \tag{2.9}$$

Differentiating (2.6) with respect to $y_k(t)$ yields,

$$\frac{\partial e_k(t)}{\partial y_k(t)} = -1 \tag{2.10}$$

Differentiating (2.1) with respect to $v_k(t)$ yields,

$$\frac{\partial y_k(t)}{\partial v_k(t)} = \varphi_k'(v_k(t)) \tag{2.11}$$

Differentiating (2.2) with respect to $w_{kj}(t)$ yields,

$$\frac{\partial v_k(t)}{\partial w_{kj}(t)} = o_j(t) \tag{2.12}$$

Differentiating (2.5) with respect to $o_j(t)$ yields,

$$\frac{\partial \xi(t)}{\partial o_j(t)} = \sum_{k=1}^{m} e_k(t)\frac{\partial e_k(t)}{\partial o_j(t)} = \sum_{k=1}^{m} e_k(t)\frac{\partial e_k(t)}{\partial v_k(t)}\frac{\partial v_k(t)}{\partial o_j(t)} \tag{2.13}$$

Substitute (2.1) in (2.6) yields,

$$e_k(t) = d_k(t) - \varphi_k(v_k(t))$$

Therefore,

$$\frac{\partial e_k(t)}{\partial v_k(t)} = -\varphi_k'(v_k(t)) \tag{2.14}$$

Differentiating (2.2) with respect to $o_j(t)$ yields,

$$\frac{\partial v_k(t)}{\partial o_j(t)} = w_{kj}(t) \tag{2.15}$$

Differentiating (2.3) with respect to $u_j(t)$ yields,

$$\frac{\partial o_j(t)}{\partial u_j(t)} = \varphi_j'(u_j(t)) \tag{2.16}$$

Differentiating (2.4) with respect to $w_{ji}(t)$ yields,

$$\frac{\partial u_j(t)}{\partial w_{ji}(t)} = x_i(t) \tag{2.17}$$

Substitute (2.9), (2.10), (2.11), and (2.12) in (2.7) and (2.13), (2.16), and (2.17) in (2.8) yields,

$$\begin{aligned}\frac{\partial \xi(t)}{\partial w_{kj}(t)} &= -e_k(t)\varphi_k'(v_k(t))o_j(t) = -\delta_k(t)o_j(t)\\ \frac{\partial \xi(t)}{\partial w_{ji}(t)} &= -\sum_{k=1}^{m} e_k(t)\varphi_k'(v_k(t))w_{kj}(t)\varphi_j'(u_j(t))x_i(t) = -\sum_{k=1}^{m} \delta_k(t)w_{kj}(t)\varphi_j'(u_j(t))x_i(t) = -\delta_j(t)x_i(t)\end{aligned}$$

where, $\delta_j(t)$ and $\delta_k(t)$ are the local gradient for the hidden and the output neurons, respectively. The weights correction are defined by the delta rule

$$\begin{aligned}\Delta w_{kj}(t) &= \eta\frac{\partial \xi(t)}{\partial w_{kj}(t)} = -\eta\ \delta_k(t)\ o_j(t)\\ \Delta w_{ji}(t) &= \eta\frac{\partial \xi(t)}{\partial w_{ji}(t)} = -\eta\ \delta_j(t)\ x_i(t)\end{aligned}$$

where, η is the learning rate. Therefore, the neural network weights are computed as follows,

$$\begin{aligned}w_{kj}(t+1) &= w_{kj}(t) + \Delta w_{kj}(t)\\ w_{ji}(t+1) &= w_{ji}(t) + \Delta w_{ji}(t)\end{aligned}$$

2.4.2 Adaptive Learning Rate

The standard backpropagation algorithm using a constant learning rate cannot handle all the error surfaces. In other words, an optimal learning rate for a given synaptic weight is not necessarily optimal for the rest of the network's synaptic weights. Henceforth, every adjustable network parameter of the cost function should have its own individual learning rate parameter. Moreover every learning rate parameter should be allowed to vary at each iteration because the error surface typically behaves differently along different regions of a single-weight dimension [18]. The current operating point in the weight space may lie on a relatively flat portion of the error surface along a particular weight dimension. In such a situation where the derivative of the cost function with respect to that weight maintains the same algebraic sign, which means pointing in the same direction, for several consecutive iterations of the algorithm, the learning rate parameter for that particular weight should be increased. When the current operating point in the weight space lies on a portion of the error surface along a weight dimension of interest that exhibits peaks and valleys (i.e., the surface is highly irregular), then it is possible for the derivative of the cost function with respect to that weight to change its algebraic sign from one iteration to the next. To prevent the weight adjustment from oscillating, the learning rate parameter for that particular weight should be decreased appropriately when the algebraic sign of the derivative of the cost function with respect to a particular synaptic weight alternates for several consecutive iterations of the algorithm [18]. It is noteworthy that the use of a different time-varying learning rate parameter for each synaptic weight in accordance to this approach modifies the standard backpropagation algorithm in a fundamental way.

2.5 Kohonen Self-Organizing Map

The Kohonen self-organizing map, also called the Kohonen feature map, was first introduced by Teuvo Kohonen in 1982 [18]. It is one of the most used neural network models for data analysis. It can also be applied to other tasks where neural networks have been tried successfully. A self-organizing map (Fig. 2.7) consists of two layers, an input layer and an output layer, called a feature map, which represent the output vectors of the output space. The weights of the connections of an output neuron j to all the n input neurons, form a vector w_j in an n-dimensional space. The input values may be continuous or discrete, but the output values are binary only.

The output neurons learn during the training phase to react to inputs that belong to some clusters to represent typical features. This characteristic is inspired from the fact that the brain is organized into regions that correspond to different sensory stimuli. Therefore, a self-organizing map is able to extract abstract information from multidimensional array and represent it as a location, in one, or more dimensional space.

In the training phase, the output layer neurons are competitive. A neuron has a strong excitatory connection to itself and the excitatory level decreases within a certain radius when moving away to its neighboring neurons. Beyond this radius, a neuron either inhibits the activation of the other neurons or does not influence them. This scheme is called the "winner-takes-all", where a competition between neurons yields only one winner neuron that represents the class or the feature to which the input vector belongs. The self-organizing map structure is depicted in Fig. 2.7.

The unsupervised training algorithm for self-organizing map is described in algorithm 2 :

Algorithm 2: Self-organizing map unsupervised training

```
begin
    – Initialize the weight vector $w_j$ to small random values.
    repeat
        repeat
            – Present an input vector $x$ at time $t$.
            – Compute the distance $d_j(t) = \sum(x(t) - w_j(t))^2$.
            – Declare the neuron with the smallest distance as a winner.
            – Change the weight vector within the neighbourhood area $R$ :
              $w_j(t+1) = \begin{cases} w_j(t) + \eta(x(t) - w_j(t)), & if j \in R \\ w_j(t), & otherwise. \end{cases}$
              $\eta$ is the learning rate.
            – Decrease $\eta$ and $R$ in time.
        until all training instances are finished.
    until convergence or satisfactory performance is reached.
end
```

Example : (classification problem)

Consider the following three vectors to be clustered into 2 classes : (0,1,0) (0,0,1) (1,0,0).

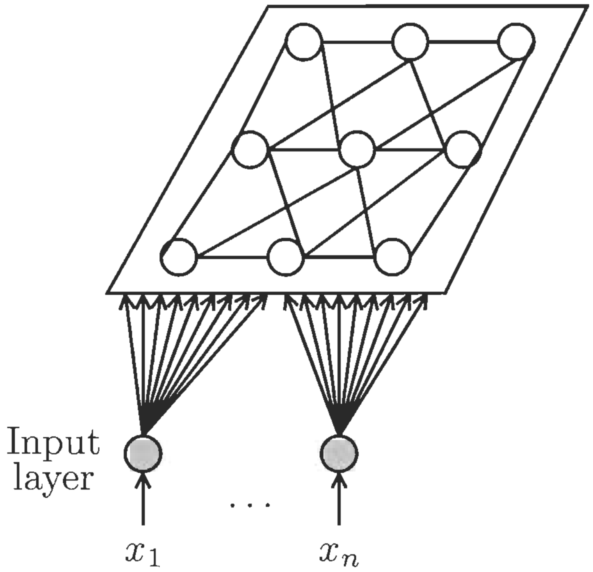


FIGURE 2.7 – Self-organizing map structure

Initialize the weights to small random values :

$$w = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.3 \\ 0.4 & 0.2 \end{bmatrix}$$

Set the radius as : $R = 0$.

Set the learning rate as : $\eta(0) = 0.4$ and $\eta(t+1) = 0.8\eta(t)$

Present the network with input vectors :

- First pattern : (0,1,0)

  Compute the distance :

  $d_1 = (0.2-0)^2 + (0.1-1)^2 + (0.4-0)^2 = 1.01$

  $d_2 = (0.1-0)^2 + (0.3-1)^2 + (0.2-0)^2 = 0.54$

  Adapt the weights of the winner node (2) :

  $w_2(k+1) = w_2(k) + \eta(x_i - w_{i2}(k)) = [0.06\ 0.58\ 0.12]^T$

- Second pattern : (0,0,1)

  Compute the distance :

$d_1 = (0.2-0)^2 + (0.1-0)^2 + (0.4-1)^2 = 0.41$

$d_2 = (0.06-0)^2 + (0.58-0)^2 + (0.12-1)^2 = 1.11$

Adapt the weights of the winner node (1) :

$w_1(k+1) = w_1(k) + \eta(x_i - w_{i1}(k)) = [0.12\ 0.06\ 0.64]^T$

- Third pattern : (1,0,0)

  Compute the distance :

  $d_1 = (0.12-1)^2 + (0.06-0)^2 + (0.64-0)^2 = 1.18$

  $d_2 = (0.06-1)^2 + (0.58-0)^2 + (0.12-0)^2 = 1.23$

  Adapt the weights of the winner node (1) :

  $w_1(k+1) = w_1(k) + \eta(x_i - w_{i1}(k)) = [0.47\ 0.03\ 0.38]^T$

- Reduce the learning rate : $\eta(t+1) = 0.8\eta(t) = 0.32$

- The new weight matrix :

$$w = \begin{bmatrix} 0.47 & 0.06 \\ 0.03 & 0.58 \\ 0.38 & 0.12 \end{bmatrix}$$

- Go for another iteration.

After many iterations, the weight matrix converges to the following :

$$w = \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \\ 0.5 & 0 \end{bmatrix}$$

The first cluster converges to the average of the two input vectors $(0,0,1)$ $(1,0,0)$, while third input vector $(0,1,0)$ is classified in the second cluster.

2.6 Hopfield Networks

Hopfield network was first introduced by John Hopfield in 1982 [18]. It is a fully connected feedback network, where each neuron is connected to all other neurons and there is no differentiation between input and output neurons. The Hopfield network structure is depicted in Fig. 2.8. There are variants of realizations of a Hopfield network. Every neuron j, $j = 1, 2, \ldots, n$ in the network is connected back to every other one, except itself. Input patterns x_j are supplied to the external inputs I_j and cause activation of the external outputs. The response of such a network, when an input vector is supplied during the recall procedure, is dynamic, that is, after supplying the new input pattern, the network calculates the outputs and then feeds them back to the neurons ; new output values are then calculated, and so on, until equilibrium is reached. Equilibrium is considered to be the state of the system when the output signals do not change for two consecutive cycles, or change within a small constant. The weights in a Hopfield network are symmetrical for reasons of stability in reaching equilibrium, that is, $w_{ij} = w_{ji}$. The network is of an additive type, that is,

$$u_j = \sum_i (w_{ij}\, o_i) + I_j, \qquad i \neq j$$

$$o_i = \begin{cases} 1, & \text{if } u_j > \theta_j \\ 0 \text{ or } -1, & \text{otherwise} \end{cases}$$

where, θ_j is a threshold for the j^{th} neuron.

A dynamical energy function E of a Hopfield network is defined at a moment t as

$$E(t) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (w_{ij}(t)\, o_i(t)\, o_j(t)), \qquad i \neq j$$

This function is considered as a Lyapunov function since it ensures the convergence of the network to local minima states when the weights of a Hopfield network are computed by

$$w_{ij} = \sum_{p=1}^{m} (2x_i^p - 1)(2x_j^p - 1)$$

where, x_i^p is the i^{th} binary value of the input pattern p. Thus, a local minima state in the energy function is a stable state for the network.

The weights update might be done according to the following modes :

- Asynchronous updating : Each neuron changes its state at a random moment with respect to other neurons.
- Synchronous updating : All neurons change their state simultaneously at a given time.
- Sequential updating : All neurons change their state sequentially, i.e., only one neuron changes its state at a given time.

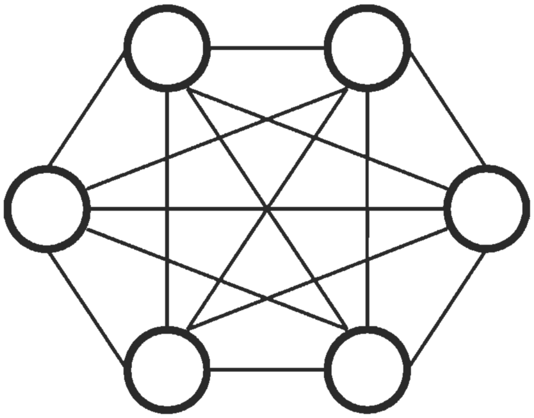


FIGURE 2.8 – Hopfield network structure

2.7 Boltzmann Machine

The Boltzmann machine was developed by Hinton and Sejnowski in 1983 to overcome the local minima problem [18]. It uses a variable called temperature to calculate the activation value of a neuron as a statistical probability

$$o_i = 1,\ with\ a\ probability\ p_i = 1/(1+e^{-u_iT})$$
$$o_i = 0,\ with\ a\ probability\ (1-p_i)$$

where u_i is the network's input to the i^{th} neuron, calculated as in the Hopfield network. This model is called a Boltzmann machine because of its similarity with the process of annealing in metallurgy, where temperature T decreases during the recall process. By changing the temperature, we shake the neural network to enable a jump from a local minima and a move towards a global minimum attractor.

2.8 Radial Basis Function (RBF)

Radial basis functions were first introduced by Powell in 1989 [18]. RBF network is an excellent approximator used for curve fitting problems and its learning is equivalent to finding a multidimensional function that provides a best fit to the training data. Moreover, it can be trained easily and quickly. The main advantage of RBF network is that the output layer weights are adjusted through linear optimization, whereas the hidden layer weights are adjusted through a nonlinear optimization. As depicted in Fig. 2.9, the RBF network output can be expressed using the Gaussian function and the weighted sum method,

$$y_k = \sum_{j=1}^{l} w_{kj}\ \phi_j, \quad k = 1,2,\ldots,m$$
$$\phi_j = exp\Big(-\frac{||x_i - c_j||^2}{\sigma_j^2}\Big), \quad i = 1,2,\ldots,n$$

where n, l, and m are the number of input, hidden, and output nodes, respectively, x_i is the i^{th} input vector of the training set and w_{kj} is the weight connecting the j^{th} hidden node to the k^{th} output node, ϕ_j, c_j, and σ_j are the j^{th} Gaussian function, center, and standard deviation, respectively. RBF network usually exhibits a slow response due to the large number of neurons needed in the second layer. However, appropriate RBF centers and widths selection can improve the accuracy and speed of the network. Hence more efficient RBF networks are built by the redistribution of centers to locations where input training data are meaningful.

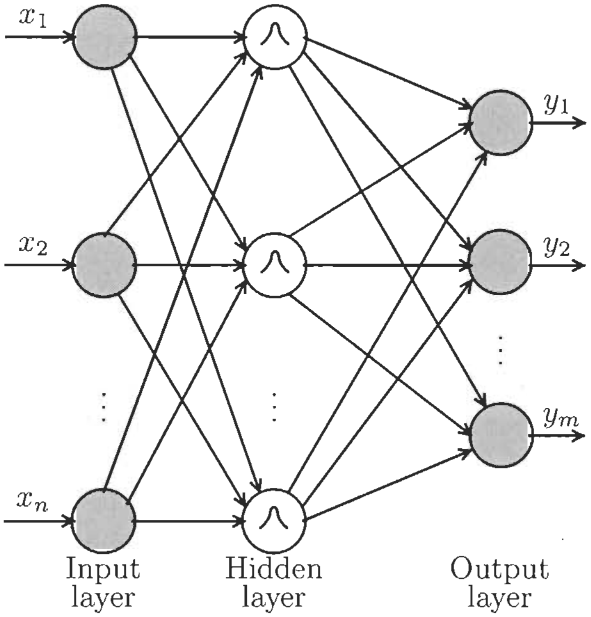


FIGURE 2.9 – RBF network structure

There are three major approaches to determine the centers. In the first approach, the locations of the centers are fixed and chosen randomly from the training data. This method can be useful if the training data are distributed in a representative manner for the specified problem. In the second approach, the radial basis functions can move the locations of their centers in a self-organized fashion. Hence, the centers of radial basis functions are placed in the regions of the input space where important data exist. In the third approach, the centers of the radial basis functions are obtained by a supervised learning process. It defines a cost function to be minimized using an error-correction learning scheme such as a gradient-descent procedure.

2.9 Conclusion

ANNs have received a thorough interest from many researchers as they are able to learn a system's behavior from its input-output data. These learning and generalization capabilities enable ANNs to more effectively address nonlinear time-varying complex problems. Thus far, MLP remains one the most popular ANNs used in control applications with backpropagation algorithm for weights adaptation. However, this algorithm is based on gradient or steepest descent methods. These methods have a serious drawback as they are not suitable for all types of surfaces. Therefore, the convergence and stability of such approaches cannot be guaranteed. In this thesis, the MLP is used for its simplicity and its stability issues are addressed by proposing new Lyapunov-based adaptation techniques as an alternative to classic gradient-based methods. On the other hand, fuzzy logic systems are also good candidates for nonlinear dynamic systems. They have been successfully used in many real-world applications. Therefore, the next chapter presents an overview of the fuzzy logic theory.

Chapitre 3

Fuzzy Logic Systems (FLSs)

3.1 Introduction

Fuzzy logic was first developed by Lotfi Zadeh in the 1960s [19]. It aims to represent uncertain and imprecise knowledge and provides an approximate but effective means of describing the behavior of systems that are too complex or ill-defined. Fuzzy logic uses graded statements rather than the strictly true or false logic. It attempts to incorporate the human-like knowledge and reasoning used for decision making. Thus, fuzzy logic provides an approximate but effective way of describing the behavior of complex systems that are not easy to describe precisely. This results in controllers that are capable of making intelligent control decisions. This chapter presents the conventional fuzzy logic systems, called type-1 FLSs, and its type-2 counterpart as an extension to the conventional fuzzy logic theory.

3.2 Type-1 Fuzzy Logic Systems

In its simplest form, an n-input m-output type-1 fuzzy logic system (FLS) 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 $j = 1, 2, \ldots, m$. The output $y = (y_1, \ldots, y_m)^T$ of an n-input m-output FLS with a center-average defuzzifier, sum-product inference, and singleton output fuzzifier, is given by

$$y_j = \frac{\sum_{p=1}^{R} \bar{y}_j^{(p)} \left(\prod_{i=1}^{n} \mu_{A_i^{(p)}}(x_i) \right)}{\sum_{p=1}^{R} \left(\prod_{i=1}^{n} \mu_{A_i^{(p)}}(x_i) \right)}, \quad j = 1, \ldots, m \tag{3.1}$$

where $x = (x_1, \ldots, x_n)^T \in U$ is the FLS's input vector, $\mu_{A_i^{(p)}}$ are the membership functions of the fuzzy sets $A_i^{(p)}$, $\prod$ and $\sum$ denote the fuzzy t-norm and t-conorm operations used, respectively, p is the rule index from a total of R rules, and $\bar{y}_j^{(p)}$ is the point in V_j at which the firing strength $\mu_{B_j^{(p)}}$ achieves its maximum value, which is assumed to be 1.

The type-1 FLS is capable of uniformly approximating any well-defined nonlinear function over a compact set U to any degree of accuracy.

Theorem 1 *(universal approximation theorem [10]) For any given real continuous function $g(x)$ on the compact set $U \subset \mathbb{R}^n$ and arbitrary $\varepsilon > 0$, there exists a function $f(x)$ in the form of (3.1) such that*

$$\sup_{x \in U} \parallel g(x) - f(x) \parallel < \varepsilon$$

The above universal approximation theorem [10] shows the power of fuzzy logic systems in approximating continuous nonlinear functions. As such, FLSs provide a natural alternative for tackling problems usually raised when attempting to model and design controllers for complex systems with ill-defined dynamics.

A block diagram of a typical type-1 FLS is depicted in Fig. 3.1. It contains four components : fuzzifier, rule base, inference engine, and defuzzifier.

3.2.1 Fuzzification

Type-1 fuzzy sets can be represented as membership functions μ_A that associate with each element x of the universe of discourse X, a number $\mu_A(x)$, i.e. membership grade, in the interval [0,1]. The fuzzifier maps a crisp input $x \in X$ into a fuzzified value $A \in U$ (Universe).

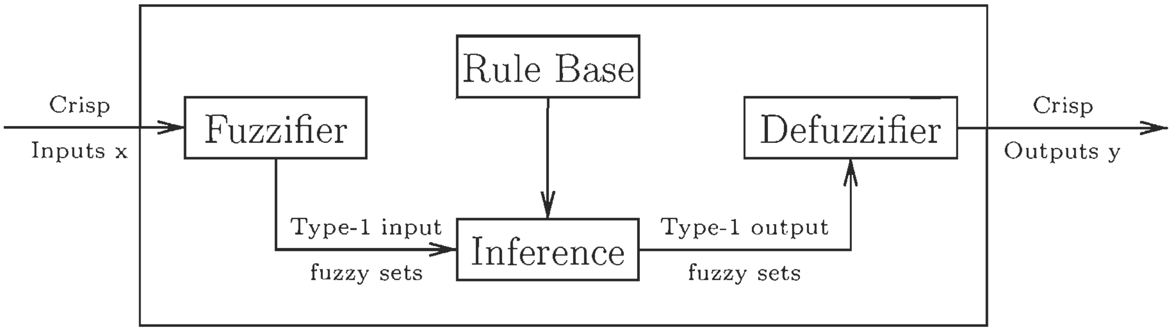


FIGURE 3.1 – Block diagram of a type-1 FLS

1. Singleton fuzzification : fuzzy set A with support x_i, where $\mu_A(x_i) = 1$, for $x = x_i$ and $\mu_A(x_i) = 0$, for $x \neq x_i$.
2. Non-singleton fuzzification : $\mu_A(x_i) = 1$, for $x = x_i$ and decreases from 1 to 0 when moving away from $x = x_i$.

3.2.2 Fuzzy Rule Base

Type-1 fuzzy rule base is a collection of multiple IF-THEN rules. The p^{th} rule has the following form :

$$R^p : \text{IF } x_1 \text{ is } F_1^p \text{ and } x_2 \text{ is } F_2^p \text{ and } \ldots \text{ and } x_n \text{ is } F_n^p$$
$$\text{THEN } y_1 \text{ is } G_1^p \text{ and } y_2 \text{ is } G_2^p \text{ and } \ldots \text{ and } y_m \text{ is } G_m^p$$

where $x_1 \in X_1, \ldots, x_n \in X_n$, and $y_1 \in Y_1, \ldots, y_m \in Y_m$, are the inputs and outputs linguistic variables, respectively. F_i^p and G_j^p, $i = 1, \ldots, n, j = 1, \ldots, m$, are input and output fuzzy labels, respectively. R^p is a fuzzy relation from fuzzy input sets X to fuzzy output sets Y.

3.2.3 Fuzzy Inference Engine

The inference engine is a key component of any FLS. It aggregates the IF-THEN rules stored in the knowledge base with the type-1 fuzzy sets generated by the fuzzifier to form an overall type-1 output fuzzy set. It provides a mapping from the input fuzzy sets to the output ones. The intersection of multiple rule antecedents is computed using a t-norm operator $\sqcap$

while the union of multiple rules is computed through a t-conorm operation $\sqcup$. Each rule p in the knowledge base is interpreted as a fuzzy implication, which, when aggregated with the fuzzified inputs, infers a fuzzy set B^p such that :

$$\mu_{B^p}(y) = \sqcup_{x \in X}[\mu_{A_x}(x) \sqcap \mu_{R^p}(x,y)].$$

3.2.4 Defuzzification

Type-1 defuzzification is the last step to get the final output crisp value. There are several methods for the type-1 defuzzification of a FLS, such as the centroid defuzzifier, centre-of-sums defuzzifier, height defuzzifier, and centre-of-sets defuzzifier. The output at instant k with a commonly used defuzzification method, such as, the centroid method is expressed as :

$$y(k) = \frac{\sum_{p=1}^{R} y_p \, \mu_{B^p}(y)}{\sum_{p=1}^{R} \mu_{B^p}(y)} \tag{3.2}$$

3.3 Type-2 Fuzzy Logic Systems

A type-2 fuzzy set is characterized by a fuzzy membership function, where the membership value or grade for each element of this set is a fuzzy set in the interval $[0,1]$; unlike a type-1 fuzzy set where the membership grade is a crisp value. As such, the membership functions of type-2 fuzzy sets are three dimensional functions, with what is known as the set's footprint of uncertainty (FOU) representing the third dimension. In fact, it is this FOU that provides type-2 FLSs with additional degrees of freedom and makes it possible for them to directly model and handle more types of uncertainties with higher magnitudes than their type-1 counterparts. Moreover, using type-2 fuzzy sets to represent a certain system's inputs and outputs can result in a smaller rule base as when a type-1 FLS was used. A block diagram of a typical type-2 FLS is depicted in Fig. 3.2. It is generally composed of five components : a fuzzifier, a rule base, a fuzzy inference engine, a type-reducer and a defuzzifier. In essence, it has a very similar structure to a type-1 FLS.

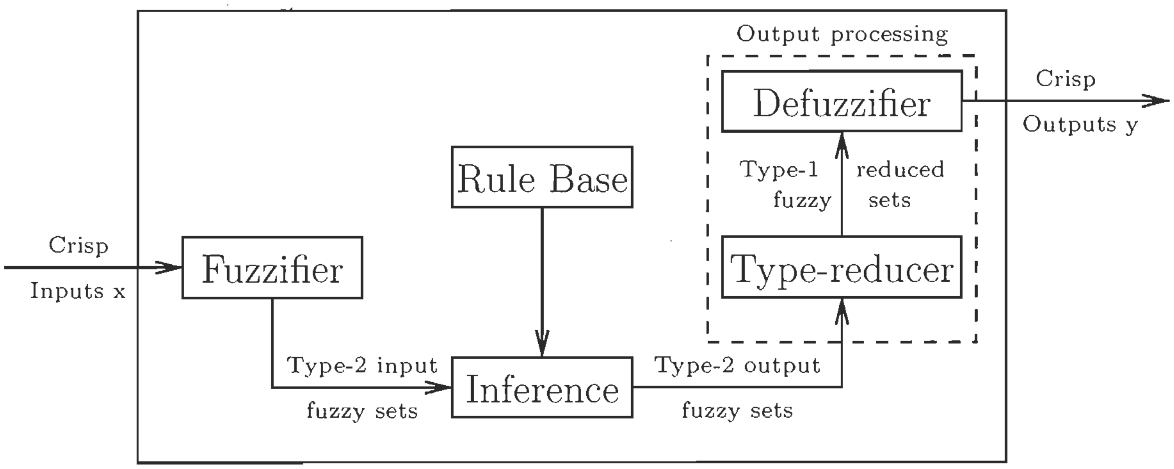


FIGURE 3.2 – Block diagram of a type-2 FLS

Type-2 fuzzy sets are generalized forms of those of type 1 (with the FOU as an additional degree of freedom). Mathematically, a type-2 fuzzy set, denoted as $\tilde{A}$, is characterized by a type-2 membership function $\mu_{\tilde{A}}(x,u)$, where $x \in X$ and $u \in J_x \subseteq [0,1]$, i.e.,

$$\tilde{A} = \{((x,u), \mu_{\tilde{A}}(x,u) \mid \quad \forall x \in X, \quad \forall u \in J_x \subseteq [0,1]\}$$

in which $0 \leq \mu_{\tilde{A}}(x,u) \leq 1$. For a continuous universe of discourse, $\tilde{A}$ can be expressed as

$$\tilde{A} = \int_{x\in X} \int_{u\in J_x} \mu_{\tilde{A}}(x,u)/(x,u) \quad J_x \subseteq [0,1]\}$$

where J_x is referred to as the primary membership of x.

As in type-1 fuzzy logic, discrete fuzzy sets are represented by the symbol $\sum$ instead of $\int$. The secondary membership function associated to $x = x'$, for a given $x' \in X$, is the type-1 membership function defined by $\mu_{\tilde{A}}(x = x', u)$, $\forall u \in J_x$. The uncertainty in the primary membership of a type-2 fuzzy set $\tilde{A}$ is represented by the FOU and is illustrated in Fig. 3.3. Note that the FOU is also the union of all primary memberships.

$$FOU(\tilde{A}) = \bigcup_{x\in X} J_x$$

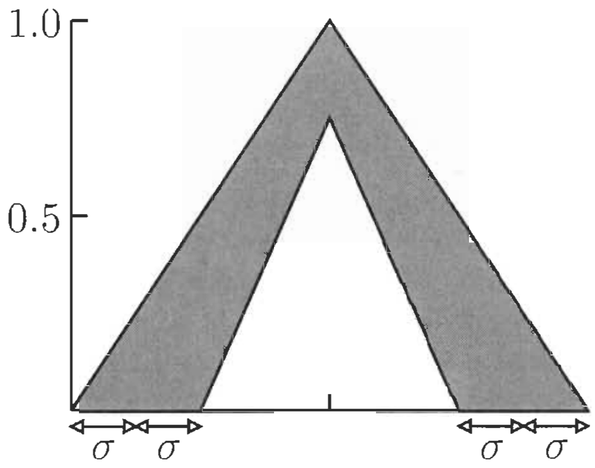


FIGURE 3.3 – Type-2 Fuzzy Logic Membership Function

The upper and lower membership functions, denoted by $\overline{\mu}_{\tilde{A}}(x)$ and $\underline{\mu}_{\tilde{A}}(x)$, respectively, are two type-1 membership functions that represent the upper and lower bounds for the footprint of uncertainty of an interval type-2 membership function $\mu_{\tilde{A}}(x,u)$, respectively [8].

3.4 Interval Type-2 Fuzzy Logic Systems

The operations of type-2 fuzzy systems are typically more computationally involved than type-1 systems. This has urged researchers to search for ways to alleviate this high computational burden if type-2 FLSs are to find their way to real-world applications. For this purpose, interval fuzzy sets were introduced [8, 19]. This type of fuzzy sets provides a simplified and efficient alternative to easily compute the input and antecedent operations for FLSs and offers a balanced trade off between performance and complexity. Before proceeding further, a few concepts need to be introduced.

Definition 1 *[19] An interval type-1 fuzzy set A on a universe of discourse X is defined by a binary type-1 membership function μ_A and a left and right points in X, say l and r, such that*

$$\mu_A(x) = \begin{cases} 1, \ \textit{if}\, x \in [l,r] \\ 0, \ \textit{otherwise} \end{cases}$$

In such a case, the membership function can be denoted by $\mu_A = [l, r]$.

Definition 2 *[19] A type-2 membership function, defined on a universe X, whose secondary membership grade at some $x' \in X$ is 1, and 0 elsewhere, is called a singleton type-2 membership function.*

An FLS with an interval singleton type-2 fuzzifier and product or minimum t-norm satisfies the following properties [19] :

1. the firing strength of the lth fuzzy rule is an interval type-1 fuzzy set defined as

$$F^p(x') \equiv \sqcap_{i=1}^{n} \mu_{\tilde{F}_i^p}(x_i') = [\underline{f}^p(x'), \overline{f}^p(x')] \equiv [\underline{f}^p, \overline{f}^p],$$

   where

$$\underline{f}^p(x') = \underline{\mu}_{\tilde{F}_1^p}(x_1') \star \cdots \star \underline{\mu}_{\tilde{F}_n^p}(x_n') \tag{3.3}$$

$$\overline{f}^p(x') = \overline{\mu}_{\tilde{F}_1^p}(x_1') \star \cdots \star \overline{\mu}_{\tilde{F}_n^p}(x_n') \tag{3.4}$$

   with the t-norm operator denoted by '$\star$'.

2. the fired output consequent set of the p rule is a type-1 fuzzy set characterized by a membership function

$$\mu_{\tilde{B}^p}(y) = \int_{b^p \in [\underline{f}^p \star \underline{\mu}_{\tilde{G}^p}(y), \overline{f}^p \star \overline{\mu}_{\tilde{G}^p}(y)]} 1/b^p \qquad \forall y \in Y$$

   with $\underline{\mu}_{\tilde{G}^p}(y)$ and $\overline{\mu}_{\tilde{G}^p}(y)$ being the lower and upper membership grades of $\mu_{\tilde{G}^p}(y)$.

3. if N out of a total of R fuzzy rules in the FLS fire, where $N \leq R$, then the overall aggregated output fuzzy set is defined by a type-1 membership function $\mu_{\tilde{B}}(y)$ obtained by combining the fired output consequent sets into one. In other words, $\mu_{\tilde{B}}(y) = \sqcup_{p=1}^{N} \mu_{\tilde{B}^p}(y)$,

where

$$\mu_{\tilde{B}^p}(y) = \int_{b \in \left[[\underline{f}^1 \star \underline{\mu}_{\tilde{G}^1}(y)] \vee \cdots \vee [\underline{f}^N \star \underline{\mu}_{\tilde{G}^N}(y)], [\overline{f}^1 \star \overline{\mu}_{\tilde{G}^1}(y)] \vee \cdots \vee [\overline{f}^N \star \overline{\mu}_{\tilde{G}^N}(y)]\right]} 1/b^p, \quad \forall y \in Y \quad (3.5)$$

The following is a brief description of the different stages of the type-2 fuzzy logic inference engine.

3.4.1 Fuzzification

Fuzzification is the first stage of the fuzzy inferencing process. At this phase, the crisp input vector with n elements $x = (x_1, \ldots, x_n)^T$ in the universe of discourse $X_1 \times X_2 \times \ldots \times X_n$ is mapped into type-2 fuzzy sets [8, 19]. The type-2 fuzzification process is schematically depicted in Fig. 3.4. For each point of the universe of discourse, the upper and lower membership functions are computed. It is worth pointing out that the acquisition of the inputs as well as their primary membership grades can be performed in parallel and regardless of the t-norm operator used. For rule p, the result of this operation is an interval type-1 set $[\underline{f}^p, \overline{f}^p]$.

3.4.2 Fuzzy Rule Base

The structure of the rules of a type-2 FLS is similar to that of type-1. A type-2 FLS with n inputs, $x_1 \in X_1, \ldots, x_n \in X_n$, and m outputs, $y_1 \in Y_1, \ldots, y_m \in Y_m$, the p^{th} rule is of the form

$$R^p : \text{IF } x_1 \text{ is } \tilde{F}_1^p \text{ and } x_2 \text{ is } \tilde{F}_2^p \text{ and } \ldots \text{ and } x_n \text{ is } \tilde{F}_n^p$$
$$\text{THEN } y_1 \text{ is } \tilde{G}_1^p \text{ and } y_2 \text{ is } \tilde{G}_2^p \text{ and } \ldots \text{ and } y_m \text{ is } \tilde{G}_m^p$$

where $\tilde{F}_i^p$ and $\tilde{G}_j^p$, $i = 1, \ldots, n, j = 1, \ldots, m$, are input and output fuzzy labels, respectively.

3.4.3 Fuzzy Inference Engine

The inference engine is a key component of any FLS. It aggregates the if-then rules stored in the knowledge base with the fuzzy sets generated by the fuzzifier to form an overall

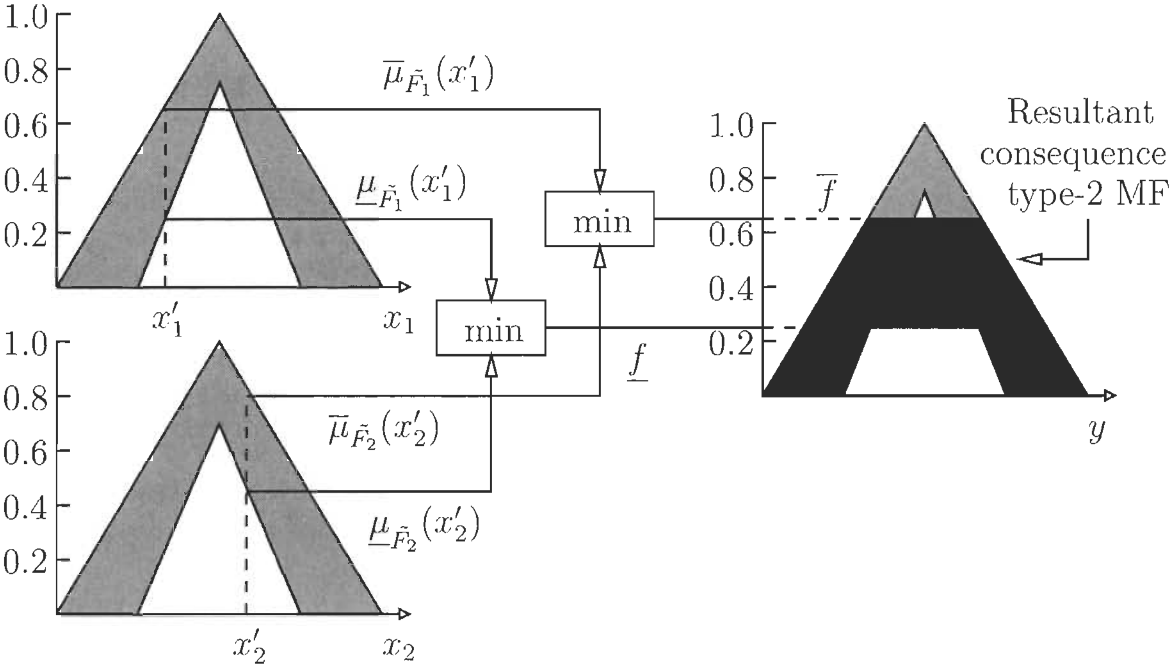

1.0
0.8
0.6
0.4
0.2
$\overline{\mu}_{\tilde{F}_1}(x'_1)$
$\underline{\mu}_{\tilde{F}_1}(x'_1)$
x'_1
x_1
min
min
1.0
0.8
0.6
0.4
0.2
$\overline{\mu}_{\tilde{F}_2}(x'_2)$
$\underline{\mu}_{\tilde{F}_2}(x'_2)$
x'_2
x_2
$\underline{f}$
1.0
0.8
0.6
0.4
0.2
$\overline{f}$
Resultant
consequence
type-2 MF
y

FIGURE 3.4 – Interval Type-2 Inference Process

output fuzzy set. Similarly, a type-2 fuzzy inference engine provides a mapping from the input type-2 fuzzy sets to the output ones. The intersection of multiple rule antecedents is computed using a t-norm operator while the union of multiple rules is computed through a t-conorm operation.

Each rule p in the knowledge base is interpreted as a type-2 fuzzy implication, which, when aggregated with the fuzzified inputs, infers a type-2 fuzzy set $\tilde{B}^p$ such that

$$\mu_{\tilde{B}^p}(y) = \sqcup_{x \in X}[\mu_{\tilde{A}_x}(x) \sqcap \mu_{R^p}(x,y)].$$

The t-norm and t-conorm used for the type-2 FLS herein are the "minimum" and "maximum" operators, respectively. These operators have been accredited in the literature for their computational efficiency and satisfactory performance.

3.4.4 Fuzzy Type Reduction

The type-2 fuzzy inference engine produces an aggregated output type-2 fuzzy set. The type reduction block operates on this set to generate a centroid type-1 fuzzy set known as the "type-reduced set" of the aggregate type-2 fuzzy set. Several type-reduction methods have been suggested in the literature, such as the center-of-sums, the height, the modified height and the center-of-sets, for example. We consider the center-of-sets type reduction technique thanks to its computational efficiency [8]. The calculation of type-reduced sets is performed in two stages. First, the centroids of the type-2 interval consequent sets of the fuzzy rules are computed. This is conducted ahead of time and is not part of the control cycle. In the second stage, the type-reduced sets are computed at each control cycle before being defuzzified [6].

Calculation of the Rule Consequents Centroids The centroid of the t^{th} output fuzzy set y_k^t is a type-1 interval set determined by its left and right most points, y_{lk}^t and y_{rk}^t, respectively, which are expressed by [19] :

$$y_k^t = [y_{lk}^t, y_{rk}^t] = y(\theta_1, \dots, \theta_z) = \frac{\sum_{z=1}^{Z} y_z \theta_z}{\sum_{z=1}^{Z} \theta_z} \tag{3.6}$$

Algorithms 3 and 4 describe the iterative procedures for computing y_{rk}^t, and y_{lk}^t, respectively [6]. Without loss of generality, we assume that y_z, $z = 1, \dots, Z$, are arranged in an ascending order ; i.e., $y_1 \leq y_2 \leq \dots \leq y_Z$.

Note that both algorithms are guaranteed to converge in at most Z iterations.

3.4.5 Fuzzy Type-Reduced Set Calculation

Instead of aggregating the consequent type-2 fuzzy sets fired by the fuzzy rules before reduction, the center-of-sets type reduction uses the centroid method to reduce these resulting type-2 fuzzy sets to obtain an interval type-1 fuzzy set $[y_{lk}^p, y_{rk}^p]$ for each rule p. The inferenced

Algorithm 3: Computing y^{l}_{rk}

```
begin
    Set $\theta_z = h_z$ for $z = 1, \ldots, Z$
    Compute $y' = y(h_1, \ldots, h_z)$ using (3.6)
    Set Stop = False
    while Stop = False do
        Find $e$, where $1 \leq e \leq Z - 1$, such that $y_e \leq y' \leq y_{e+1}$
        Set $\theta_z = h_z - \Delta_z$ for $z \leq e$
        Set $\theta_z = h_z + \Delta_z$ for $z > e + 1$
        Compute $y'' = y(h_1 - \Delta_1, \ldots, h_e - \Delta_e, h_{e+1} + \Delta_{e+1}, \ldots, h_z + \Delta_z)$ using (3.6)
        if $y'' = y'$ then
            Stop = True
            Set $y^{l}_{rk} = y''$      ($y''$ is the maximum value of $y(\theta_1, \ldots, \theta_z)$)
        else
            Set $y' = y''$
end
```

Algorithm 4: Computing y^{l}_{lk}

```
begin
    Set $\theta_z = h_z$ for $z = 1, \ldots, Z$
    Compute $y' = y(h_1, \ldots, h_z)$ using (3.6)
    Set Stop = False
    while Stop = False do
        Find $e$, where $1 \leq e \leq Z - 1$, such that $y_e \leq y' \leq y_{e+1}$
        Set $\theta_z = h_z + \Delta_z$ for $z \leq e$
        Set $\theta_z = h_z - \Delta_z$ for $z > e + 1$
        Compute $y'' = y(h_1 + \Delta_1, \ldots, h_e + \Delta_e, h_{e+1} - \Delta_{e+1}, \ldots, h_z - \Delta_z)$ using (3.6)
        if $y'' = y'$ then
            Stop = True
            Set $y^{l}_{lk} = y''$      ($y''$ is the maximum value of $y(\theta_1, \ldots, \theta_z)$)
        else
            Set $y' = y''$
end
```

interval type-1 fuzzy set is then defined by $[y_{lk}, y_{rk}]$, such as :

$$y_{lk} = \frac{\sum_{p=1}^{R} f_l^p y_{lk}^p}{\sum_{p=1}^{R} f_l^p} \tag{3.7}$$

$$y_{rk} = \frac{\sum_{p=1}^{R} f_r^p y_{rk}^p}{\sum_{p=1}^{R} f_r^p} \tag{3.8}$$

where f_l^p, f_r^p are the firing strengths corresponding to y_{lk}^p and y_{rk}^p of rule p, to minimize y_{lk}^p and maximize y_{rk}^p. Algorithms 5 and 6 reveal the iterative procedures to compute y_{lk} and y_{rk}, respectively [19]. Without loss of generality, we here assume that the pre-computed y_{lk}^p, $p = 1, \ldots, R$, are arranged in an ascending order ; i.e., $y_{lk}^1 \leq y_{lk}^2 \leq \ldots \leq y_{lk}^R$. Hence, y_{lk} and y_{rk} can be mathematically expressed as :

$$y_{lk} = \frac{\sum_{u=1}^{Q} \overline{f}^u y_{lk}^u + \sum_{v=Q+1}^{R} \underline{f}^v y_{lk}^v}{\sum_{u=1}^{Q} \overline{f}^u + \sum_{v=Q+1}^{R} \underline{f}^v}$$

$$y_{rk} = \frac{\sum_{u=1}^{R} \underline{f}^u y_{rk}^u + \sum_{v=R+1}^{R} \overline{f}^v y_{rk}^v}{\sum_{u=1}^{R} \underline{f}^u + \sum_{v=R+1}^{R} \overline{f}^v}$$

It is worth pointing out that both procedures are proven to converge in no more than R iterations, where R is the total number of rules [19].

3.4.6 Defuzzification

Eventually, the type-reduced set $\mu_{\tilde{B}^l}(y)$ determined by its left most and right most points, y_{lk} and y_{rk}, respectively, is defuzzified using the interval set average formula to get a crisp output value. As such, the defuzzified crisp output for each output k is formulated as [8] :

$$Y_k(x) = \frac{y_{lk} + y_{rk}}{2}$$

Algorithm 5: Computing y_{lk}

```
begin
    Set $f_l^p = (\underline{f}^p + \overline{f}^p)/2$, for $p = 1, \ldots, R$, with $\underline{f}^p$ and $\overline{f}^p$ are as defined in (3.3) and (3.4)
    Set $y'_{lk} = y_{lk}$
    Set Stop = False
    while $Stop = False$ do
        Find $Q$ $(1 \leq Q \leq R - 1)$ such that $y_{lk}^Q \leq y'_{lk} \leq y_{lk}^{Q+1}$
        Compute $y_{lk}$ as in (3.7), using $f_l^p = \overline{f}^p$ for $p \leq Q$ and $f_l^p = \underline{f}^p$ for $p > Q$
        Let $y''_{lk} = y_{lk}$
        if $y''_{lk} = y'_{lk}$ then
            Stop = True
            Set $y''_{lk} = y_{lk}$
        else
            Set $y'_{lk} = y''_{lk}$
end
```

Algorithm 6: Computing y_{rk}

```
begin
    Set $f_r^p = (\underline{f}^p + \overline{f}^p)/2$, for $p = 1, \ldots, R$, with $\underline{f}^p$ and $\overline{f}^p$ are as defined in (3.3) and (3.4)
    Set $y'_{rk} = y_{rk}$
    Set Stop = False
    while $Stop = False$ do
        Find $R$ $(1 \leq R \leq R - 1)$ such that $y_{rk}^R \leq y'_{rk} \leq y_{rk}^{R+1}$
        Compute $y_{rk}$ as in (3.8), using $f_r^p = \overline{f}^p$ for $p \leq R$ and $f_r^p = \underline{f}^p$ for $p > R$
        Let $y''_{rk} = y_{rk}$
        if $y''_{rk} = y'_{rk}$ then
            Stop = True
            Set $y''_{rk} = y_{rk}$
        else
            Set $y'_{rk} = y''_{rk}$
end
```

3.5 Soft-Computing Based Optimization

Neural Networks and Fuzzy Logic have recently received growing attention, thanks to their ability in solving real-life nonlinear time variant complex problems. Although these techniques have had successes over the years, they have limitations as well. Intelligent combinations of these two technologies can exploit their advantages while eliminating their limitations. Such combinations of neural networks and fuzzy logic are called neuro-fuzzy systems. On the other hand, scientists studied in the early 1950s evolutionary algorithms (EA) to be used as an optimization tool for engineering problems. The idea is to develop a search engine using operators inspired by nature to find optimal solutions to a given problem. The evolutionary systems we are concerned with in this chapter are Genetic Algorithms (GA) and Ant Colony Optimization (ACO) algorithms. Recently, researchers attempted the use of evolutionary computing for the design of fuzzy logic systems, and for automatic training and generation of neural network architectures. This chapter serves as an overview of the most popular approaches.

3.5.1 Genetic Algorithms (GA)

Genetic algorithm (GA), invented by John Holland in the 1960s, is a heuristic search technique inspired by evolutionary biology [20]. In contrast with evolution strategies, Holland's studied the nature selection phenomenon as an adaptation mechanism to be imported into computer systems. This mechanism consists of the use of candidate solutions, called elements of population, represented in a binary form, called chromosomes. The adaptation is done with moving from one population of chromosomes to a new population by using natural selection genetic inspired operators of crossover, mutation, and inversion. In each generation, the fitness of every individual in the population is evaluated and the selection operator chooses the chromosomes in the population (based on their fitness) that are allowed to reproduce evolving toward better solutions. The new population is then used in the next generation and the process usually terminates when a satisfactory fitness level has been reached for the population.

GA Operators

The simplest form of genetic algorithm involves three types of operators : selection, crossover (single point), and mutation.

- **Selection :** This operator selects chromosomes with the best fitness level in the population for reproduction.
- **Crossover :** This operator mimics a biological recombination between two single chromosome organisms. The subsequences of two chromosomes exchange before and after a randomly chosen locus to form two offsprings. For example, the crossover after the third locus of chromosomes 01000101 and 11111111 produce the two offspring 01011111 and 11100101.
- **Mutation :** This operator randomly toggles a chromosome bit. For example, the chromosome 00000010 mutated in its fourth position yields 00001010.

A simple genetic algorithm is described in algorithm 7.

Algorithm 7: Genetic

```
begin
    – Generate randomly a population of $n$ chromosomes.
    repeat
        – Calculate the fitness $f(x)$ of each chromosome $x$ in the population.
        repeat
            – Select a pair of parent chromosomes based on their fitness level.
            – Form two offsprings wih a probability crossover of parent chromosomes.
            – Mutate the two offsprings at each locus with a mutation probability.
            – If $n$ is odd, one new population member can be discarded at random.
        until n offsprings have been created.
    until satisfactory fitness level is reached.
end
```

3.5.2 Ant Colony Optimization (ACO) Algorithms

The Ant Colony Optimization (ACO) algorithm is a probabilistic technique used to solve optimization and distributed control problems [21]. It is derived from the behavior of real ants in finding the best path from food to the colony. Ants lay down pheromone when they find a good path, which increases the probability that other ants follow the same path instead of traveling randomly. Over time, the best path receives more pheromones and is likely to be followed by all ants. However, pheromone evaporates reducing the attractive strength of its path. This phenomenon has the advantage of avoiding the convergence to a local optimal solution.

ACO algorithm is characterized by two phases : forward and backward, where the forward phase is when ants move from the nest toward the food. In this phase, ants build a solution by choosing probabilistically the next node among those in the neighborhood in which they are located. This probabilistic choice is biased by pheromone trails previously deposited by other ants. It is noteworthy that ants do not deposit any pheromone while moving forward to avoid the formation of loops. In the backward phase, ants use an explicit memory to retrace the followed path and eliminate loops from it. While moving backward, ants leave pheromone on the trails they traverse. Moreover, ants deposit a higher amount of pheromone on short paths or rich food sources making pheromone update a function of the generated solution quality that helps in directing future ants more strongly toward better solutions. Therefore, path searching is more quickly biased toward the best solutions as described in algorithm 8.

Algorithm 8: Ant colony optimization algorithm

```
begin
    Initialize data
    repeat
        – Forward phase :
          - Construct pheromone biased paths : build solution candidates.
          - Evaluate paths : search for optimal solutions.
        – Backward phase :
          - Eliminate loops from the forward paths.
          - Update pheromone as a function of solution quality.
    until terminate condition is reached.
end
```

3.5.3 Hybrid Systems

Input-output data can be used to generate fuzzy logic rules and membership functions, significantly reducing the design time and complexity. This provides a more cost effective solution as fuzzy implementation is typically a less expensive alternative than neural networks for embedded control applications. Moreover, expressing the ANN weights by fuzzy rules provides greater insights and leads to a design of better solution. The hybrid intelligent system can be classified in three main categories : cooperative, concurrent and integrated neuro-fuzzy systems.

In a cooperative neuro-fuzzy system, the ANN learning mechanism is used to determine the membership functions or the fuzzy rules, parameters of fuzzy sets, rule weights, etc. In this system, a supervised learning technique uses a gradient descent procedure to tune the fuzzy parameters. The learning takes place in an offline mode and once the training is done, the ANN is disconnected from the control structure. The same technique can be applied to the fuzzy operators, and the defuzzification engine.

In a concurrent neuro-fuzzy system, ANN assists the fuzzy system continuously (or vice versa) to determine the unknown parameters. As such, it improves the performance of the overall system. The learning takes place in the ANN that acts as a postprocessor of fuzzy outputs or a preprocessor of the input data.

In an integrated neuro-fuzzy system, the fuzzy inference system takes the form of ANN structure and uses human expertise by storing its essential knowledge-based components. The nodes of the structure are presented in layers representing linguistic labels, rules, antecedent and consequent fuzzy parts. A supervised learning technique (back-propagation) is then used to adapt the weights.

However, gradient-based supervised learning techniques are likely to be trapped in local optima when the error surface is irregular. This problem can be avoided by using global evolutionary optimization procedures such as genetic and ant colony algorithms.

3.6 Conclusion

Fuzzy logic approximates complex systems using human reasoning. Thus, our knowledge can be used to model effectively the uncertainty and nonlinearity within complex systems. Moreover, fuzzy logic is easy to implement and is cost effective for a wide range of applications. However, it becomes more challenging to determine the correct set of rules and membership functions as the system complexity increases. Membership functions tuning and rules adjustment can be a time consuming process. In addition, the use of fixed geometric-shaped membership functions and rules limits fuzzy logic systems knowledge. These weaknesses have addressed with the combination of fuzzy logic system and other soft-computing techniques.

On the other hand, GAs were presented as an algorithmic concept based on Darwin's theory of evolution by natural selection, where stronger individuals in the population have a higher chance of reproduction. A genetic algorithm implements a search and optimization procedure that creates a population of candidate solutions, and then let that population evolve over time through competition using the three fundamental genetic operations of selection, crossover and mutation. On the other hand, ant colony optimization is another procedure that can also be applied to this kind of problem. In spite of the simplicity of ant colony individuals, their union presents a highly structured organization able to accomplish complex tasks that far exceed the single ant capabilities. These evolutionary algorithms span many points in the search space simultaneously which reduce chances of converging to local minima and provide a rapid convergence to an optimum solution. Despite of the witnessed success of evolutionary algorithms, they suffer from heavy computational complexity when used for complex problems. Finally, different ways to learn fuzzy inference systems using ANN learning techniques were also presented. The data acquisition and preprocessing training data is also quite important for the success of neuro-fuzzy systems. Many neuro-fuzzy models use supervised/unsupervised techniques to learn the different parameters of the fuzzy inference system. Unlike evolutionary algorithms, ANNs and FLSs are used in this thesis for their popularity and low computational complexity. Therefore, the resulted power from their combination is exploited to design advanced intelligent adaptive control techniques for nonlinear dynamic systems with high uncertainties. Moreover, new Lyapunov-based adaptation techniques are proposed for online tuning to guarantee the stability of such systems.

Chapitre 4

Nonlinear Control

4.1 Introduction

In real life, most of physical systems are nonlinear. Thus, a control system is often required to cope with these nonlinearities. However, if the nonlinear system's operating range is small enough to be reasonably approximated by a linearized system, a variety of powerful linear control techniques can then apply efficiently. However, nonlinearities can be continuous and discontinuous. Discontinuous nonlinearities, also called hard nonlinearities, cannot be approximated by linear systems. Therefore, this chapter presents an overview of advanced control design techniques for nonlinear systems. These control techniques will be used latter to design some Lyapunov-based adaptive control approaches that are suitable for uncertain complex systems.

4.2 Nonlinear Systems

A nonlinear system is usually represented by

$$\dot{x} = f(x, u, t) \tag{4.1}$$

where $u \in \mathbb{R}^n$ is the control input vector, $f \in \mathbb{R}^n$ is a nonlinear vector function, $x \in \mathbb{R}^n$ is a state vector, and n is number of states, also called the order of the system.

Definition 3 *[22] The nonlinear system (4.1) is autonomous if f does not depend explicitly on time index t, i.e., the system can be written as*

$$\dot{x} = f(x,u) \tag{4.2}$$

Otherwise, the system is called non-autonomous.

4.3 Lyapunov Stability Theory

In control design, we need to investigate system's stability since unstable systems are useless and undesirable. The study of the behavior of a Lyapunov function candidate of a nonlinear closed-loop system is a general and efficient way to analyze its stability. Along this spirit, Lyapunov functions can be used to design stabilizing feedback and adaptive control laws. Lyapunov stability was first introduced in 1892 by the mathematician Lyapunov. However, this concept was only brought to the attention of the control engineering community in the early 1960's.

Definition 4 *[22] A state x^* is an equilibrium state (or equilibrium point) of the system if once $x(t)$ is equal to x^*, it remains equal to x^* for all future time.*

A nonlinear system (4.2) can have several equilibrium points that can be found by solving for,

$$f(x^*,u) = 0$$

A specific equilibrium point can also be defined as,

$$y = x - x^* = 0$$

Therefore, $x = x^*$ is an equilibrium point.

There are two methods for stability analysis, indirect and direct method. The Lyapunov's indirect method, also called linearization method is used to analyse stability of linear time-invariant (LTI) systems and local stability of nonlinear systems around an operating point. The Lyapunov's direct method is not restricted to local motion. It is used to analyse the stability properties of nonlinear systems with a time-varying "energy-like" function.

4.3.1 Autonomous Systems

In this section, we present an overview of the Lyapunov stability theory applied to autonomous systems.

Fondamentals of Lyapunov theory :

Definition 5 *[22] A scalar continuous function $V(x)$ is locally positive definite if $V(0) = 0$ and, in a ball B*

$$V(x) > 0 \quad \forall\, x \neq 0, ||x|| < B$$

If $V(0) = 0$ and the above property holds over the whole state space, then $V(x)$ is globally positive definite.

Example : (positive definite)

Consider the following scalar continuous function :

$$V(x_1,x_2) = 4x_1^2 + 5x_2^2 = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x^T P x$$

P is a positive definite matrix and $V(x)$ is positive definite.

Definition 6 *A square matrix M is positive definite if for all $x \neq 0$, $x^T M x > 0$.*

Therefore,

$$V(x_1,x_2) = x^T P x > 0 \quad \forall\, x \neq 0$$

This implies that $V(x_1,x_2)$ is globally positive definite.

Example : (positive semi-definite)

The following scalar continuous function,

$$V(x_1,x_2) = x_1^2 = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x^T P x \geq 0$$

is positive semi-definite.

The concept of stability :

Stability can be viewed as if a system starts close to the origin, its trajectory will remain close to it. Otherwise, the system is unstable. However, since nonlinear systems are governed by complex dynamics, more refined stability concepts, such as asymptotic, exponential and global asymptotic stability, need to be defined. In the following, these concepts are introduced for autonomous systems.

Definition 7 *(Stability in the sense of Lyapunov [22]) The equilibrium state $x = 0$ (the origin) is stable if, for any $R > 0$, there exists $r > 0$, such that if $||x(0)|| < r$, then $||x(t)|| < R$ for all $t \geq 0$. Otherwise, the origin is unstable.*

Example : (stability)

Consider the pendulum system,

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\frac{g}{l} sin x_1$$

Consider the following energy function as a Lyapunov candidate function :

$$V(x) = \frac{g}{l}(1 - cos x_1) + \frac{1}{2}x_2^2$$

It is noteworthy that $V(0) = 0$ and $V(x)$ is positive definite over the domain $-\pi < x_1 < \pi$. The time-derivative of $V(x)$ is given by :

$$\dot{V}(x) = \frac{g}{l} sinx_1\dot{x}_1 + x_2\dot{x}_2 = 0$$

$V(x)$ is positive definite and $\dot{V}(x) = 0$ implies that the origin is stable.

The Lyapunov function candidate satisfies the stability conditions. However, it does not imply that the system's trajectory converges to the origin. Asymptotic stability states that if a system starts close to the origin, its trajectory will converge to it.

Definition 8 *[22] The origin is asymptotically stable if it is stable, and if in addition there exists some $r > 0$ such that $||x(0)|| < r$ implies that $x(t) \to 0$ as $t \to \infty$.*

On the other hand, exponential stability shows how fast is the convergence to the origin.

Definition 9 *[22] The origin is exponentially stable if there exists two strictly positive numbers α and λ such that*

$$\forall\, t > 0, \quad ||x(t)|| \leq \alpha ||x(0)|| e^{-\lambda t}$$

in some ball B around the origin.

When a Lyapunov function candidate fails to satisfy the conditions for stability, asymptotic or exponential stability, it does not imply that the equilibrium is not stable, asymptotically or exponentially stable. This only means that such properties cannot be established by this Lyapunov function candidate. In many applications, it is not easy to find a Lyapunov function for a given system. Its existence is defined as follows,

Definition 10 *[22] If, in a ball B, the function $V(x)$ is positive definite and has continuous partial derivatives, and if its time derivative along any state trajectory of the system (4.2) is negative semi-definite, i.e.,*

$$\dot{V}(x) \leq 0$$

then $V(x)$ is a Lyapunov function of the system (4.2).

The above definitions are stated to describe the system's local behavior, in a ball B. Therefore, the local stability theorem can be formulated as,

Theorem 2 *(local stability) [22]*
If, in a ball B, there exists a scalar function $V(x)$ with continuous first partial derivatives such that

- *$V(x)$ is positive definite (locally in B).*
- *$\dot{V}(x)$ is negative semi-definite (locally in B).*

then the origin is stable. And if the derivative $\dot{V}(x)$ is locally negative definite in B, then the origin is asymptotically stable.

Global stability is another importance aspect as the system's behavior when initial states are far from the origin is not covered by local stability concept.

Definition 11 *[22] If the origin is asymptotically or exponentially stable for any initial states, then the equilibrium state is globally asymptotically or exponentially stable.*

Therefore, the global stability theorem can be formulated as,

Theorem 3 *(global stability) [22]*
If there exists a scalar function $V(x)$ with continuous first order derivatives such that

- *$V(x)$ is positive definite.*
- *$\dot{V}(x)$ is negative definite.*
- *$V(x) \to \infty$ as $||x|| \to \infty$.*

then the origin is globally asymptotically stable.

Example : (global stability)

Consider the scalar system,

$$\dot{x} = -x^3 \qquad x \in \mathbb{R}$$

Consider the candidate Lyapunov function :

$$V(x) = \frac{1}{2}x^2$$

It is noteworthy that $V(x) \to \infty$ as $||x|| \to \infty$. The time-derivative of $V(x)$ is given by :

$$\dot{V}(x) = -x^4 \leq 0$$

$\dot{V}(x)$ is negative definite for all $x \in \mathbb{R}$. Therefore, the origin is globally asymptotically stable.

4.3.2 Non-Autonomous Systems

In this section, we present an overview of the Lyapunov stability theory applied to non-autonomous systems.

Definition 12 *[22] The origin is stable if, for any $R > 0$, there exists a positive scalar $r(R,t_0)$ such that*

$$\forall\, t \geq t_0, \quad ||x(t_0)|| < r \quad \Rightarrow \quad ||x(t)|| < R$$

Otherwise, the origin is unstable.

As in autonomous systems, the state is kept in a ball of arbitrarily small radius R when the state trajectory starts in a ball of sufficiently small radius r. However, the radius r of the initial ball may depend on the initial time t_0 in non-autonomous systems. Since the concept of stability does not guarantee the convergence of the system's trajectory to the origin, asymptotic stability of non-autonomous systems is defined as,

Definition 13 *[22] The origin is asymptotically stable if*

- *it is stable.*
- *there exists $r(R,t_0) > 0$ such that $||x(t_0)|| < r(R,t_0)$ implies that $x(t) \to 0$ as $t \to \infty$.*

Definition 14 *[22] The origin is globally asymptotically stable if*

$$\forall\, x(t_0), \quad x(t) \to 0 \; as \; t \to \infty$$

Figure 4.1 shows the concepts of stability.

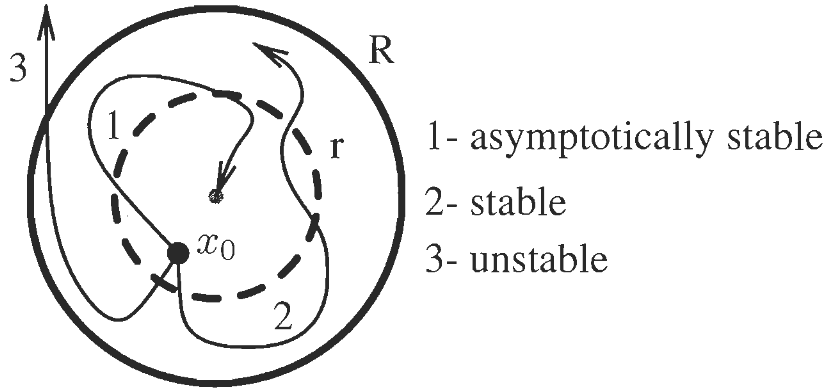


FIGURE 4.1 – Concepts of stability

Although the system is stable, the main concern in some applications remains the system trajectory convergence speed, i.e, how fast it approaches the origin. This requirement is captured by the concept of exponential stability.

Definition 15 *[22] The origin is exponentially stable if there exists two strictly positive numbers α and λ such that for sufficiently small $x(t_0)$*

$$\forall\, t \geq t_0, \quad ||x(t)|| \leq \alpha ||x(t_0)|| e^{-\lambda(t-t_0)}$$

So far, the concepts of stability are defined at a time instant t_0. Uniform stability is a concept which guarantees that the equilibrium point stability is independent of t_0. It is defined as follows

Definition 16 *[22] The origin is locally uniformly stable if the scalar r in definition 12 can be chosen independently of t_0, i.e., $r = r(R)$.*

For time-invariant systems, uniform stability and asymptotic stability implies uniform asymptotic stability, which is defined as follows

Definition 17 *[22] The origin is locally uniformly asymptotically stable if*

- *it is uniformly stable.*
- *there exists a ball of attraction B, whose radius is independent of t_0, such that all system trajectories with initial states in B converge to 0 uniformly in t_0.*

Before introducing the concept of global uniform asymptotic stability, let us define positive definite and decrescent functions.

Definition 18 *[22] A scalar time-varying function $V(x,t)$ is locally positive definite if $V(0,t) = 0$ and there exists a time-invariant positive definite function $V_0(x)$ such that*

$$\forall\, t \geq t_0, \quad V(x,t) \geq V_0(x)$$

Globally positive definite functions can be defined similarly [22].

Definition 19 *[22] A scalar time-varying function $V(x,t)$ is said to be decrescent if $V(0,t) = 0$ and there exists a time-invariant positive definite function $V_1(x)$ such that*

$$\forall\, t \geq 0, \quad V(x,t) \leq V_1(x)$$

Therefore, the concept of global uniform asymptotic stability is defined as follows

Theorem 4 *[22] If, in a ball B around the origin, there exists a scalar function $V(x,t)$ with continuous partial derivatives such that*

- *$V(x,t)$ is positive definite.*
- *$\dot{V}(x)$ is negative semi-definite ($\dot{V}(x,t) \leq 0$).*

then the origin is stable in the sense of Lyapunov. If, furthermore,

- *$V(x,t)$ is decrescent.*

then the origin is uniformly stable. If $\dot{V}(x)$ is negative definite ($\dot{V}(x,t) < 0$), then the origin is uniformly asymptotically stable.

In addition, if the ball B is replaced by the whole system state space and

- *$V(x,t)$ is radially unbounded.*

then the origin is globally uniformly asymptotically stable.

For non-autonomous systems, it is difficult to find a Lyapunov function $V(x,t)$ with a negative definite derivative ($\dot{V}(x,t) < 0$) to conclude the asymptotic stability. In the case of autonomous systems, even if $\dot{V}(x)$ is negative semi-definite, the asymptotic stability can still be determined by using invariant-set theorems. Since these theorems do not apply for time-varying systems, Barbalat's lemma is then applied.

Lemma 1 *(Barbalat)*

If the differentiable function $f(t)$ has a finite limit as $t \to \infty$, and if $\dot{f}(t)$ is uniformly continuous, then $\dot{f}(t) \to 0$ as $t \to \infty$.

Lemma 2 *(Lyapunov-Like Lemma) [22]*

If a scalar function $V(x,t)$ satisfies the following conditions

- *$V(x,t)$ is lower bounded.*
- *$\dot{V}(x,t)$ is negative semi-definite.*
- *$\dot{V}(x,t)$ is uniformly continuous in time.*

then $\dot{V}(x,t) \to 0$ as $t \to \infty$.

Example : (Barbalat's Lyapunov-based stability)

Consider the following nonlinear system :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -2x_1 + x_2 u \\ -x_1 u \end{bmatrix}$$

where, u is a bounded input signal.

Consider the following Lyapunov candidate :

$$V = \frac{1}{2}||x||^2 = \frac{1}{2}\left(x_1^2 + x_2^2\right)$$

Take the time-derivative of V :

$$\dot{V} = x_1\dot{x}_1 + x_2\dot{x}_2$$

Substitute for $\dot{x}_1$ and $\dot{x}_2$ yields,

$$\dot{V} = x_1(-2x_1 + x_2 u) - x_2 x_1 u$$

Thus,

$$\dot{V} = -2x_1^2 \leq 0$$

Therefore, the system is stable in the sense of Lyapunov.

A positive function V, which is decreasing ($\dot{V} \leq 0$) must converge to a finite limit. This implies that x_1 and x_2 are bounded. Take the time-derivative of $\dot{V}$,

$$\ddot{V} = -4x_1\dot{x}_1 = -4x_1(-2x_1 + x_2u)$$

Thus, $\dot{V}$ is uniformly continuous since $\ddot{V}$ exists and is bounded. Therefore, Barbalat's Lemma shows that $\dot{V} \rightarrow 0$, and hence, $x_1 \rightarrow 0$ as $t \rightarrow \infty$.

4.4 Adaptive Control

Adaptive control aims to update the controller online in an automatic fashion to adjust its performance to dynamic changes. An adaptive controller combines a control law, usually designed from the known parameter case, and an online parameter estimator, called an adaptive law. There are two different approaches, indirect and direct adaptive control [23].

- **Indirect Case**

  The indirect adaptive control scheme is depicted in Fig. 4.2. In this case, the plant parameters are explicitly estimated online and used to find the controller parameters.

- **Direct Case**

  The direct adaptive control scheme is depicted in Fig. 4.3. In this case, the controller is designed based on the plant model. Hence, the controller is parameterized in terms of the plant parameters that are estimated implicitly from the controller parameters estimation.

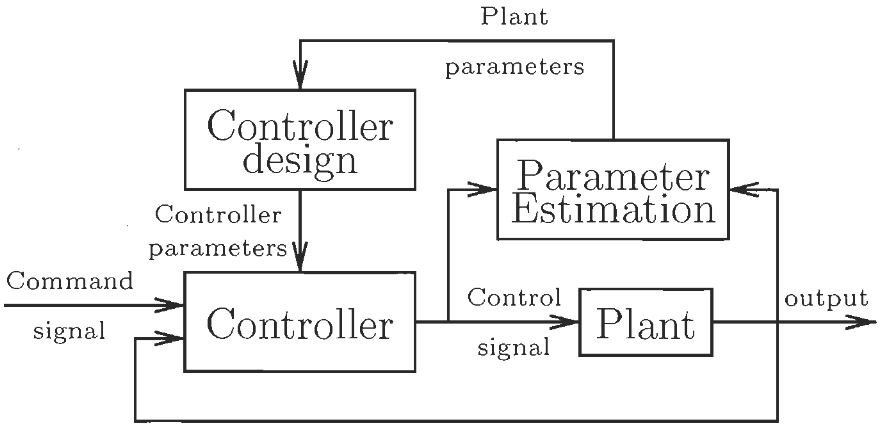


FIGURE 4.2 – Indirect adaptive control

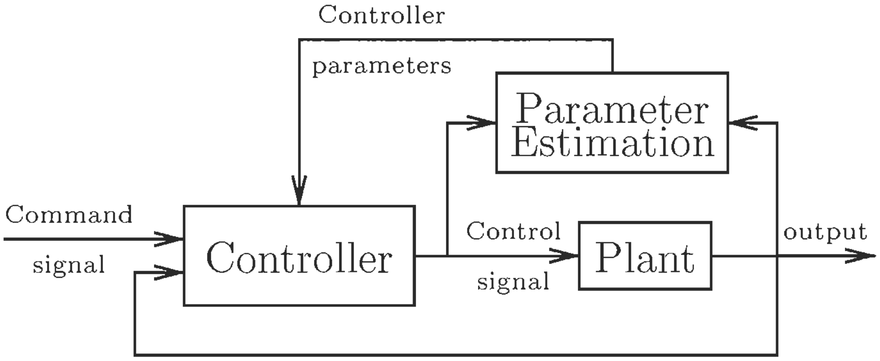


FIGURE 4.3 – Direct adaptive control

4.4.1 Online Parameter Estimation

Online parameter estimation is a key element of adaptive control systems. In the direct adaptive control literature, the online parameter estimator has often been referred to as the adaptive law, or update law, which is crucial for the stability of adaptive controllers. In this section, we introduce the most popular techniques for online parameter estimation.

Least Squares (LS)

Least Squares (LS) was introduced by Gauss at the late 1790s [23]. It is a basic parameter estimation technique applied when the model has a linear in parameters property

$$y(t) = \phi_1(t)\ \theta_1 + \phi_2(t)\ \theta_2 + \ldots + \phi_n(t)\ \theta_n = \Phi^T(t)\ \Theta(t)$$

where $y(t)$ is the observed variable, $\Theta(t)$ is a vector of the unknown parameters of the model, and $\Phi(t)$ is a vector of known functions, called regression vector or regressor.

$y(t)$ and $\Phi(t)$ are obtained from measurements and we aim to estimate $\Theta(t)$ by minimizing the least-squares loss function

$$V(\theta,t) = \frac{1}{2}\sum_{t=1}^{N}(y(t) - \Phi^T(t)\Theta(t))^2 \tag{4.3}$$

Theorem 5 *(least-squares estimation)*
Function (4.3) is minimal for parameters $\hat{\Theta}$ such that

$$\Phi^T\Phi\hat{\Theta} = \Phi^T Y$$

If the matrix $\Phi^T\Phi$ is nonsingular, the minimum is unique and given by

$$\hat{\Theta} = (\Phi^T\Phi)^{-1}\Phi^T Y$$

Recursive Least Squares (RLS)

Theorem 6 *(recursive least-squares estimation) [23]*
Assume that the matrix $\Phi(t)$ has full rank, that is, such that $\Phi^T(t)\Phi(t)$ is nonsingular, for all $t \geq t_0$. Given $\hat{\Theta}(t_0)$ and $P(t_0) = ((\Phi^T(t_0)\Phi(t_0))^{-1}$, the least-squares estimate $\hat{\Theta}(t)$ then satisfies the recursive equations

$$\begin{aligned}
\hat{\Theta}(t) &= \hat{\Theta}(t-1) + Q(t)(Y(t) - \Phi^T(t)\hat{\Theta}(t-1)) \\
Q(t) &= \frac{P(t-1)\Phi(t)}{\lambda I + \Phi^T(t)P(t-1)\Phi(t)} \\
P(t) &= \frac{1}{\lambda}\left(P(t-1) - \frac{P(t-1)\Phi(t)\Phi(t)^T P(t-1)}{\lambda I + \Phi^T(t)P(t-1)\Phi(t)}\right) \\
&= \frac{1}{\lambda}(I - Q(t)\Phi^T(t))P(t-1)
\end{aligned}$$

where P is the covariance matrix and λ is the forgetting factor.

The forgetting factor λ sets the effective length or memory of the estimator. When $\lambda = 1$, the estimator has long memory and the estimates become smoother since $Q(t)$ converges to zero. When $\lambda < 1$, $Q(t)$ does not converge to zero and the estimates always fluctuate. As a rule of thumb, the memory of the estimator is

$$N = \frac{2}{1-\lambda}$$

Least Mean Squares (LMS)

Least mean squares (LMS) was introduced by Widrow in the 1960s [23]. It is a stochastic gradient descent method that aims to minimize the cost function described as

$$\xi(\Theta) = \frac{1}{2}e^2(t)$$

where $e(t) = d(t) - y(t)$ is the error signal at time t, $d(t)$ and $y(t)$ are the measured and estimated output, respectively. Using the linear in parameters property, this signal can be written as

$$e(t) = d(t) - \Phi^T(t)\ \Theta(t)$$

Differentiating $e(t)$ with respect to the parameter vector $\Theta(t)$ yields,

$$\frac{\partial e(t)}{\partial \Theta(t)} = -\Phi(t)$$

Hence,

$$\frac{\partial \xi(\Theta)}{\partial \Theta(t)} = \frac{\partial \xi(\Theta)}{\partial e(t)} \frac{\partial e(t)}{\partial \Theta(t)} = -\Phi(t)e(t)$$

Therefore, the LMS estimate can be formulated as,

$$\hat{\Theta}(t+1) = \hat{\Theta}(t) + \eta\Phi(t)e(t)$$

where η is the adaptation rate.

Kalman Filter

Kalman filter is a discrete state-space recursive based estimation technique that provides solution to the linear optimal filtering problem. Therefore, there is no need for storing the entire past observed data since the algorithm is recursive. Therefore, Kalman filter is computationally more efficient. However, Kalman filter addresses only the estimation of a state vector of linear dynamical systems. When the model is nonlinear, the method is extended using a linearization of the nonlinear model. The resulting filter, called the extended Kalman filter (EKF), assumes that the deviation from linearity is of a first order. This first-order linearization of the nonlinear system may introduce large errors in the weight estimates and covariance matrix. The unscented Kalman filter (UKF) is then introduced as an alternative to the EKF providing a more accurate solution without increasing the computational complexity [18].

The MIT Rule

The MIT rule is used to design adaptive laws for various adaptive control schemes. It is a popular method, used with MRAC, to compute the approximate sensitivity functions by using online parameter estimates. However, it is difficult to prove global closed-loop stability and convergence of the tracking error to zero. Nevertheless, this method performs well when the adaptive gain γ and the reference input signal are small. This method is based on minimizing the square of the prediction error

$$J(\theta) = \frac{1}{2}e^2(\theta)$$

The parameter estimates are computed as follows

$$\frac{d\theta}{dt} = -\gamma e \frac{de}{dt}$$

Persistent Excitation

In many adaptive systems, one important aspect is the tracking error convergence. However, it gives a false impression that exponential parameter convergence is achieved. Persistent excitation conditions can ensure parameter convergence in adaptive algorithms if the following condition :

$$\alpha_0 I_n \leq \int_{t_0}^{t_0+\beta} \Phi\Phi^T dt \leq \alpha_1 I_n$$

is met for all t_0, where α_0, α_1 and β are all positive and Φ is the regressor vector. Note that the integral of $\Phi\Phi^T$ must be positive definite and bounded over all intervals of length β. In other word, Φ must vary sufficiently over the interval β so that the entire dimensional space is spanned.

4.4.2 Self-Tuning Regulators

Self-tuning regulators aim automatic tuning of controllers to respect some specifications. It is a slowly time-varying parameters estimation process, which often uses least squares methods. The self-tuning regulator control structure is depicted in Fig. 4.4.

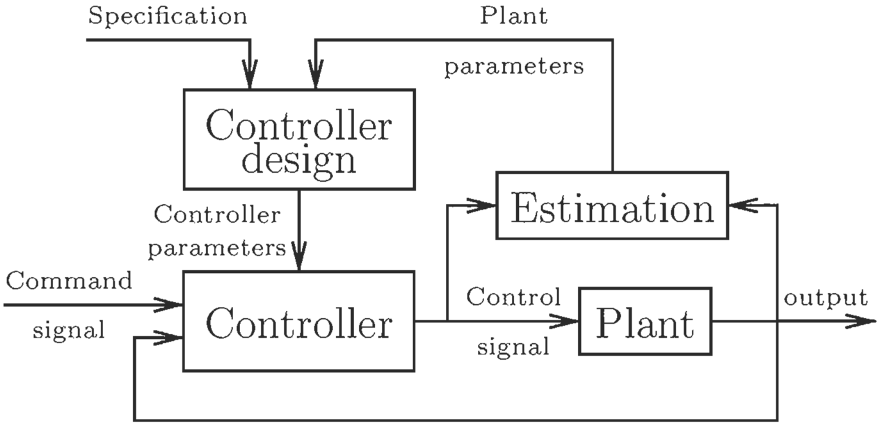


FIGURE 4.4 – Self-tuning regulator

4.4.3 Gain Scheduling

Gain scheduling is used when the plant dynamics variation due to different operating conditions is known or predictable. Hence, the controller gains vary in a predetermined fashion with the operating conditions. The gain scheduler consists of a lookup table where controller gains are stored. An appropriate logic detects the operating point and chooses the corresponding controller gains from the lookup table. One of the disadvantages of this technique is that the controller gains are computed offline and, therefore, it provides no feedback to compensate for incorrect schedules. On the other hand, the main advantage remains the fast change in the controller gains in response to quick plant parameter variations. The gain scheduling control structure is depicted in Fig. 4.5.

4.4.4 Model Reference Adaptive Control (MRAC)

Model reference adaptive control (MRAC) is one of the popular approaches in adaptive control and has been exhaustively explored by many researchers. MRAC uses reference and parameter adjustable models. Based on predefined desired system dynamics, the reference

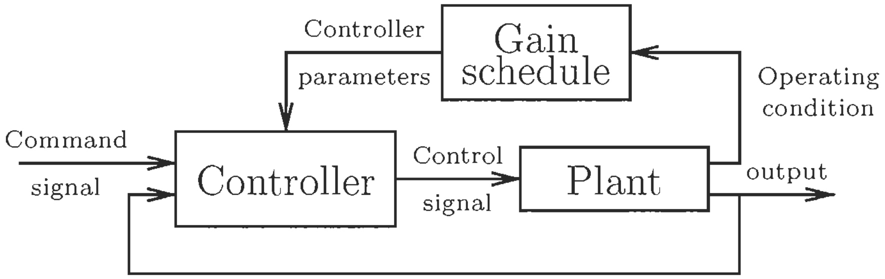


FIGURE 4.5 – Gain scheduling

model is chosen to generate the desired trajectory y_m. The tracking error $e = y_m - y_p$ is used to adjust the model parameters through an adaptation law. Thus, the reference model output y_m follows the one of the plant y_p. The MRAC structure is depicted in Fig. 4.6.

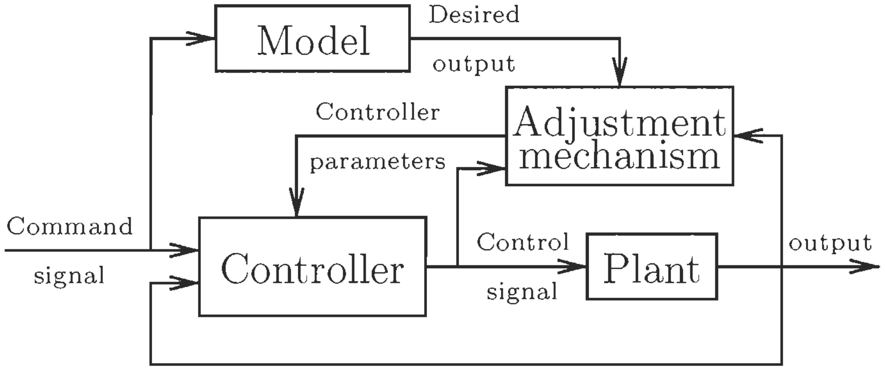


FIGURE 4.6 – Model-reference adaptive control

4.5 Sliding-Mode Control

Since the early 1960s, sliding mode has attracted the attention of many researchers thanks for its design simplicity, robustness to uncertainties and disturbance, and wide variety of applications such as, trajectory control, model following, and state observation. Sliding mode control is a well known discontinuous control technique that drives a trajectory to a switching surface and maintains it on this surface once it has been reached. The local attractivity of the sliding surface can be expressed by the condition

$$s\dot{s} < 0$$

where s represents the error from the trajectory to the sliding line.

In principle, sliding mode is achieved by discontinuous control and switching at infinite frequency. However, switching frequency is limited in real life applications. The switching imperfections, such as switching time delays and small actuators time constants combined with the control discontinuities result in a particular dynamic behavior in the vicinity of the surface, which is commonly referred to as chattering. This phenomenon may excite unmodeled high frequency modes, which degrades the performance of the system and may even lead to instability. To overcome this problem, the boundary solution replaces the discontinuous control with a saturation function that approximates the sign function in a boundary layer of sliding mode manifold. This solution preserves partially the invariance property of sliding mode where states are confined to a small vicinity of the manifold, and convergence to zero cannot be guaranteed. Furthermore, robustness to uncertain parameter variations and disturbance is also compromised since it is obtained only when sliding mode truly occurs.

4.6 Conclusion

In this chapter, various concepts of stability have been defined for autonomous and non-autonomous systems. Lyapunov indirect method, also called linearization method, is concerned with the stability analysis of nonlinear systems' small motions around equilibrium points. On the other hand, Lyapunov direct method is a powerful tool used for global stability analysis of nonlinear systems. Different types of adaptive systems have been described. An overview of online parameter estimation techniques such as, least-squares method, LMS, RLS, and Kalman filter has been proposed. Self-Tuning Regulators use parameter estimation techniques as system identification and automatically tune the controller parameters to obtain the desired closed-loop performance. Gain scheduling is used when measurable variables correlate well with the plant dynamic changes. MRAC defines through a reference model the plant desired dynamics and adjusts the controller parameters to minimize the error between the plant and the reference model. Finally, sliding-mode control technique has been presented showing its simplicity and robustness to uncertainties. This chapter serves as a theoretical background for our research. As such, the adaptive control theory is combined to the Lyapunov stability theory in the following chapters to design Lyapunov-based adaptive control techniques for complex dynamic systems.

Chapitre 5

Intelligent Control of Nonlinear Multiple Input Multiple Output Systems

5.1 Introduction

A nonlinear multiple input multiple output (MIMO) system can be represented by,

$$\dot{x} = f(x, u, t)$$

where $f \in \mathbb{R}^n$ is a nonlinear vector function, $x \in \mathbb{R}^n$ is a state vector, u is the control input, and n is number of states, also called the order of the system.

General control structures are shown in Fig. 5.1, which are used in this thesis as basic schemes to design intelligent-based controllers for nonlinear dynamic systems. As we shall see later, a second controller can be added for more complex problems.

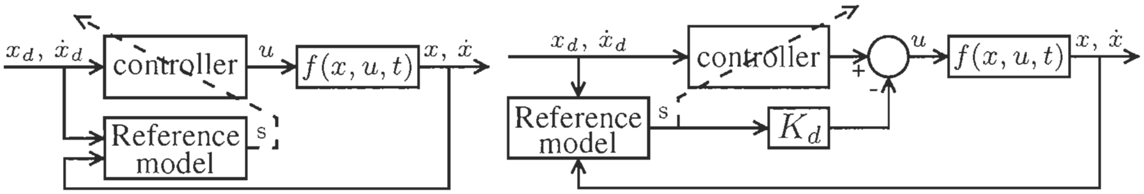


FIGURE 5.1 – General control structures

The control objective is to design a control law u to force the system's position x and velocity $\dot{x}$ to track their pre-defined time-dependent desired values x_d and $\dot{x}_d$. This objective is to be achieved under unknown dynamics and in the presence of external disturbances. Assume that the system's states x, and $\dot{x}$ are measurable and the nonlinear vector function $f(x,u,t)$ is *a priori* unknown.

Let $e = x - x_d$ and $\dot{e} = \dot{x} - \dot{x}_d$ denote the system's position and velocity errors, respectively and let us define the following reference error model,

$$s = \dot{e} + \Psi\, e$$

where Ψ is a positive diagonal constant gain matrix. The control objective is to minimize the reference model signal s. Although nonlinear control techniques such as adaptive control, variable structure control, and model predictive control are able to cope with parametric uncertainties, they are still vulnerable in the presence of unstructured modeling uncertainties. Since f is considered to be *a priori* unknown, soft-computing techniques such as, neural networks and fuzzy logic are good candidates for coping with such uncertainties, thanks to their learning and generalization capabilities. However, these tools are based on heuristic and gradient descent adaptation techniques, which do not guarantee their stability. Therefore, a Lyapunov stability-based adaptation technique is used as an alternative to the conventional heuristic tuning methods.

5.2 Neural Learning Algorithm

We can describe the input-output relationship of a single hidden layer neural network with a linear output layer activation function by :

$$v_j(k) = \sum_{i=1}^{m} \hat{w}_{ji}^{(1)}(k)\; x_i(k)$$

$$\hat{\phi}_j(k) = \varphi_j(v_j(k))$$

where φ_j is the jth node's activation function of layer, m is the number of input nodes, $x_i(k)$ and $\hat{\phi}_j(k)$ are, respectively, the input of node i and the output of node j of the first layer at time index k. v_j is the net input of node j. $\hat{w}_{ji}^{(1)}$ is the weight linking node i of the first layer to node j of the second layer. Then, the neural network output can be expressed as

$$\begin{aligned} y(k) &= \sum_{j=1}^{n} \hat{\phi}_j(k)\, \hat{w}_{1j}^{(2)}(k) \\ &= \sum_{j=1}^{n} \hat{w}_{1j}^{(2)}(k)\, \varphi\left(\sum_{i=1}^{m} \hat{w}_{ji}^{(1)}(k)\, x_i(k)\right) \end{aligned} \tag{5.1}$$

with n being the number of hidden nodes. The signal $y(k)$ defines the output of the neural network at instant k.

Let $e(k) = y(k) - d(k)$ denotes the neural network's error at instant k, where $d(k)$ represents the desired output of the neural network at instant k. The new hidden layer weights matrices are computed by the following rule :

$$\hat{w}_{ji}^{(1)}(k) = \frac{1}{m x_i(k)} \varphi^{-1}\left(\frac{y(k-1) - \eta e(k-1)}{n \hat{w}_{1j}^{(2)}(k)}\right) \tag{5.2}$$

where $0 < \eta < 2$ is the neural network's learning rate. If, for some k, numerical singularity occurs, that particular weight is not updated for the instant k. We also stop the online learning when the error is very small, preventing limit cycles. The proposed weight adaptation technique is an extension of the work presented in [17].

Theorem 7 *Assume a process resulting into continuous, bounded and sufficiently differentiable trajectories $x_i(k)$ and $d(k)$. The learning law (5.2) ($\forall\, x_i(k) \neq 0$ and $\forall\, \hat{w}_{1j}^{(2)}(k) \neq 0$) with a learning rate with η satisfying $0 < \eta < 2$ guarantees convergence of the estimation error to a small neighborhood of the origin that decreases with smaller sampling rate or lower time derivative of the desired output $d(k)$.*

Proof 1 *Consider the following Lyapunov candidate function :*

$$V(k) = \frac{1}{\eta} e^2(k)$$

$$\begin{aligned} \Delta V(k) &= V(k) - V(k-1) \\ &= \tfrac{1}{\eta} e^2(k) - \tfrac{1}{\eta} e^2(k-1) \end{aligned}$$

Substitute for $e(k)$, then :

$$\Delta V(k) = \frac{1}{\eta}(y(k) - d(k))^2 - \frac{1}{\eta} e^2(k-1)$$

Substitute $y(k)$ from (5.1) :

$$\Delta V(k) = \tfrac{1}{\eta}\left(\textstyle\sum_{j=1}^{n} \hat{w}_{1j}^{(2)}(k)\varphi\left(\sum_{i=1}^{m} \hat{w}_{ji}^{(1)}(k)x_i(k)\right) - d(k)\right)^2 - \tfrac{1}{\eta} e^2(k-1)$$

Substitute $\hat{w}_{ji}^{(1)}(k)$ from (5.2), leads to :

$$\Delta V(k) = \tfrac{1}{\eta}\left(\textstyle\sum_{j=1}^{n} \hat{w}_{1j}^{(2)}(k)\varphi\left(\sum_{i=1}^{m} \frac{1}{m x_i(k)}\varphi^{-1}\left(\frac{y(k-1)-\eta e(k-1)}{n\hat{w}_{1j}^{(2)}(k)}\right) x_i(k)\right) - d(k)\right)^2 - \tfrac{1}{\eta} e^2(k-1)$$

$$\Delta V(k) = \tfrac{1}{\eta}\left(\textstyle\sum_{j=1}^{n} \hat{w}_{1j}^{(2)}(k)\varphi\left(\sum_{i=1}^{m} \frac{1}{m \cancel{x_i(k)}}\varphi^{-1}\left(\frac{y(k-1)-\eta e(k-1)}{n\hat{w}_{1j}^{(2)}(k)}\right) \cancel{x_i(k)}\right) - d(k)\right)^2 - \tfrac{1}{\eta} e^2(k-1)$$

$$\Delta V(k) = \tfrac{1}{\eta}\left(\textstyle\sum_{j=1}^{n} \hat{w}_{1j}^{(2)}(k)\cancel{\varphi}\left(\cancel{\varphi}^{-1}\left(\frac{y(k-1)-\eta e(k-1)}{n\hat{w}_{1j}^{(2)}(k)}\right)\right)\right) - d(k)\Bigg)^2 - \tfrac{1}{\eta} e^2(k-1)$$

$$\Delta V(k) = \tfrac{1}{\eta}\left(\textstyle\sum_{j=1}^{n} \cancel{\hat{w}_{1j}^{(2)}(k)}\left(\frac{y(k-1)-\eta e(k-1)}{n\cancel{\hat{w}_{1j}^{(2)}(k)}}\right) - d(k)\right)^2 - \tfrac{1}{\eta} e^2(k-1)$$

$$\Delta V(k) = \tfrac{1}{\eta}(y(k-1) - \eta e(k-1) - d(k))^2 - \tfrac{1}{\eta} e^2(k-1)$$

Add and substract $e(k-1)$,

$$\Delta V(k) = \tfrac{1}{\eta}((1-\eta)e(k-1) - \Delta d(k))^2 - \tfrac{1}{\eta} e^2(k-1)$$

with,

$$d(k-1) = y(k-1) - e(k-1)$$
$$\Delta d(k) = d(k) - d(k-1)$$

If $\Delta d(k) = 0$ *then,*

$$\Delta V(k) = (\eta - 2)e^2(k-1) < 0$$

By setting $0 < \eta < 2$, $V(k)$ *is monitically decreasing and* $e(k)$ *converges to zero. However, if* $\Delta d(k) \neq 0$, *then convergence can only be guaranteed to a small neighborhood of the origin. Set* $\eta = 1$,

$$\Delta V(k) = \Delta d^2(k) - e^2(k-1)$$

Therefore, the neural network is stable in the sense of Lyapunov and converges to a small neighborhood of $e(k) = 0$, *which is a region defined by* $\Delta d(k)$ *and gets smaller for slowly time-varying systems or for fast neural network sampling rates.*

5.3 Fuzzy Learning Algorithm

The fuzzy logic system (FLS) output at instant k with a commonly used defuzzification method, such as, the centroid method is expressed as :

$$y_j(k) = \frac{\sum_{p=1}^{R} y_p \ \mu_{B^p}(y)}{\sum_{p=1}^{R} \mu_{B^p}(y)}$$

where, $p = 1, \ldots, R$ and R is the number of the fuzzy rules. Therefore, the output of a multiple input single output FLS can be expressed as,

$$y(k) = w^T(k)\ G(k) \tag{5.3}$$

with $w(k) = [y_1, y_2, \ldots, y_p]^T$ being the fuzzy logic consequent part weight vector at instant k, $p = 1, \ldots, R$, and $G(k)$ being fuzzy logic antecedent part vector of known functions (regressor) at instant k defined as :

$$G(k) = \frac{\mu_{B^p}(y)}{\sum_{p=1}^{R} \mu_{B^p}(y)}$$

Let $e(k) = y(k) - d(k)$ denotes the FLS's error at instant k, where $d(k)$ represents the desired output of the FLS at instant k. The new weight vector is computed by the following rule :

$$w^T(k) = \frac{y(k-1) - \lambda e(k-1)}{G(k)} \tag{5.4}$$

where λ is the FLS's adaptation rate. If, for some k, numerical singularity occurs, that particular weight is not updated for the instant k. Equation (5.3) shows that in this case the output $y(k)$ is independent of that particular weight.

Theorem 8 *Assume a process resulting into continuous, bounded and sufficiently differentiable trajectories $x_i(k)$ and $d(k)$. The adaptation law (5.4) ($\forall\ G(k) \neq 0$) with λ satisfying $0 < \lambda < 2$ guarantees convergence of the estimation error to a small neighborhood of the origin that decreases with smaller sampling rate or lower time derivative of the desired output $d(k)$.*

Proof 2 *Consider the following Lyapunov candidate function :*

$$V(k) = e^2(k)$$

$$\begin{aligned}\Delta V(k) &= V(k) - V(k-1) \\ &= e^2(k) - e^2(k-1)\end{aligned}$$

Substitute for $e(k)$, then :

$$\Delta V(k) = (y(k) - d(k))^2 - e^2(k-1)$$

Substitute $y(k)$ from (5.3) :

$$\Delta V(k) = \left(w^T(k)\ G(k) - d(k)\right)^2 - e^2(k-1)$$

Setting the weight adaptation law as,

$$w^T(k) = \frac{y(k-1) - \lambda e(k-1)}{G(k)}$$

yields,

$$\Delta V(k) = (y(k-1) - \lambda e(k-1) - d(k))^2 - e^2(k-1)$$

Substitute $y(k-1) = e(k-1) + d(k-1)$,

$$\Delta V(k) = ((1-\lambda)e(k-1) - \Delta d(k))^2 - e^2(k-1)$$

with, $\Delta d(k) = d(k) - d(k-1)$. Hence, $\Delta V(k)$ can be written as,

$$\Delta V(k) = a\varepsilon^2 + b\varepsilon + c$$

where, $a = (1-\lambda)^2 - 1$, $b = -2(1-\lambda)\Delta d(k)$, $c = \Delta d^2(k)$, and $\varepsilon = e(k-1)$. The fact that $a < 0$ yields $\Delta V(k) < 0$ for some range of ε, which guarantees the stability of the fuzzy logic system for $0 < \lambda < 2$. The roots of $\Delta V(k)$ are,

$$\varepsilon_1 = \frac{1}{2-\lambda}\Delta d(k)$$
$$\varepsilon_2 = -\frac{1}{\lambda}\Delta d(k)$$

Henceforth, $\Delta V(k) < 0$ when ε is outside of the span of these two roots. Thus, the stability and convergence of the fuzzy logic system is guaranteed to a small neighborhood of $e(k) = 0$ defined by these roots. This region is minimized when $\lambda = 1$,

$$\Delta V(k) = \Delta d^2(k) - e^2(k-1)$$

Therefore, the region is defined by $\Delta d(k)$ and gets smaller for slowly time-varying systems or for fast fuzzy logic sampling rates.

5.4 A Case of Underactuated Systems : Inverted Pendulums

Inverted pendulums have received a thorough attention and have been extensively used lately to demonstrate the effectiveness of different kinds of controllers. They are considered as a well established benchmark challenge for many control problems, such as robotic manipulators and missile control [24]. The severe nonlinearities, varying operating conditions, structured and unstructured dynamical uncertainties, such as Coulomb friction and external disturbances, are among the numerous challenges that need to be addressed to successfully control such highly complex nonlinear unstable non-minimum phase systems. Inverted pendulums have a single input multiple output (SIMO) structure and hence, are under-actuated [24]. In other words, the motion of the cart and the angle of the pendulum have to be controlled simultaneously by one single input force.

The destabilizing effect of nonlinear friction has been thoroughly studied in many control systems for high quality servomechanisms. Failing to compensate for modeling uncertainties in controlling such systems can have negative consequences, such as severe tracking errors, limit cycles, chattering, and excessive noise [14, 25]. Many control laws have been proposed for inverted pendulums [1, 2, 26], including classical, robust and adaptive control laws, but they generally consider (structured) parametric uncertainties only. Linear control law design methods [1] are used for their simplicity. However, linearization does not guarantee the stability in all operating conditions. Moreover, they suffer from sensitivity to parameters variation such as, friction and inertia parameters that are often changing dynamically and unknown. In [26], a grey prediction model combined with a proportional derivative (PD) controller is proposed to control a sliding inverted pendulum to swing it up from the downward position to the upright position and guide its slider to the center of the track. A sliding mode controller is proposed in [27]. However, robustness to parameter variations and uncertain disturbances is obtained only when sliding mode truly occurs. In addition, most of these controllers have been proposed with no analytical solution for stability, which has not been given much attention in the literature. A stability analysis has been provided in [14] that copes with modeling and parameters uncertainties. Also, an adaptive robust controller is proposed in [2] for balance and motion control, in the presence of parametric and functional uncertainties. Like in [26], a fuzzy swing-up technique was developed in [28] with a fuzzy sliding balance controller for a planetary gear-type inverted pendulum. In [4], an adaptive neural fuzzy network controller is proposed for solving control problems. In this work, an evolutionary learning method is also proposed to optimize the controller parameters.

The presence of high, particularly unstructured, nonlinearities such as in the form of Coulomb friction significantly changes the system's dynamics [14]. So, modeling the system's dynamics based on presumably accurate mathematical models cannot be applied efficiently in this case. Studies have shown that the design of robust controllers for mathematically ill-defined systems that may be subjected to structured and unstructured uncertainties was made possible with computational intelligence tools, such as artificial neural networks and fuzzy logic controllers [29]. The universal approximation theorem has been the main driving force behind the increasing popularity of such methods as it shows that they 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 intelligent control [30]. Satisfactory performance was achieved with various neural network models for complex systems control [31]. H. Chaoui *et al.* [17] used a sliding mode control approach to learn system's inverse dynamics through a feedforward neural network. A time-delay neurofuzzy network was suggested in [32], where a linear observer is used to estimate the joint velocity signals and eliminated the need to measure them explicitly. Subudhi *et al.* [33] presented a hybrid architecture composed of a neural network to control the slow dynamic subsystem and an H_∞ to control the fast subsystem. Despite the success witnessed by neural network-based control systems, they remain incapable of incorporating any human-like expertise already acquired about the dynamics of the system in hand, which is considered one of the main weaknesses of such soft computing methodologies. On the other hand, fuzzy logic control provides human reasoning capabilities to capture uncertainties. However, their learning ability is almost inexistent as opposed to artificial neural networks. In the last decade, many researchers have put their efforts into combining between the advantages of these two methods. Recently, hybrid control laws gave fuzzy logic controllers more powerful abilities, such as adaptive learning, parallelism, and generalization. Better control performance was achieved by using neural networks to adjust and optimize parameters of fuzzy controllers through offline or online learning. However, stability and robustness analysis of such heuristic methods cannot be easily derived. Therefore, the proposed Lyapunov-based adaptive learning mechanisms are applied to inverted pendulums to highlight their respective performances.

5.4.1 Modeling

The inverted pendulum system depicted in Fig. 5.2 consists of a pendulum mounted on a cart driven along a track by a dc servomotor [24]. As such, the nonlinear dynamics of the pendulum is coupled with the cart dynamics and the friction high nonlinearities of the hinge and the wheels.

Assumption 3 *The body of the inverted pendulum is assumed to be symmetric along y axis.*

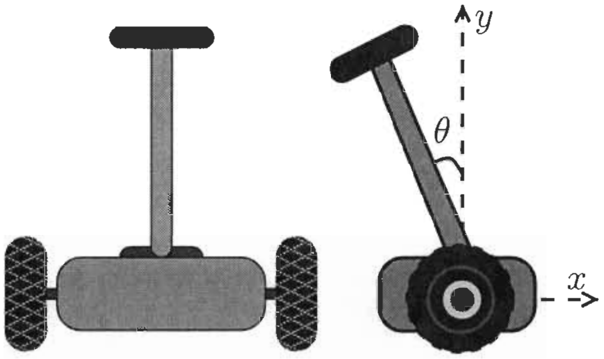


FIGURE 5.2 – Inverted Pendulum System.

The dynamical mathematical model based on Euler-Lagrange formulation can be described by the following equations [24] :

$$J\ddot{\theta} + m_o L\ddot{x}cos\theta - m_o gLsin\theta = -\tau_{F\theta} \tag{5.5a}$$

$$m_t\ddot{x} + m_o L\ddot{\theta}cos\theta - m_o L\dot{\theta}^2 sin\theta = F - F_x \tag{5.5b}$$

where,

m_c	mass of the cart (kg)
m_b	mass of the body (kg)
m_p	mass of the pole (kg)
g	gravitational constant (m/s^2)
L	length of the pole (m)
F_x	friction force of the wheels (N)
$\tau_{F\theta}$	friction torque of the hinge (N $\cdot$ s/m)
F	force applied to the cart (N)
x	position of the cart (m)
$\dot{x}$	velocity of the cart (m/s)
$\ddot{x}$	acceleration of the cart (m/s^2)
θ	angular position of the pole (rad)
$\dot{\theta}$	angular velocity of the pole (rad/s)
$\ddot{\theta}$	angular acceleration of the pole (rad/s^2)

with, $J = \frac{1}{3}m_pL^2 + m_bL^2$, $m_o = \frac{1}{2}m_p + m_b$, and $m_t = m_c + m_b + m_p$.

5.4.2 Friction

The system complexity is increased even further by adopting a highly nonlinear *a priori* unknown friction model that is composed of Coulomb, viscous, and static friction terms [34]. The model of such a memoryless friction operating along a displacement σ is described by

$$F_{friction} = F_c \operatorname{sign}(\dot{\sigma}) + F_v \, \dot{\sigma} + F_s \operatorname{sign}(\dot{\sigma}) \, e^{-(\dot{\sigma}/\eta_s)^2} \tag{5.6}$$

where F_c, F_v and F_s are the Coulomb, viscous and static friction parameters, respectively, η_s is the rate of decay of the static friction term. The term $\text{sign}(\dot{\sigma})$ is defined by

$$\text{sign}(\dot{\sigma}) = \begin{cases} 1 & , \text{if } \dot{\sigma} \geq 0 \\ -1 & , \text{if } \dot{\sigma} < 0 \end{cases}$$

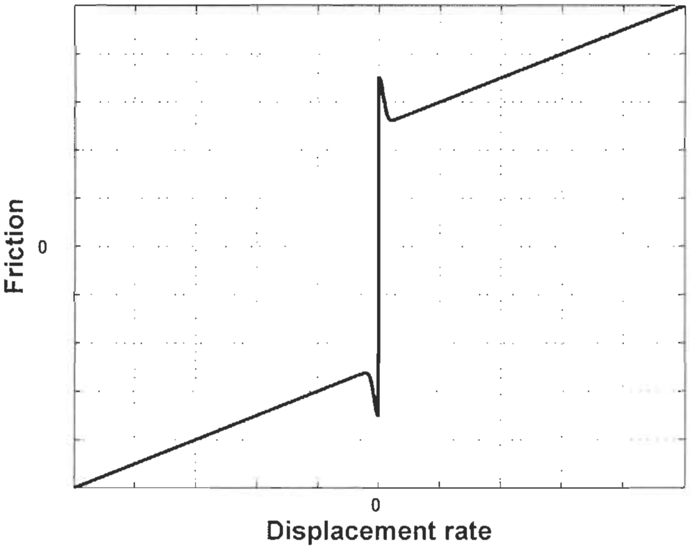


FIGURE 5.3 – Stribeck friction model.

Remark 1 *It is important to point out that although such a friction model results in a drastic increase in the system's nonlinear complexity, it generally provides a more accurate representation of the system's dynamics [35].*

5.4.3 ANN-Based Control of Inverted Pendulums

The control objective is to design an artificial neural network to control the pendulum's motion and balance with unknown dynamics (Fig. 5.4).

Let $e_x = x - x_d$ and $\dot{e}_x = \dot{x} - \dot{x}_d$ denote the inverted pendulum motion position and velocity errors, respectively, and let $e_\theta = \theta - \theta_d$ and $\dot{e}_\theta = \dot{\theta} - \dot{\theta}_d$ denote the inverted pendulum balance

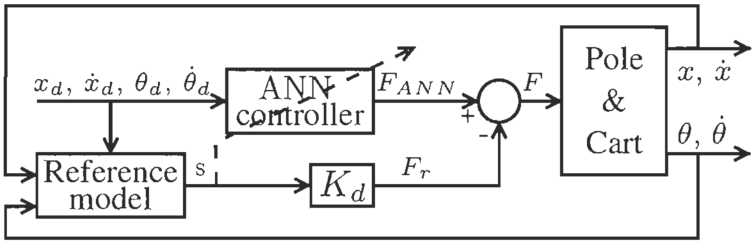


FIGURE 5.4 – Lyapunov-based inverted pendulums neural control scheme

position and velocity errors, respectively. The control objective is to track the errors e_x, $\dot{e}_x$, e_θ, and $\dot{e}_\theta$ to zero. For that, we define the following reference model :

$$s_x = \dot{e}_x + \psi_x\, e_x$$
$$s_\theta = \dot{e}_\theta + \psi_\theta\, e_\theta$$

where ψ_x and ψ_θ are positive constant gains.

Since the system in hand is under-actuated, the controller needs to track both error signals to zero using one signal input force F. It is noteworthy that motion and posture control cannot be achieved at the same time. Hence, a trade-off strategy is needed between the errors s_x and s_θ. Therefore, a compounded error is defined as follows :

$$s = (1 - \lambda)\, s_x + \lambda s_\theta$$

where $0 < \lambda < 1$ defines the trade-off coefficient.

Given the designed signals x_d, $\dot{x}_d$, θ_d, and $\dot{\theta}_d$, a multi-layer perceptron ANN is trained to minimize the error s to achieve motion and balance tracking (Fig. 5.4). Due to the iterative nature of the neural network's learning mechanism and because of the high complexity order of the system's dynamical model, the neural network may take a relatively long time to converge, which may lead to unstable dynamics or unsatisfactory performance. Henceforth, a robustifying term F_r, corresponding to a PD controller, which injects damping into the system, is introduced :

$$F_r = K_d \; s$$

with K_d being a positive constant gain. Therefore, the control law is :

$$F = F_{ANN} - F_r$$

The neural network is composed of three layers each : one input layer of two neurons, one hidden layer with six neurons and one neuron for the output layer. The sigmoid function is used as activation function for all neurons except for the output neuron which uses a linear function. The ANN controller uses the Lyapunov-based learning law described in (5.2).

Setup

To demonstrate the performance of the proposed adaptive control scheme, a set of computer simulation runs is carried out on an inverted pendulum system. Table 5.1 summarizes the system's parameters along with their respective values, where $m_b = 0.05$ kg and $g = 9.8$ m/s^2. The controller gains are set to $\psi_x = 1$ and $\psi_\theta = 4$, which correspond to a time constant of $1s$ for the cart and $0.25s$ for the pole. The parameter λ is set to 0.4 to give the cart a higher weight, and the damping parameter K_d is set to 10.

TABLE 5.1 – Inverted Pendulum's parameters

Parameter	Cart	Pole
Mass	$m_c = 0.22$ kg	$m_p = 0.06$ kg
Length		$L = 0.25$ m
Coulomb friction	$F_{c\alpha} = 7 \cdot 10^{-2}$ N	$F_{c\theta} = 3 \cdot 10^{-3}$ N·m
Viscous friction	$F_{v\alpha} = 2 \cdot 10^{-2}$ N·s/m	$F_{v\theta} = 1 \cdot 10^{-3}$ N·m·s/rad
Static friction	$F_{s\alpha} = 3 \cdot 10^{-2}$ N	$F_{s\theta} = 2 \cdot 10^{-3}$ N·m
Decreasing rate	$\eta_{s\alpha} = 5 \cdot 10^{-2}$ m/s	$\eta_{s\theta} = 4 \cdot 10^{-2}$ rad/s

The desired motion position and velocity signals are taken as the step response of a critically damped second order system with a natural frequency of 1 rad/s, as shown in Fig. 5.5. On the other hand, the desired balance position and velocity signals are taken to be zero, i.e., $(\theta_d = \dot{\theta}_d = 0)$.

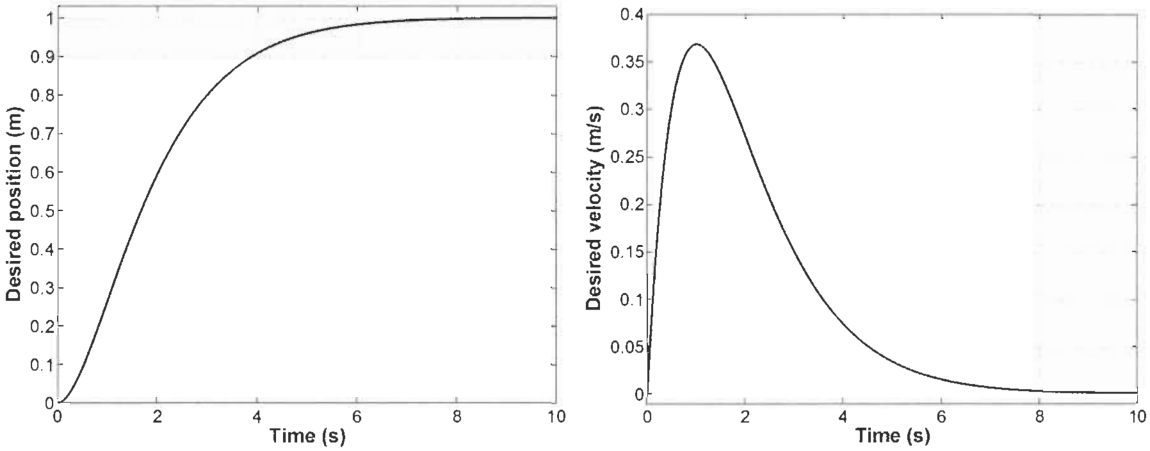
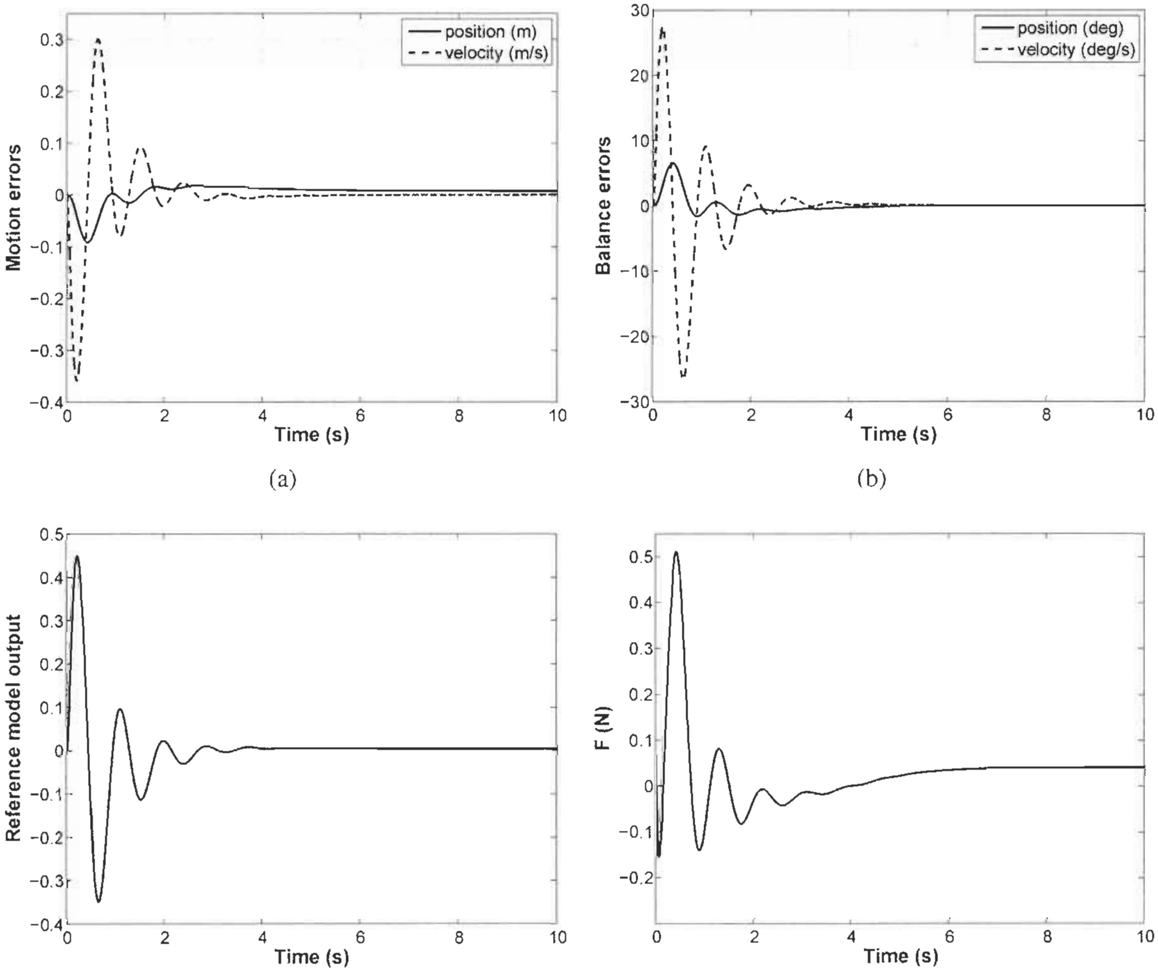


FIGURE 5.5 – Motion position and velocity reference signals

Results

The aforementioned nominal values are used to simulate the system's dynamics without the ANN controller. As shown in Fig. 5.6, the damping term F_r is the key to stabilize the inverted pendulum system as it is able to attenuate the motion and balance tracking errors. However, these errors along with the applied force are fairly oscillatory in this case.

In the second simulation, the ANN controller is introduced to evaluate its ability to better compensate for the system's nonlinearities. As shown in Fig. 5.7, tracking errors converge more steadily to zero with less control effort. It is noteworthy the good trade off achieved between balance and motion control. Furthermore, the controller copes with friction nonlinearities. The next simulation is meant to show the modularity of the proposed controller in compensating for external disturbances. For that, a 0.5 (N) external force step introduced at t = 5s. It is worth pointing out the fact that the introduced external disturbance is not explicitly modeled in the design of the proposed controller. In general, external disturbance may significantly affect the precision of the positioning system and causes unacceptable high frequency oscillations. The controller's performance under such conditions is revealed in Fig. 5.8, the controller is successful in coping with the unexpected force change. Moreover, the motion and balance errors remain small, which yielded a smooth control signal.



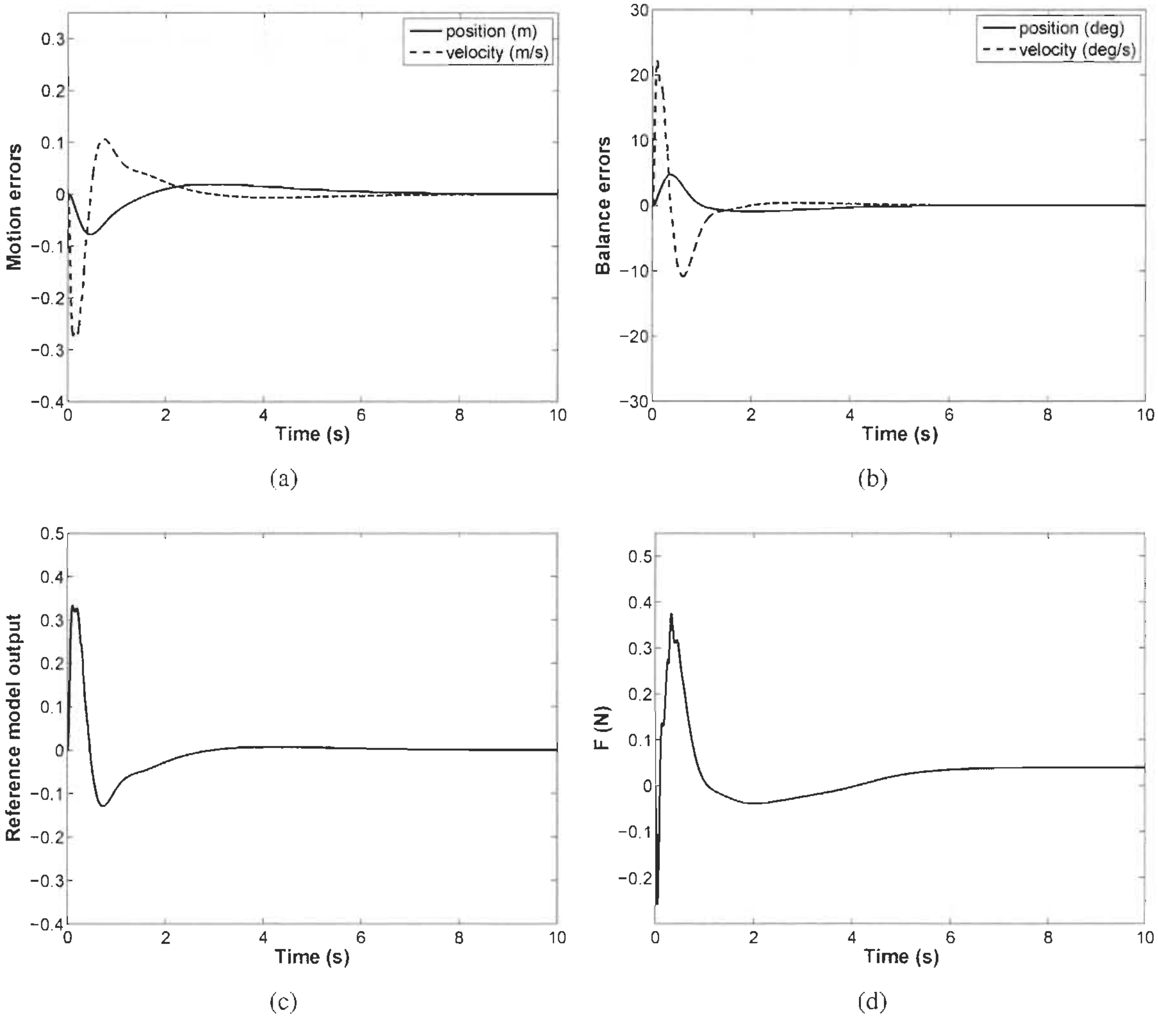
(a)

(b)

(c)

(d)

FIGURE 5.6 – Inverted pendulum response without ANN controller : (a) motion errors ; (b) balance errors ; (c) compounded error ; and (d) control force.



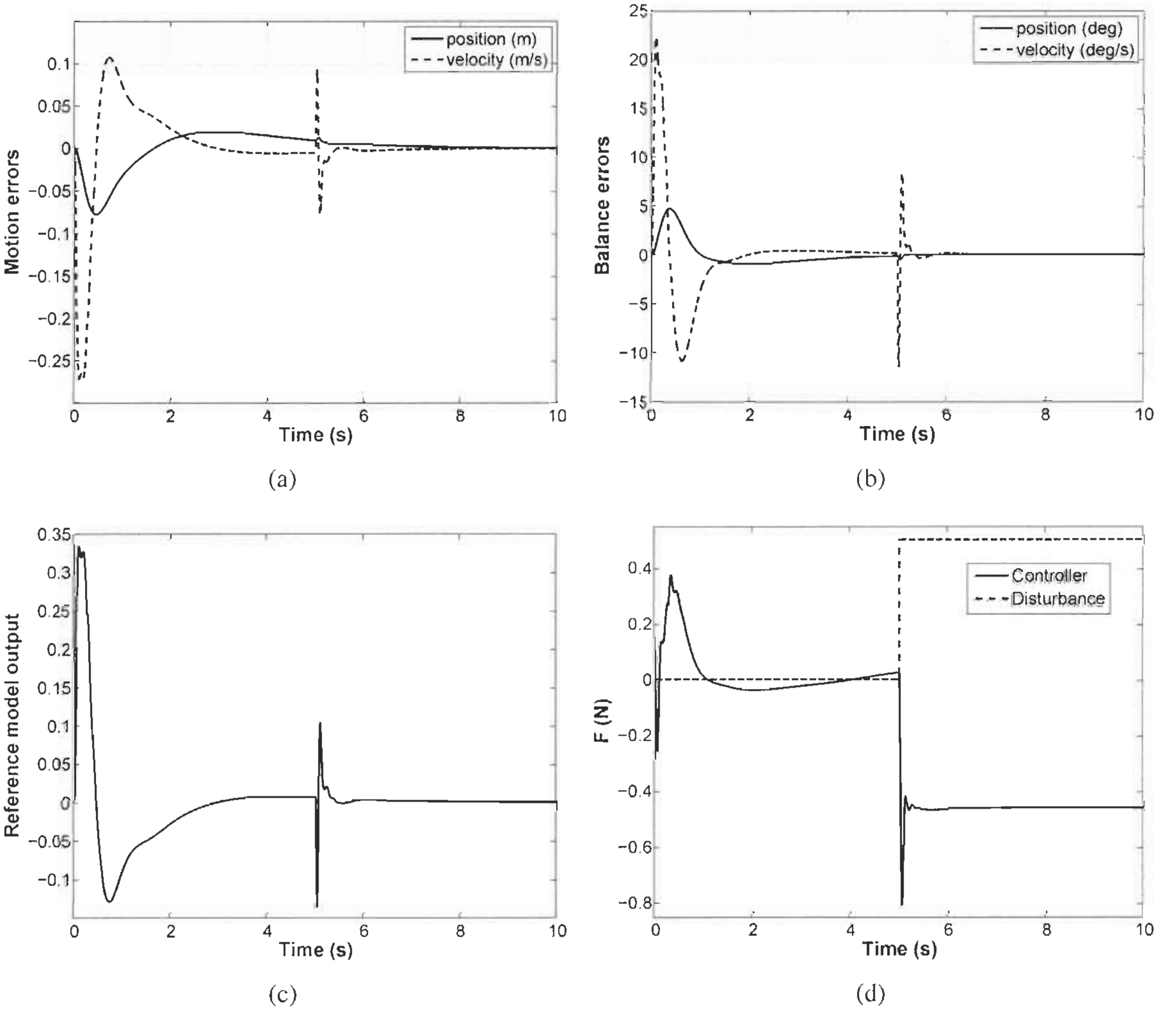
(a)

(b)

(c)

(d)

FIGURE 5.7 – Inverted pendulum response with ANN controller : (a) motion errors ; (b) balance errors ; (c) compounded error ; and (d) control force.



(a)

(b)

(c)

(d)

FIGURE 5.8 – ANN response with disturbance : (a) motion errors ; (b) balance errors ; (c) compounded error ; and (d) control force.

5.4.4 Adaptive Fuzzy Logic Control of Inverted Pendulums

A different strategy is proposed here to obtain the required trade-off between posture and motion control. Two adaptive fuzzy logic controllers FLC$_x$ and FLC$_\theta$ are designed to control the motion along the x axis and the pendulum posture with unknown dynamics (Fig. 5.9).

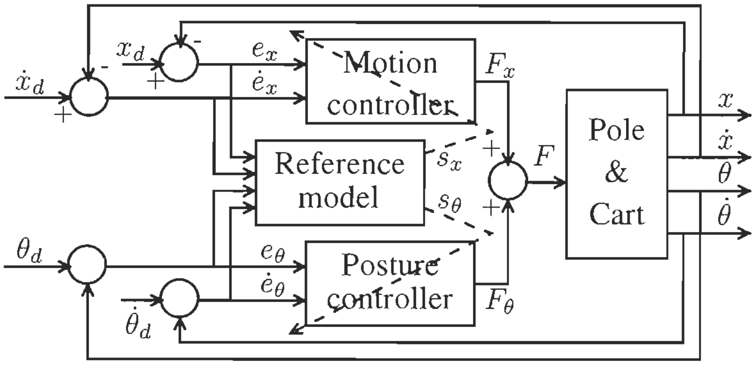


FIGURE 5.9 – Lyapunov-based inverted pendulums fuzzy control scheme

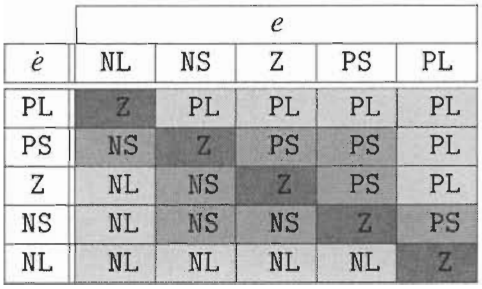
The control objective is to track the errors e_x, $\dot{e}_x$, e_θ, and $\dot{e}_\theta$ to zero. For that, we define the following reference model :

$$s = \dot{e} + \Psi\, e$$

where $s = [s_x, s_\theta]^T$, $e = [e_x, e_\theta]^T$, $\dot{e} = [\dot{e}_x, \dot{e}_\theta]^T$, and $\Psi = \text{diag}(\psi_\alpha, \psi_\theta)$ with $\psi_\bullet$ being a positive constant, that defines the desired bandwidth of the closed loop system.

The fuzzy control strategy is based on a human operator experience to interpret a situation and initiate its control action. A block diagram for the fuzzy controller is illustrated in Fig. 5.9. Given the desired control signals x_d, $\dot{x}_d$, θ_d, and $\dot{\theta}_d$, the motion and posture position and velocity errors e_x, $\dot{e}_x$, e_θ, $\dot{e}_\theta$ are computed. Each FLC takes its two inputs and provides a control action F_x and F_θ. The inputs are quantized into 5 levels represented by a set of linguistic variables : Negative Large (NL), Negative Small (NS), Zero (Z), Positive Small (PS), and Positive Large (PL). The two FLCs are based on the same fuzzy rules, which are represented by Table 5.2. These rules are chosen is such a way as to accomplish the following controller's behavior : (i) when the input signals are far from their respective nominal zero-valued surfaces, then the FLC's output assumes a high value ; (ii) when the inputs are approaching the nominal zero-valued surface, the output is adjusted to a smaller value for a smoother approach ; (iii) once the inputs are close or equal to zero, then the output is set to zero. In this work, we use the "min" and "max" operators as the t-norm and t-conorm, respectively. Triangular membership functions are also used, mainly due to their high computational and performance efficiencies. The membership functions adopted by both controllers are shown in Fig 5.10 and Fig 5.11. Both FLCs use the Lyapunov-based adaptation law described in (5.4).

TABLE 5.2 – Fuzzy logic rules for inverted pendulums



	e				
$\dot{e}$	NL	NS	Z	PS	PL
PL	Z	PL	PL	PL	PL
PS	NS	Z	PS	PS	PL
Z	NL	NS	Z	PS	PL
NS	NL	NS	NS	Z	PS
NL	NL	NL	NL	NL	Z

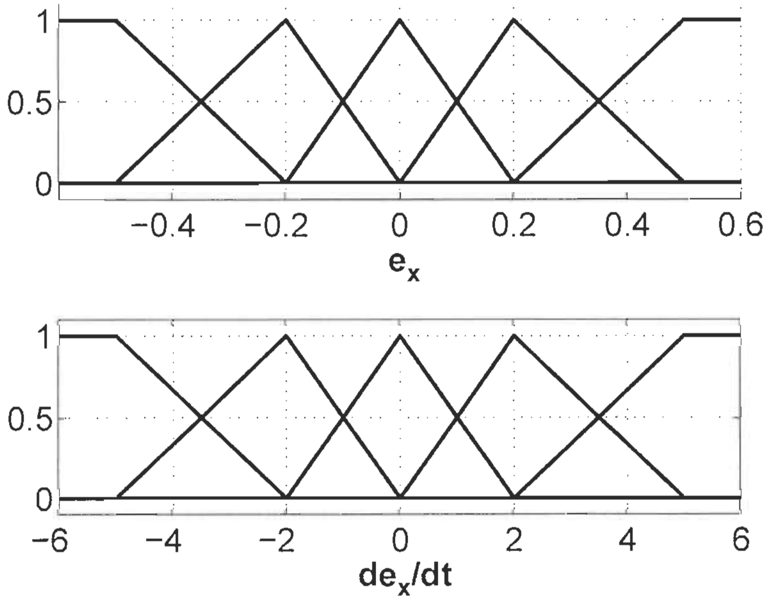


FIGURE 5.10 – Motion controller fuzzy membership functions

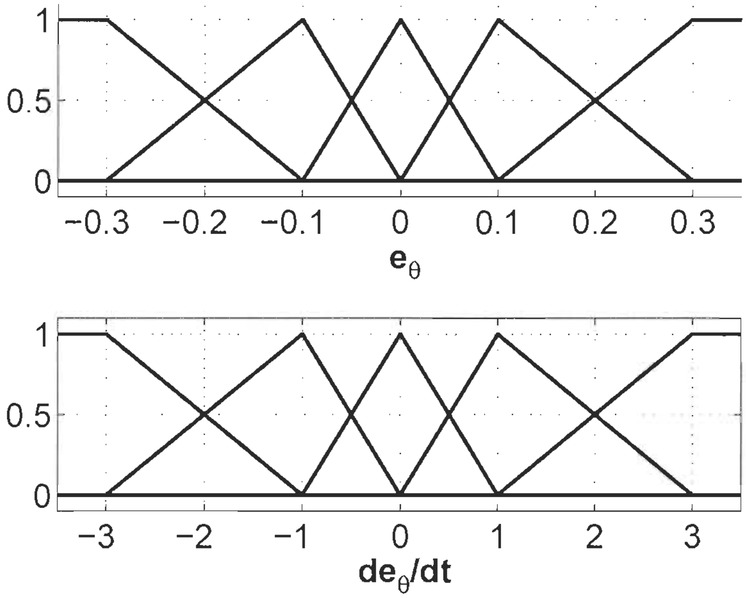
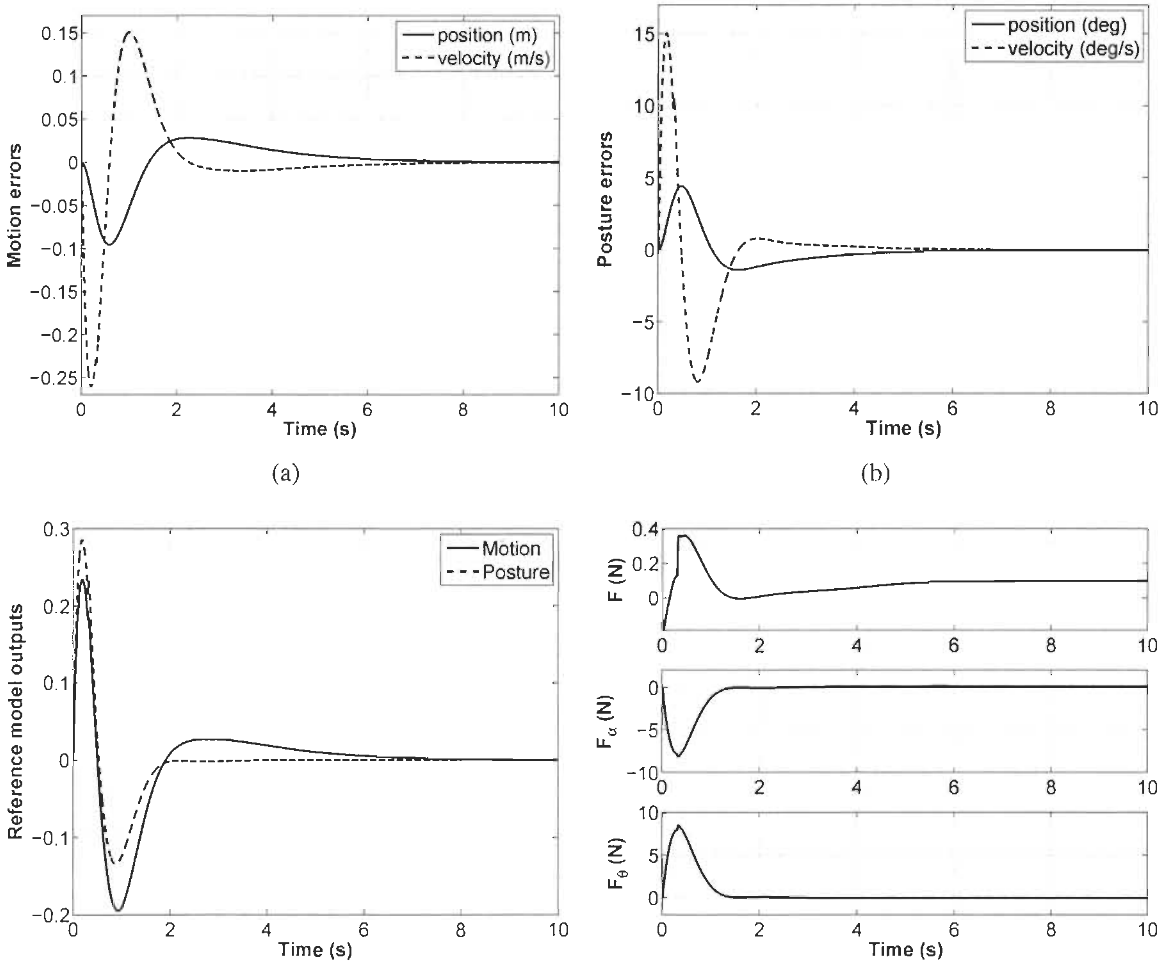


FIGURE 5.11 – Posture controller fuzzy membership functions

Results

Two simulation sets are carried out to study the proposed controller's performance. For each set, the system's response is studied taking into account the pendulum's motion position and velocity errors, the posture position and velocity errors, the reference model outputs (s_x, s_θ), and the control forces F_x, F_θ, and F. In the first simulation, the nominal values in table 5.1 are used to simulate the system's dynamics. As shown in Fig. 5.12, the motion and posture tracking errors decay gradually to zero. On the other hand, smooth control signals are obtained with the proposed adaptive control scheme. Furthermore, the controller copes with friction nonlinearities and achieves fast response. The next simulation is meant to show the modularity of the proposed controller in compensating for external disturbances. For that, a 0.5 (N) external force step is applied to the cart at t = 5s. It is worth pointing out the fact that disturbance is not explicitly modeled in the control design. In general, disturbance affects significantly the precision of positioning systems and causes high frequency oscillations. The controller's performance under such conditions is revealed in Fig. 5.13, the controller is successful in coping with the unexpected force change. Moreover, the motion and posture errors remain small, which yielded smooth control signals. Furthermore, the ability of the proposed adaptive fuzzy controller to cope with uncertainties is clearly shown in this simulation.



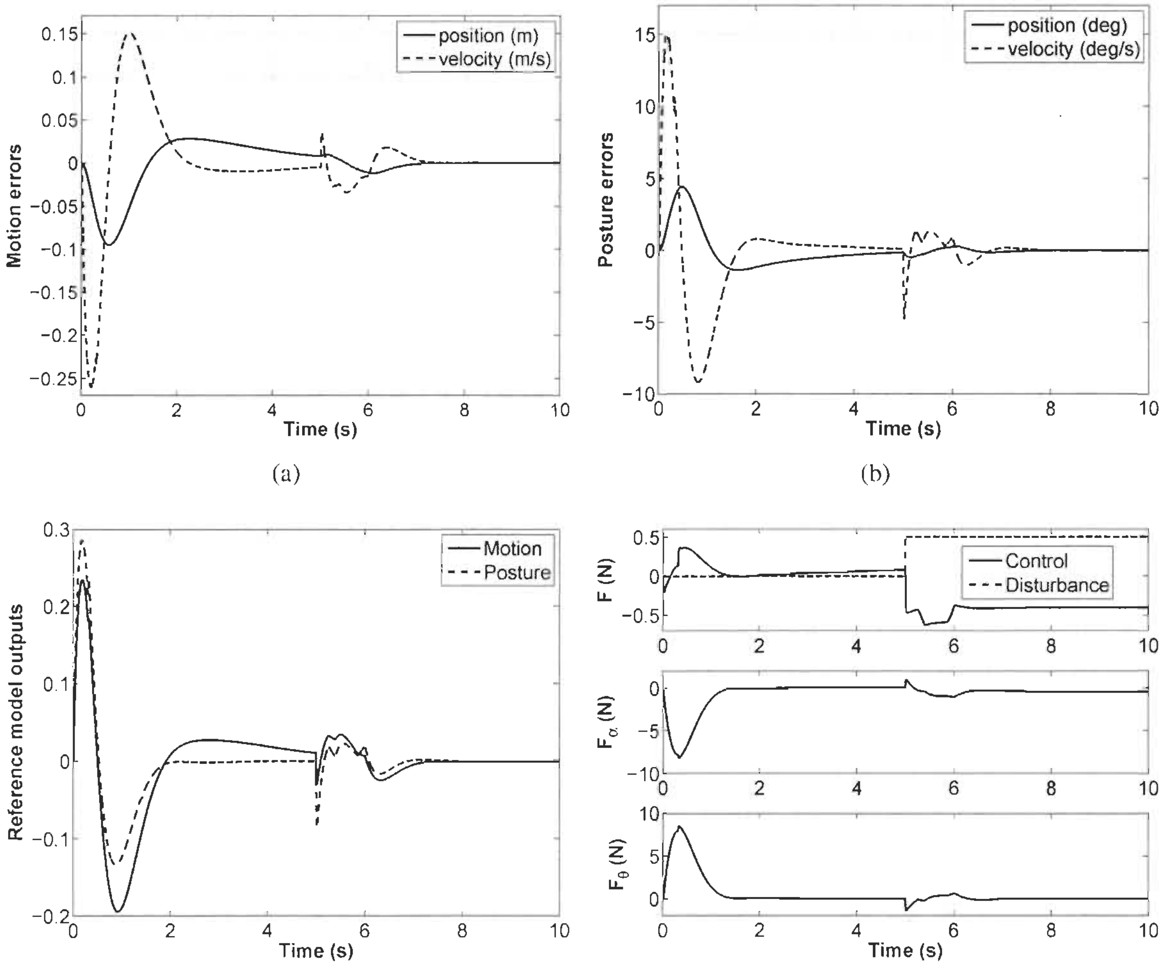
(a)

(b)

(c)

(d)

FIGURE 5.12 – FLC response with nominal values : (a) motion errors ; (b) posture errors ; (c) compounded error ; and (d) control forces F_x, F_θ, and F.



(a)

(b)

(c)

(d)

FIGURE 5.13 – FLC response with disturbance : (a) motion errors ; (b) posture errors ; (c) compounded error ; and (d) control forces F_x, F_θ, and F.

5.5 Hysteresis Compensation for Piezoelectric Actuators

Piezoelectric materials are commonly used as actuators and sensors in many high precision micro and nanopositioning systems, thanks to their ultrafine resolution, light weight, low cost, low thermal losses, high output force, high stiffness, and fast response [36]. These advantages make them good candidates for many applications, such as microrobotics, microassembly, microsurgery, optical fiber alignment, and hard disk drives [37]. However, their efficient operation is limited by the intrinsic hysteresis behavior (Fig. 5.14) in their response to an applied electric field due to the fact that piezoelectric material is ferroelectric [38]. In fact, the electrical charge of a piezoelectric material increases when it is subjected to mechanical stress, which is known as direct piezoelectric effect. Controlling piezoelectric actuators by charge or current instead of voltage is known to significantly reduce the hysteresis effects. But, building charge/current amplifiers capable of driving highly capacitive loads such as piezoelectric actuators is not an easy task to undertake. Moreover, costly material is needed for the induced charge amplification and measurement. Furthermore, the responsiveness and accuracy of the actuator are decreased. This raises the urgency to consider alternative approaches for the control of this type of actuators to keep up with their increasingly demanding design requirements.

The destabilizing effect of hysteresis nonlinearities has been thoroughly studied in many high performance piezoactuated mechanisms [36,37]. The hysteresis characteristics are usually unknown and modeling them can be a difficult task to undertake. Thus, failing to compensate for modeling uncertainties in controlling such systems can have negative consequences, such as severe inaccuracy, limit cycles, chattering, and even instability [38]. Many control laws have been proposed for piezoactuator control problem, including robust and adaptive laws [39,40]. In [36], a robust adaptive controller has been proposed for a piezo-flexural nanopositioning system. In this strategy, a Lyapunov-based controller is designed to cope with hysteresis nonlinearity and parametric uncertainties. It has been also shown that a proportional-integral (PI) controller lacks accuracy in high-frequency tracking and hence, is not suitable for high performance nanopositioning systems. On the other hand, an adaptive sliding mode controller has been introduced for piezoelectric actuators with nonlinear uncertainties [40]. The controller compensates for uncertainties in a sliding mode fashion while the adaptive control part provides the system with more dynamics by coping with residual errors. A stability analysis is also provided. In [39], a radial basis function (RBF) neural network has been used as function approximator for piezoelectric actuation systems where an offline learning procedure is proposed to improve the motion tracking performance. Therefore, the proposed ANN-based learning technique is applied to show its ability to cope with hysteresis effects.

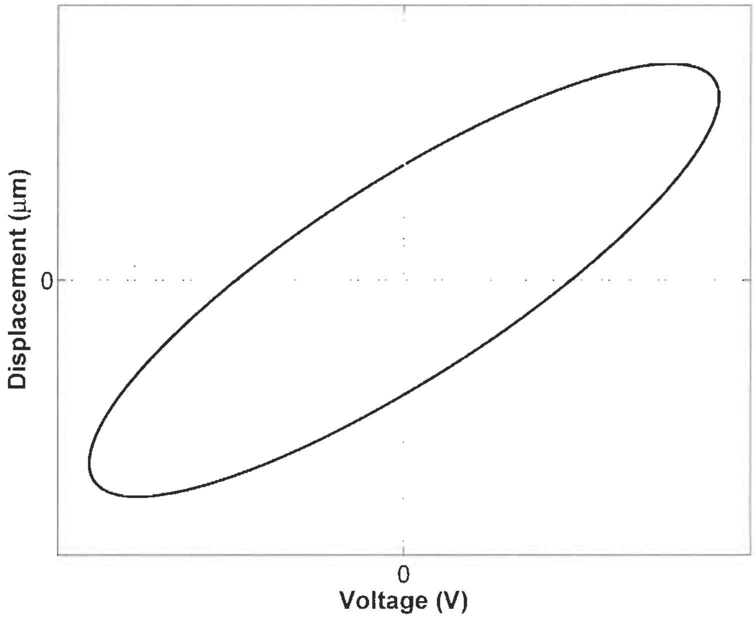


FIGURE 5.14 – Hysteresis characteristic with a sinusoidal applied voltage.

5.5.1 Modeling

A nanopositioning system driven by piezoactuator [39] is depicted in Fig. 5.15. The dynamics of the positioning piezomechanism can be described by the following differential equation coupled with hysteresis in the presence of external disturbance :

$$m\ddot{x} + b\dot{x} + kx + d = T_{em}(V_a - V_h) \tag{5.7a}$$

$$\dot{V}_h = k_1\dot{V}_a - k_2 x|\dot{V}_a|V_h - k_3\dot{V}_a|V_h| \tag{5.7b}$$

where,

m	mass of the piezoactuator
b	viscous friction coefficient
k	stiffness coefficient
d	disturbance
V_a	applied voltage
V_h	hysteresis effect
T_{em}	voltage-to-force coefficient
$k_\bullet$	piezoelectric coefficients

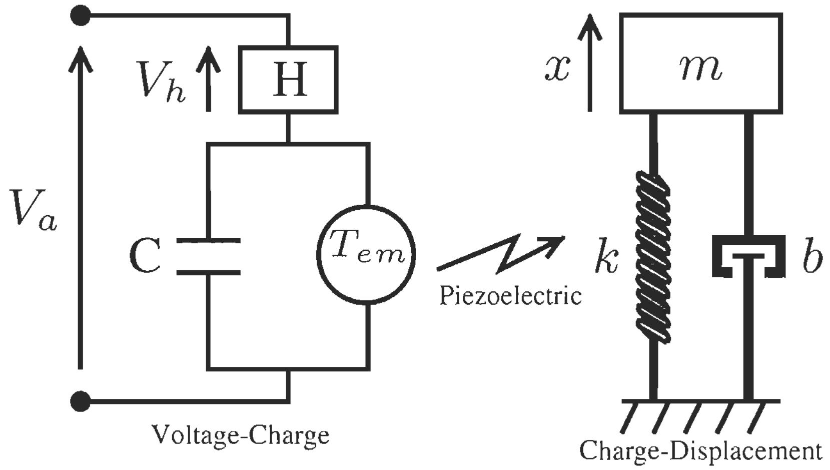


FIGURE 5.15 – Piezoelectric actuator model.

5.5.2 ANN-Based Control of Piezoelectric Actuators

Let $e = x - x_d$ and $\dot{e} = \dot{x} - \dot{x}_d$ denote the system's position and velocity errors, respectively. The control objective is to track these errors to zero. For that, we define the following reference model :

$$s = \dot{e} + \psi\, e$$

where ψ being a positive constant that defines the desired bandwidth of the closed loop system.

Given the desired signals x_d and $\dot{x}_d$, a multi-layer perceptron ANN is trained to minimize the error s to achieve motion tracking. The neural network is composed of three layers each : one input layer of two neurons, one hidden layer with six neurons and one neuron for the output layer. The sigmoid function is used as activation function for all neurons except for the output neuron which uses a linear function. The proposed online ANN-based Lyapunov learning strategy (5.2) is used. The control structure is shown in Fig. 5.16.

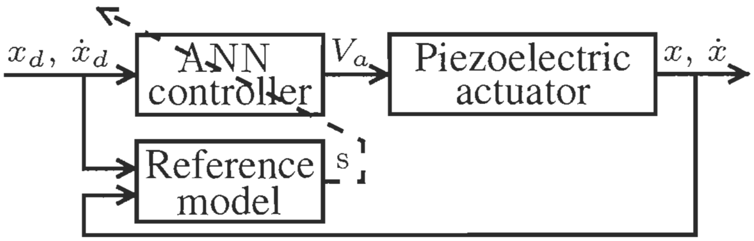


FIGURE 5.16 – Lyapunov-based piezoelectric actuators neural control scheme

Setup

To demonstrate the performance of the proposed controller, a set of simulation runs is carried out on a nanopositioning piezoactuated system. The piezoactuator's parameters along with their respective values are set as follows : $m = 0.016$ (kg), $b = 1$ (N·s/m), $k = 1.5 \cdot 10^{-6}$ (N/m), $T_{em} = 1.152$ (N/V), $k_1 = 3.5$, $k_2 = 0.6$, and $k_3 = 0.5$. The desired position trajectory is taken as the step response of a critically damped second order system with a natural frequency $w_n = 100$ rad/s, as shown in Fig. 5.17.

Results

Two simulation runs are carried out to study the performance of the proposed control scheme. For each simulation, the system's response is studied taking into account the piezoac-

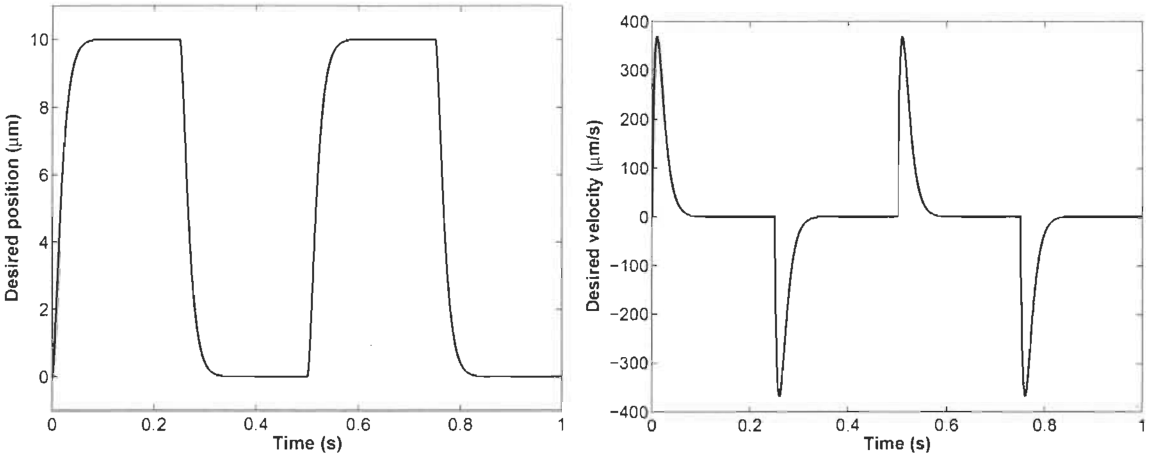


FIGURE 5.17 – Position and velocity reference signals

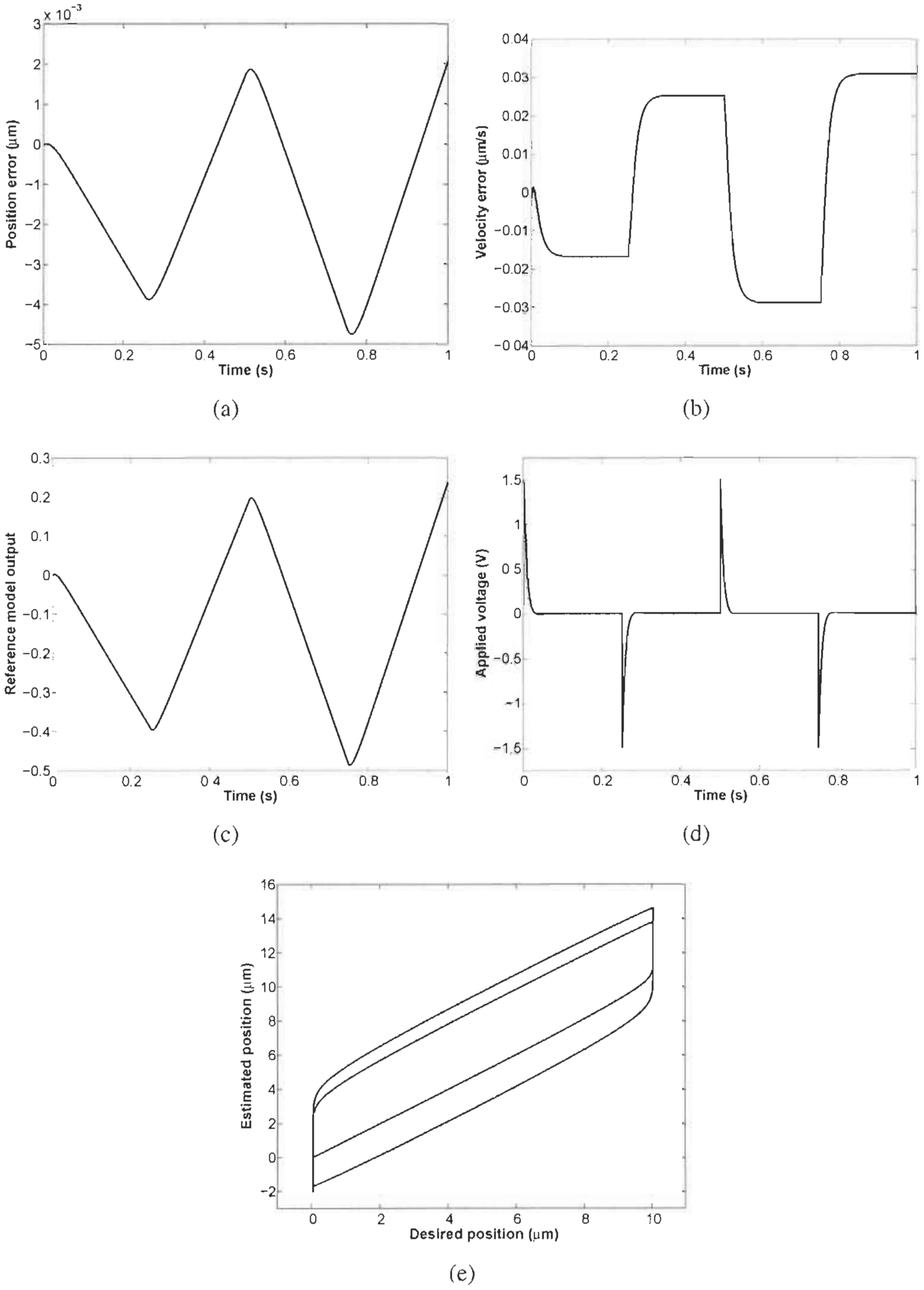
tuator's position and velocity errors, the reference model signal s, and the applied voltage V_a.

In the first simulation, the aforementioned nominal values are used to simulate the piezoactuator's dynamics and its exact inverse model is used without hysteresis compensation to show its negative consequence. As shown in Fig. 5.18, the hysteresis effect results in severe position and velocity tracking errors in spite of the smooth applied control voltage. On the other hand, the hysteresis loop obtained in Fig. 5.18(e) shows the need of hysteresis compensation to cope with its effect.

In the second simulation, the proposed controller is used to show its ability to compensate for hysteresis effect and hence achieve better tracking than simulation 1. As shown in Fig. 5.19, the position and velocity tracking errors decay gradually before stabilizing within a negligible amplitude. Moreover, the proposed controller copes with the hysteresis effect while providing smooth control voltage.

5.6 Conclusion

This chapter presents modeling and control of inverted pendulums. The control approach is designed for general nonlinear MIMO systems. In this strategy, the control scheme takes advantage of learning and generalization capabilities to ensure precise tracking by cop-



(a)

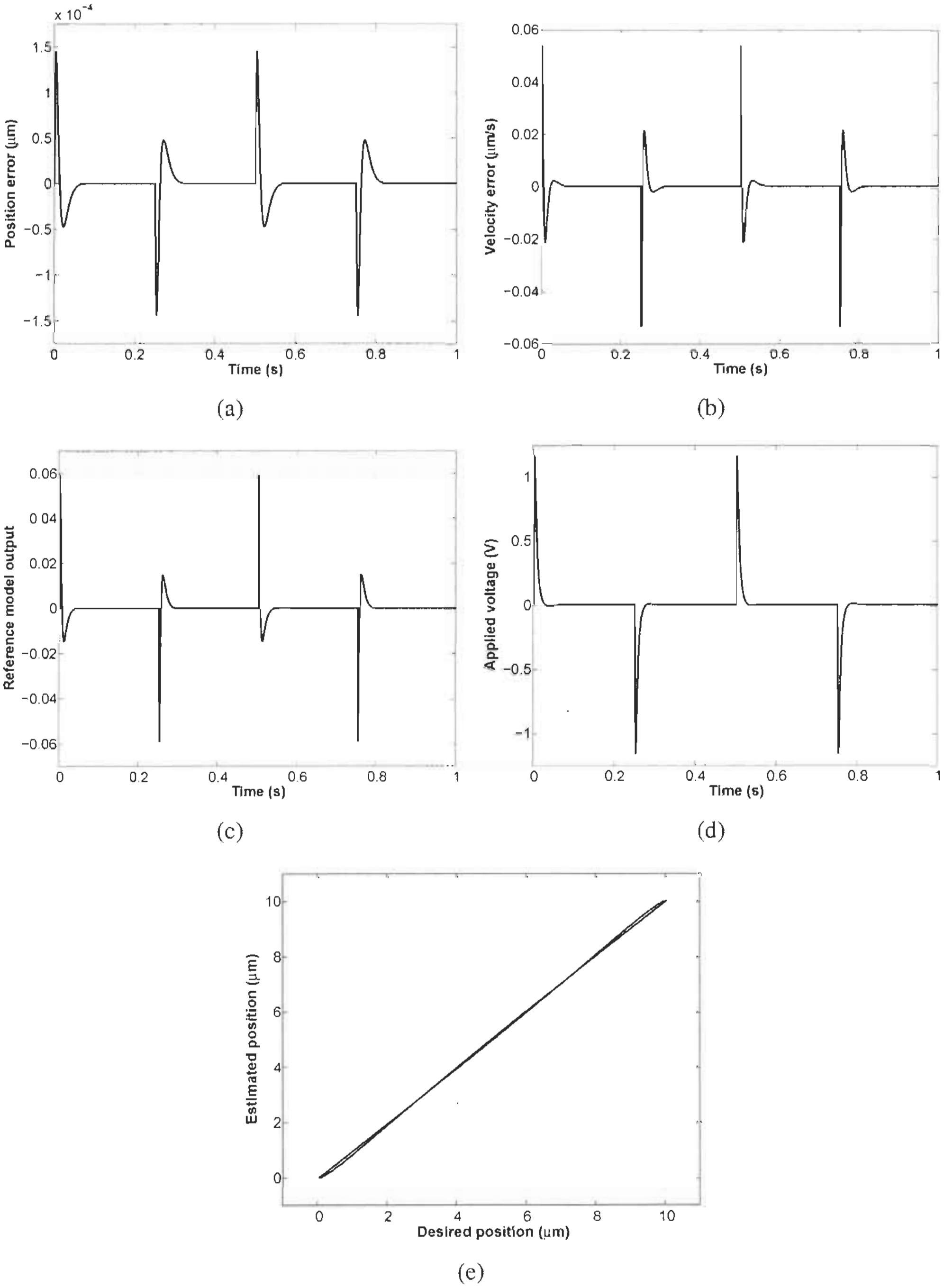
(b)

(c)

(d)

(e)

FIGURE 5.18 – Piezoactuator response without hysteresis compensation : (a) position error ; (b) velocity error ; (c) reference model ; (d) applied voltage ; and (e) hysteresis loop.



(a)

(b)

(c)

(d)

(e)

FIGURE 5.19 – Piezoactuator response with hysteresis compensation : (a) position error ; (b) velocity error ; (c) reference model ; (d) applied voltage ; and (e) hysteresis loop.

ing with unstructured uncertainties. Adaptive neural and fuzzy based adaptive control strategies have been applied for inverted pendulum motion and balance control problem. Simulation results showed good performance in the presence of friction nonlinearities and external disturbance.

The ANN-based learning strategy is also applied for high performance nanopositioning systems to cope with hysteresis effects. Simulations showed that the piezoactuator's speed and position can be tracked to a high precision scale. The proposed controller demonstrates its effectiveness in dealing with this burdensome hysteresis effect. Unlike other controllers, no *a priori* offline training or weights initialization knowledge is required. Furthermore, ANN capabilities are a key to achieve high control accuracy needed for high performance nanopositioning systems.

The problem of hysteresis within piezoelectric actuators is similar to the effect of joints elasticity in robotics. Thus, similar adaptive control techniques can also be applied. Therefore, next chapter presents few intelligent adaptive control strategies for flexible-joint manipulators. Thus far, the proposed Lyapunov-based neural and fuzzy learning techniques have been proposed for nonlinear systems without taking into account their dynamics. The stability of these universal adaptation laws is studied using Lyapunov stability theory. However, some nonlinear systems introduce additional constraints on the control stability and therefore, it is important to consider a system's dynamics when studying its stability.

Chapitre 6

Control of Robotic Manipulators with Elasticity, Friction, and Disturbance

6.1 Introduction

Flexible-joint and flexible-link manipulators offer several advantages with respect to their rigid counterpart such as, light weight, lower cost, smaller actuators, larger work volume, better maneuverability and transportability, higher operational speed, power efficiency, and larger number of applications. Thus, they are often required to operate at high speed to yield high productivity. The conflicting requirements between high speed and high accuracy make the robotic control task a challenging research problem. Reducing the weight of the arms and/or increasing the operation speed make many industrial flexible-joint manipulators face arm vibration problems, particularly in high speed motion, because of the low stiffness. This can be resolved by increasing the stiffness. However, it increases the mass, depleting the advantages listed above. The friction severe nonlinearities, coupling stemming from the manipulator's flexibility, varying operating conditions, structured and unstructured dynamical uncertainties, and external disturbances, are among the typical challenges to be faced.

Flexibility and nonlinear friction may have some destabilizing effects when failing to compensate for modeling uncertainties in controlling flexible structures. These phenomena have been thoroughly studied in many control systems for high quality servomechanisms. Several studies show these negative consequences, such as severe tracking errors, limit cycles, chattering, and excessive noise [35, 41]. Many control laws have been proposed for flexible joints [34, 35, 42], including classical, robust and adaptive control laws, using techniques such as singular perturbations and energy methods [43], but they generally consider (structured) parametric uncertainties only. Several models and compensation schemes have been proposed. Adaptive control techniques have been regarded among the most rewarding research avenues for such type of problems. However, not much has been achieved yet for systems that exhibit both flexibility and severe nonlinearities.

Flexible-joint manipulators are governed by complex dynamics and hence controlling them depends on their dynamic models. There are many modeling techniques for mechanical systems, such as, Lagrangian approach, Hamilton's principle, and Kane method. Yet, the system is inevitably subjected to the ubiquitous presence of high, particularly unstructured, modeling nonlinearities, such as Coulomb friction and external disturbances, for instance. The presence of such uncertainties on a manipulator driven through a flexible joint significantly changes the system's dynamics as opposed to when the load is driven with a rigid joint [41, 44]. In this case, solving the inverse dynamics of the system is not realizable since the motor position is not uniquely defined at standstill. This last condition also illustrates that the actuator's state cannot be observed continuously from the load output. Henceforth, only an approximate inverse model can be realized. So, modeling the system's dynamics based on presumably accurate mathematical models cannot be applied efficiently in this case. This raises the urgency to consider alternative approaches for the control of this type of manipulator systems to keep up with their increasingly demanding design requirements.

Various control techniques were proposed over the years to control flexible-joint manipulators [45, 46]. De Luca *et al.* [47] and Khorasani [48] proposed feedback linearization-based controllers. However, it depends on excessively noisy joint acceleration and jerk signal measurements and hence unreliable in most real-life robotic systems. On the other hand, C. de Wit [49] proposed a robust control scheme for friction compensation due to uncertainties in friction models. Even as such, the suggested controllers require the full *a priori* knowledge of the system's dynamics. This problem has been partially overcome by several adaptive control schemes [50–52]. Most of these control techniques capitalize on the singular perturbation theory to extend the adaptive control theory developed for rigid bodies to flexible ones [5,46,53,54]. M. Spong [46] reduced flexible-joint manipulators model to standard rigid manipulators model as the joint stiffness tends to infinity. This model has been widely used by many researchers to achieve better tracking performance. For example, F. Ghorbel *et al.* [43] used a rigid manipulator's conventional method as slow controller and a fast feedback control law is used to damp out the oscillations of the joint flexibility modes. In a similar way, K. Khorasani *et al.* [55] illustrated how standard adaptive control schemes for rigid robots may be generalized for flexible-joint manipulators under a certain set of assumptions. Although many of these controllers are shown to be quite performant in theory, they failed to address important issues that might stand against their practical implementation, like basing the control laws on joint torques and their derivative [56,57], for instance, which are well known to be extremely noisy in real-life applications.

Moreover, such type of control algorithms uses online continuous estimation through well-defined adaptation laws of a set of the plant's physical parameters to approximate the system's dynamics. 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 possibly to minor unstructured uncertainties. Moreover, the unknown physical parameters must have constant or slowly varying nominal values. An explicit linear parametrization of the uncertain dynamics parameters must also exist, and even if it does, it might not be trivial to derive, especially with complex dynamic systems. Although the latter condition is guaranteed for robotic systems, it might not be the case for many other dynamic models. Although some conventional adaptive control techniques, proposed in the literature, did indeed tackle external disturbance attenuation, in addition to the compensation for parametric uncertainties, they did not take into consideration the effects of modeling uncertainties [50].

On the other hand, computational intelligence tools, such as artificial neural networks and fuzzy logic controllers, 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 [29,58]. The universal approximation theorem has been the main driving force behind the increasing popularity of such methods as it shows that they are theoretically capable of uniformly approximating any continuous real function to any degree of accuracy. This has led to recent advances in the area of intelligent control [30,59]. Various neural network and fuzzy logic models have been applied in the control of flexible joint manipulators, which have led to a satisfactory performance [17,60]. H. Chaoui *et al.* [31,61] used a neural network based adaptive control approach inspired by sliding mode control to learn the system's dynamics. A time-delay neurofuzzy network was suggested in [32], where a linear observer was used to estimate the joint velocity signals and eliminate the need to measure them explicitly. Subudhi *et al.* [33] presented a hybrid architecture composed of a neural network to control the slow dynamic subsystem and an H_∞ to control the fast subsystem. A feedback linearization technique using a Takagi-Sugeno neuro-fuzzy engine was adopted in [12]. Despite the success witnessed by neural network-based control systems, they remain incapable of incorporating any human-like expertise already acquired about the dynamics of the system in hand, which is considered one of the main weaknesses of such soft-computing methodologies.

Motivated by the aforementioned challenges, this chapter presents an overview of classical adaptive control theory for rigid manipulators. Then, several advanced intelligent controllers are developed for flexible-joint manipulators to cope with different types of uncertainties.

6.2 Modeling

6.2.1 Rigid Manipulators

A rigid robotic manipulator can be modeled by an actuator coupled directly to the load. Consider a robot manipulator with n revolute joints. Using Euler-Lagrange formulation, the dynamic equations of the manipulator can be written as

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)=\tau \tag{6.1}$$

where,

$M(q) \in \mathbb{R}^{n\times n}$: manipulator's positive definite inertial matrix

$C(q,\dot{q}) \in \mathbb{R}^{n\times n}$: Coriolis and centripetal matrix

$G(q) \in \mathbb{R}^{n}$: vector of gravitational torques

$q \in \mathbb{R}^{n}$: vector of links' positions

$\dot{q} \in \mathbb{R}^{n}$: vector of links' velocities

$\tau \in \mathbb{R}^{n}$: actuators' generalized torque vector (control input)

The dynamics of a robotic manipulator is characterized by the following properties :

Property 1 *The inertia matrix $M(q)$ is,*

1) Positive Definite Symmetric (PDS), i.e., $M^T(q)=M(q)$ and $x^T M(q)\,x>0$ for any non-null vector x.

2) Upper and lower bounded, i.e., there exists two scalars $\alpha_1(q)$ and $\alpha_2(q)$ such that $\alpha_1(q)I \leq M(q) \leq \alpha_2(q)I$, where I is the identity matrix.

Property 2 *Assuming that the Coriolis and centripetal matrix $C(q,\dot{q})$ is defined in terms of Christoffel symbols, the matrix $C(q,\dot{q})$ has the following properties.*

1) Matrix $\dot{M}(q)-2C(q,\dot{q})$ is skew symmetric, i.e.,

$$x^T\left(\dot{M}(q)-2C(q,\dot{q})\right)x=0 \quad \forall\, x\in\mathbb{R}^n$$

2) $C(q,\dot{q})\dot{q}$ is quadratic in $\dot{q}$ and bounded, i.e., there exists a scalar vector $\alpha_3(q)$ such that $\|C(q,\dot{q})\| \leq \alpha_3(q)\|\dot{q}\|$ or equivalently $\|C(q,\dot{q})\dot{q}\| \leq \alpha_3(q)\|\dot{q}\|^2$.

Property 3 *The gravity vector $G(q)$ is bounded, i.e., $\|G(q)\| \leq \alpha_4(q)$, for a scalar vector $\alpha_4(q)$.*

6.2.2 Flexible-Joint Manipulators

The schematic representation for the *i*th flexible-joint in a multi-joint manipulator is shown in Fig. 6.1. The actuator is coupled to a flexible transmission through an $r:1$ reduction gear. The transmission is dynamically simplified as a linear torsional spring linked directly to the load (e.g., manipulator link.)

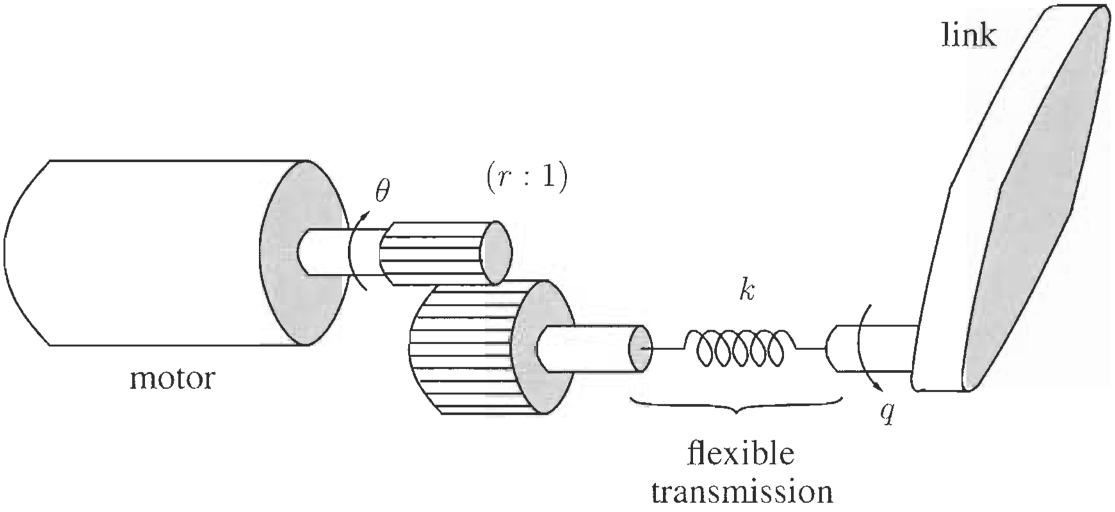


FIGURE 6.1 – Flexible-joint model.

Consider a robot manipulator with n revolute flexible joints. Using Euler-Lagrange formulation, the dynamic equations of the manipulator can be written as :

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)=\tau_t-\tau_{fl}-\tau_{dl} \tag{6.2a}$$

$$J_m\ddot{\theta}=\tau_m-\frac{1}{r}\tau_t-\tau_{fm}-\tau_{dm} \tag{6.2b}$$

$$\tau_t=K(\frac{\theta}{r}-q) \tag{6.2c}$$

where,

$q\in\mathbb{R}^n$: vector of links' positions

$\theta \in \mathbb{R}^n$: vector of motors' positions

$M(q) \in \mathbb{R}^{n\times n}$: manipulator's positive definite inertial matrix

$C(q,\dot{q}) \in \mathbb{R}^{n\times n}$: matrix of Coriolis and centrifugal terms

$G(q) \in \mathbb{R}^n$: vector of gravitational torques

$J_m \in \mathbb{R}^{n\times n}$: motors' diagonal inertial matrix

$\tau_t \in \mathbb{R}^n$: vector of transmission torques

$\tau_m \in \mathbb{R}^n$: motors' generalized torque vector (control input)

$\tau_{fl} \in \mathbb{R}^n$: load friction torque vector

$\tau_{fm} \in \mathbb{R}^n$: motors' friction torque vector

$\tau_{dl} \in \mathbb{R}^n$: load's unmodeled dynamics and external disturbance torque vector

$\tau_{dm} \in \mathbb{R}^n$: motors' unmodeled dynamics and external disturbance torque vector

$K \in \mathbb{R}^{n\times n}$: diagonal matrix of joints' stiffness coefficients

$r \in \mathbb{R}$: gear ratio

Before we proceed further, we introduce the following practical assumption.

Assumption 4 *The norm of the unknown disturbance τ_d is upper bounded by a scalar b_d, i.e., $||\tau_d|| \leq b_d$.*

6.3 Adaptive Control for Rigid Manipulators

Over the years, many adaptive control strategies have been proposed for rigid manipulators. These developments represent an important step towards high precision robotic applications. This section presents one of the most popular adaptive techniques [22], which is used in this thesis as a starting point for the design of robust adaptive control techniques able to cope with different types of uncertainties.

Let $e_q = q - q_d$ and $\dot{e}_q = \dot{q} - \dot{q}_d$ denote the manipulator position and velocity errors, respectively, with q_d and $\dot{q}_d$ being the desired time-dependent position and velocity vectors. Define the following error signal :

$$s = \dot{e}_q + \Psi e_q = \dot{q} - \dot{q}_r \tag{6.3}$$

$$\dot{q}_r = \dot{q}_d - \Psi e_q$$

where $\Psi = \text{diag}(\psi_1, \psi_2, \ldots, \psi_n)$, with ψ_i a positive constant, $i = 1, \ldots, n$. Recall Euler-Lagrange formulation :

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau$$

This equation satisfies the following linear regression :

$$M(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r + G(q) = \Phi(\ddot{q}_r, \dot{q}_r, \dot{q}, q)^T W \tag{6.4}$$

where, $\Phi \in \mathbb{R}^{n \times m}$ is a matrix of known functions (regressor), and W is a m-dimensional vector of parameters. The control law is :

$$\tau = \Phi^T \hat{W} - K_D s \tag{6.5}$$

where, K_D is a positive diagonal matrix gain and the symbol $\hat{\bullet}$ denotes the parameter estimate vector.

Theorem 9 *Consider the nonlinear system in (6.1) under the assumption of high stiffness so that singular perturbation applies with reference signal (6.3) and control law (6.5). The adaptive control law is asymptotically stable and the tracking error converges to zero with the following adaptation law :*

$$\dot{\hat{W}} = -\Gamma\, \Phi\, s$$

where $\Gamma = diag(\gamma_1, \gamma_2, \ldots, \gamma_n)$ and γ_i is a positive constant, $i = 1, \ldots, n$. Therefore, the manipulator's position q and velocity $\dot{q}$ converge to their pre-defined time-dependent desired values q_d and $\dot{q}_d$, respectively.

Proof 3 *Take the derivative of (6.3) :*

$$\dot{s} = \ddot{q} - \ddot{q}_r$$

$$M(q)\dot{s} = M(q)\ddot{q} - M(q)\ddot{q}_r$$

Substituting $M(q)\ddot{q}$ from (6.1),

$$M(q)\dot{s} = \tau - M(q)\ddot{q}_r - C(q,\dot{q})\dot{q}_r - G(q) - C(q,\dot{q})s$$

The linear in parameters property yields,

$$M(q)\dot{s} = \tau - \Phi^T W - C(q,\dot{q})s \tag{6.6}$$

Set

$$\tau = \Phi^T \hat{W} - K_D s \tag{6.7}$$

Equation (6.6) becomes,

$$M(q)\dot{s} = \Phi^T \tilde{W} - C(q,\dot{q})s - K_D s \tag{6.8}$$

where, $\tilde{W} = \hat{W} - W$.

Choose the following Lyapunov candidate function :

$$V = \frac{1}{2}\{\ s^T M(q)s\ +\ \tilde{W}^T\ \Gamma^{-1}\ \tilde{W}\ \}$$

Taking the time-derivative of V :

$$\dot{V} = s^T M(q)\dot{s} + \frac{1}{2}s^T \dot{M}(q)s + \tilde{W}^T \Gamma^{-1} \dot{\tilde{W}}$$

Since the unknown parameters W are assumed to be constant, $\dot{\tilde{W}} = \dot{\hat{W}}$.

Substituting $M(q)\dot{s}$ from (6.8),

$$\dot{V} = s^T \Phi^T \tilde{W} + s^T \{\frac{1}{2}(\dot{M}(q) - 2C(q,\dot{q}))\} s + \tilde{W}^T \Gamma^{-1} \dot{\hat{W}} - s^T K_D s$$

$s^T \{(\dot{M}(q) - 2C(q,\dot{q}))\} s = 0$ *due to the skew-symmetry property. Hence,*

$$\dot{V} = s^T \Phi^T \tilde{W} + \tilde{W}^T \Gamma^{-1} \dot{\hat{W}} - s^T K_D s$$

Setting the adaptation law as

$$\dot{\hat{W}} = -\Gamma \, \Phi \, s$$

leads to

$$\dot{V} = -s^T K_D \, s \leq 0$$

Therefore V ; and so s, $\tilde{W}$, and $\hat{W}$, are bounded and converge to finite values. It implies from (6.3) that e_q, $\dot{e}_q$, and the manipulator's states q, and $\dot{q}$ are bounded. Hence, $\dot{q}_r$ and $\ddot{q}_r$ are also bounded. Using (6.4) and the manipulator's properties, we state that the regressor matrix Φ is bounded. From (6.5), τ is bounded, which implies from (6.6) that $\dot{s}$ is bounded. Thus, $\ddot{V} = -2s^T K_D \dot{s}$ is also bounded. Hence, from Barbalat's Lemma, it implies that $\lim_{t\to\infty} \dot{V} = 0$*. Therefore,* $\lim_{t\to\infty} s = 0$*. Since $\dot{s}$ is bounded, $\ddot{e}_q$ is also bounded. Thus, Barbalat's Lemma also shows that* $\lim_{t\to\infty} \dot{e}_q = 0$ *and so* $\lim_{t\to\infty} e_q = 0$*. Therefore, the manipulator's position q and velocity $\dot{q}$ converge to their pre-defined time-dependent desired values q_d and $\dot{q}_d$, respectively, i.e.,* $\lim_{t\to\infty} q = q_d$ *and* $\lim_{t\to\infty} \dot{q} = \dot{q}_d$.

6.3.1 Setup

To demonstrate the performance of the proposed controller, a set of computer simulation runs is carried out on a planar manipulator (Fig. 6.2). Both links have the same length, i.e., $l = l_1 = l_2 = 1$ m and their weight is $m_1 = 1$ kg, $m_2 = 2$ kg, respectively and the matrix $G(q)$ is taken to be equal to zero. The manipulator's dynamics is defined by (6.9), where $g = 9.8$ m/s^2 is the gravitational constant. The controller is set to obtain a bandwidth of 100 Hz. The regression matrix Φ is given by (6.10) while the parameter vector $W = [m_1 l^2 \; m_2 l^2]$. Its estimate vector is initialized to $\hat{W} = [1.5 \; 3]$.

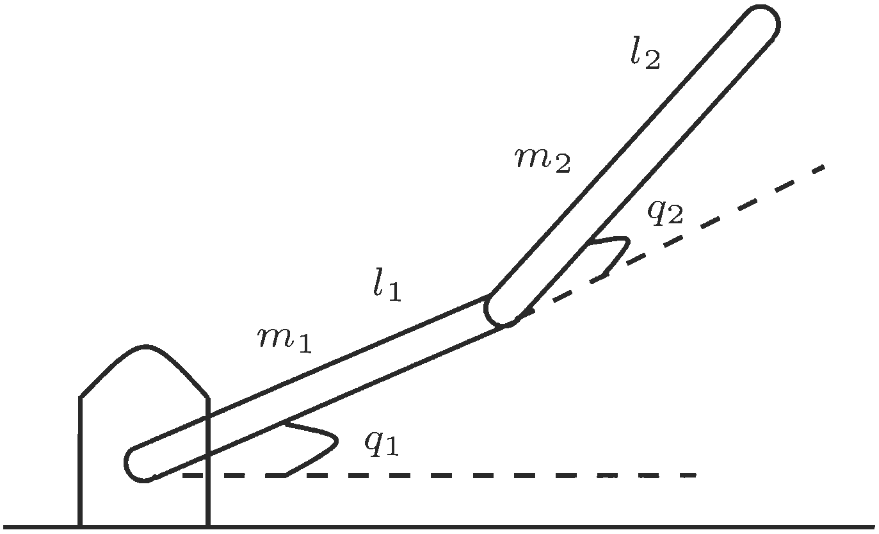


FIGURE 6.2 – Planar robotic manipulator

$$M(q) = \begin{bmatrix} m_1 l^2 + 2m_2 l^2(1+\cos(q_2)) & m_2 l^2(1+\cos(q_2)) \\ m_2 l^2(1+\cos(q_2)) & m_2 l^2 \end{bmatrix} \tag{6.9a}$$

$$C(q,\dot{q}) = \begin{bmatrix} -2m_2 l^2 \sin(q_2)\dot{q}_2 & -m_2 l^2 \sin(q_2)\dot{q}_2 \\ m_2 l^2 \sin(q_2)\dot{q}_1 & 0 \end{bmatrix} \tag{6.9b}$$

$$\Phi = \begin{bmatrix} \ddot{q}_1 & (1+\cos(q_2))2\ddot{q}_1 + (1+\cos(q_2))\ddot{q}_2 - (2\dot{q}_1\dot{q}_2 + \dot{q}_2)\sin(q_2) \\ 0 & (1+\cos(q_2))\ddot{q}_1 + \ddot{q}_2 + \dot{q}_1^2 \sin(q_2) \end{bmatrix} \tag{6.10}$$

The load's desired angular positions in both joints are taken as the step response of a critically damped second order system with a natural frequency $w_n = 2$ rad/s, as shown in Fig. 6.3.

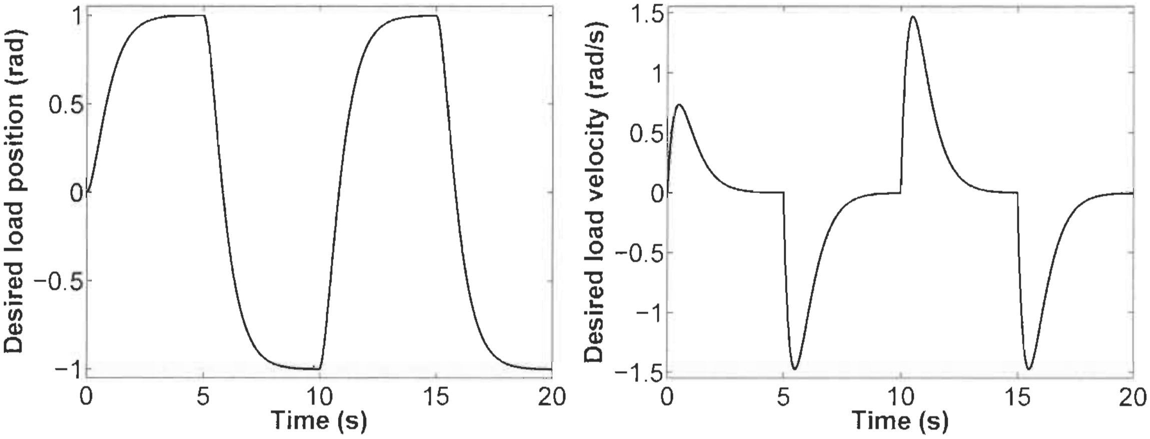


FIGURE 6.3 – Joints position and velocity reference signals

6.3.2 Results

The manipulator's position and velocity errors for both links decay gradually before stabilizing within negligible amplitude while its parameters converge to their respective values (Fig. 6.4). Although the satisfactory tracking performance obtained by the controller, it neglects important aspects such as friction and disturbance. This has negative effects on the performance and the stability of such systems, which is considered the major issue of classic adaptive controllers (Fig. 6.5). Next, we will address these drawbacks with the design of friction and disturbance compensators for flexible-joint manipulators.

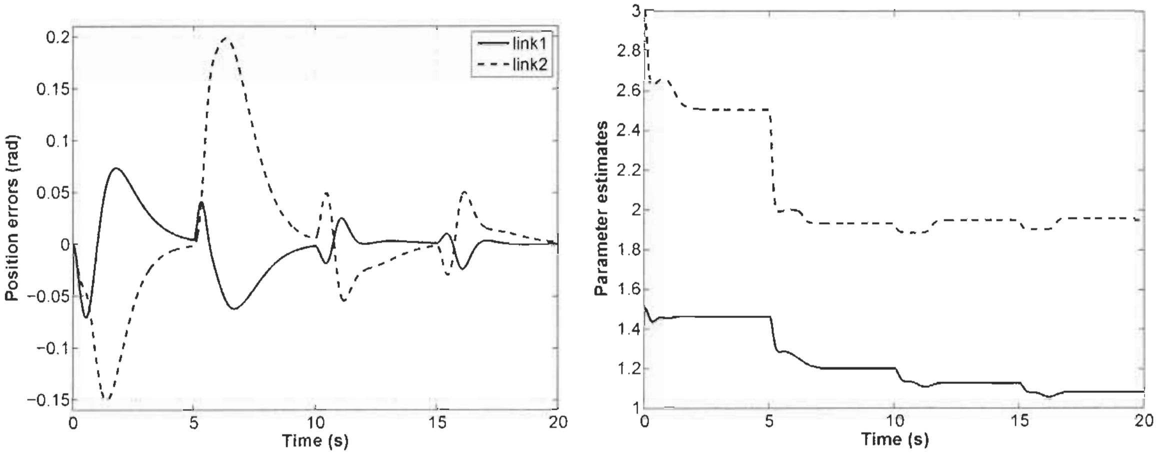
6.4 Adaptive Friction Compensation for Flexible-Joint Manipulators

In here, we present two formulations for compensating friction. The model (5.6) can be formulated in a linear regression form as $F(\dot{\sigma}) = (F_1, \ldots, F_n)$, where

$$F = \Phi^T(\dot{\sigma}, \eta_s)W \tag{6.11a}$$

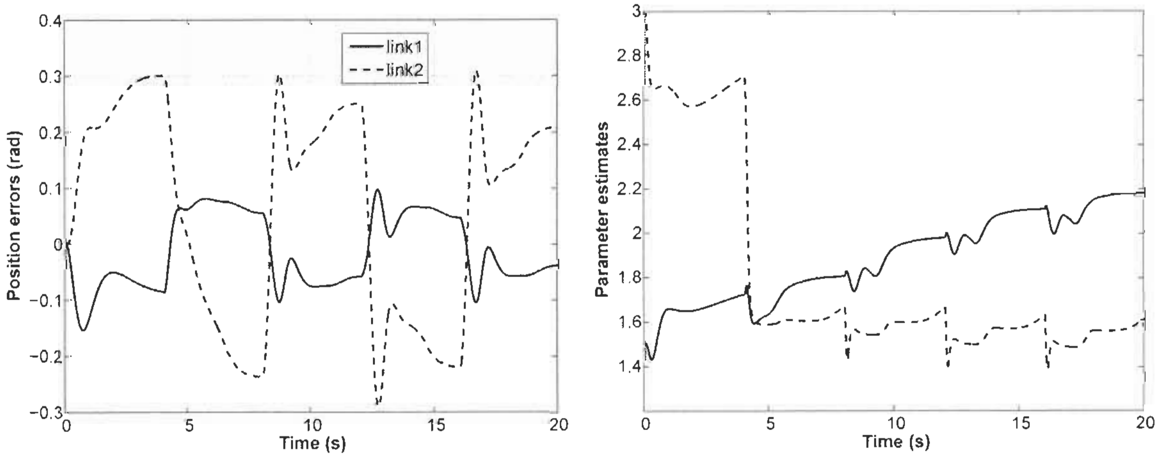
$$\Phi^T(\dot{\sigma}, \eta_s) = \begin{bmatrix}\text{sign}(\dot{\sigma}) & \dot{\sigma} & \text{sign}(\dot{\sigma})e^{-(\dot{\sigma}/\eta_s)^2}\end{bmatrix} \tag{6.11b}$$

$$W = \begin{bmatrix}F_c & F_v & F_s\end{bmatrix} \tag{6.11c}$$



(a) (b)

FIGURE 6.4 – Manipulator response with nominal values : (a) position errors ; and (b) parameters $\hat{W}$.



(a) (b)

FIGURE 6.5 – Manipulator response with Coulomb friction $\tau_F = \text{sign}(\dot{q})$: (a) position errors ; and (b) parameters $\hat{W}$.

In the following, we will refer to model (6.11) as the full friction model.

In a single stage speed reduction system, a lumped flexibility model can be used if the dominant flexion appears in the gear teeth. In the following, the input gears inertia are combined with the actuators inertia J_m while the output gears inertia is lumped with the load inertia $M(q)$. As shown in [58] and references therein, this model reduction method has been used in multi-stage reduction systems, such as planetary gears, and in multi-mass flexibility models, like harmonic drives, for instance.

Adopting the full friction model (6.11), τ_{fl} and τ_{fm} in (6.2) can be expressed with

$$\tau_{fm} = \Phi^T(\dot{\theta}, \eta_{sm})W_{fm}$$
$$\tau_{fl} = \Phi^T(\dot{q}, \eta_{sl})W_{fl}$$

for some unknown/uncertain or slowly time-varying positive parameters $W_{fm}, W_{fl} \in \mathbb{R}^n$ and $\eta_{sm}, \eta_{sl} \in \mathbb{R}$, $i = 1, \ldots, n$, which would be tracked by the adaptive controller to be designed later.

The manipulator's overall friction $\tau_{Fr} \in \mathbb{R}^n$ can be regarded as the sum of the actuators and loads friction terms, which leads to

$$\tau_{Fr} = \tau_{fm} + \tau_{fl}$$

As such, the friction term τ_{Fr} can be written as

$$\tau_{Fr} = \Phi^T(\dot{\theta}, \eta_{sm})W_{fm} + \Phi^T(\dot{q}, \eta_{sl})W_{fl} \tag{6.12}$$

Despite its precision in reflecting the real friction dynamics, friction model (6.12) is nonlinear in the rate of decay parameters η_{sm}, η_{sl}. Therefore, a nonlinear approximation method, such as a gradient descent technique, would have to be used by the controller to track down these parameters.

Nevertheless, the regression vector $\Phi^T(\dot{x}, \eta_s)$ in (6.11) can be approximated by a simpler linear form as [49] :

$$\Phi^T(\dot{x}) = \begin{bmatrix} 1 & |\dot{x}| & \sqrt{|\dot{x}|} \end{bmatrix} \operatorname{sign}(\dot{x}) \tag{6.13}$$

This yields the formulation of the manipulator's friction in a pure linear regression,

$$\tau_{Fr} = \Phi^T(\dot{\theta})\, W_{fm} + \Phi^T(\dot{q})\, W_{fl} \tag{6.14}$$

Hence, the simplified linear friction model (6.14) can be regarded as an alternative for the nonlinear full friction model represented by (6.12). It is clear that the former has an obvious computational complexity advantage over that of the latter. Next, we will verify if adopting the linear friction compensation model for controlling a flexible-joint manipulator would have any negative effect on the controller's performance as opposed to opting for the full friction model.

Let $e_q = q - q_d$ and $e_\theta = \theta - \theta_d$ denote the links' and motors' position errors, respectively, with θ_d being the unknown desired time-dependent motor position vector. The control strategy is based on the design of an adaptive controller that not only leads to a precise tracking of the system's nominal desired signals, but also improves the motors' internal stability. Should the motors' desired position θ_d have been available, the control strategy would be based on tracking e_q and e_θ to zero. As that is not the case, we define the following compounded velocity error signal [59] :

$$\dot{e}_q = \left(\Lambda\dot{q} + (1-\Lambda)\frac{1}{r}\dot{\theta}\right) - \dot{q}_d \tag{6.15}$$

for a diagonal matrix $\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ with $\lambda_i \in [0, 1]$, $i = 1, \ldots, n$. The feedback matrix gain Λ is introduced to provide a trade off between the link tracking performance and internal stability, due to the high nonlinear coupling between the two.

An adaptive Lyapunov stability-based feedforward (FF) is designed to learn online the manipulator's inverse dynamics (6.2). Due to the iterative nature of the learning mechanism and because of the high complexity order of the system's dynamical model, the feedforward controller may take a relatively long time to converge which may lead to an unstable or unsatisfactory performance. To alleviate this problem, a linear in parameters friction compensation

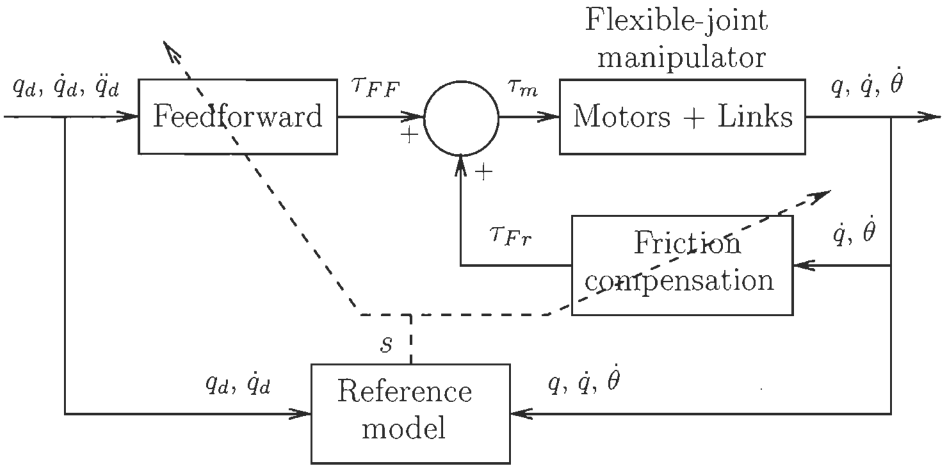


FIGURE 6.6 – Friction compensation control scheme.

technique is used to reduce the complexity of the flexible manipulator. A first order reference model is introduced to define the desired dynamics of the error between the desired and actual load positions and between the motor's and load's velocities to assure the controller's internal stability. A block diagram of the control structure is illustrated in Fig. 6.6.

Define a filtered error s and joint reference signal $\dot{q}_r$ such that

$$s = \dot{e}_q + \Psi e_q = \Lambda \dot{q} + (1 - \Lambda)\frac{1}{r}\dot{\theta} - \dot{q}_r \tag{6.16}$$

$$\dot{q}_r = \dot{q}_d - \Psi e_q$$

where $\Psi = \mathrm{diag}(\psi_1, \psi_2, \ldots, \psi_n)$ with ψ_i being a positive constant, $i = 1, \ldots, n$. Using singular perturbation method, we can let $K \to \infty$, i.e., displacement $(\frac{\theta}{r} - q) \to 0$. Equation 6.16 can be written as,

$$s = \dot{q} - \dot{q}_r \tag{6.17}$$

The Euler-Lagrange formulation (6.2) can be reformulated as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_{fl} + \tau_{dl} - \tau_t = 0$$

$$J_m\ddot{\theta} + \tau_{fm} + \tau_{dm} - \tau_m = -\frac{1}{r}\tau_t$$

where q, $\dot{q}$, $\ddot{q}$, $\ddot{\theta} \in \mathbb{R}^n$. Substituting for τ_t and letting the stiffness constant tend to infinity $(K \to \infty)$, i.e., displacement $(\frac{\theta}{r} - q) \to 0$. We obtain for following rigid model [52] :

$$M_t(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_{Fr} = r\tau_m \tag{6.18}$$

where,

$$M_t(q) = J_m r^2 + M(q)$$
$$\tau_{Fr} = r\tau_{fm} + \tau_{fl}$$

In this design, we aim friction compensation and therefore, disturbance attenuation is beyond our scope. Hence, the terms τ_{dl}, τ_{dm} are ignored. Therefore, by using the linear in parameter property of the manipulator dynamics, we can write the model as a linear regression :

$$M_t(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r + G(q) + \tau_{Fr} = \Phi^T(\ddot{q}_r, \dot{q}_r, \dot{q}, q)\, W$$
$$\Phi_{plant}\, W_{plant} + \Phi_{fm}\, W_{fm} + \Phi_{fl}\, W_{fl} = \Phi^T\, W$$

where $\Phi_{f\bullet} = \Phi^T(\dot{x}, \eta_{s\bullet})$ or $\Phi^T(\dot{x})$ as described earlier. $\Phi_{plant} \in \mathbb{R}^{n\times m}$ is a matrix of known functions (regressor), and W_{plant} is a m-dimensional vector of parameters.
The control law is :

$$r\tau_m = \Phi^T\, \hat{W} - K_d s \tag{6.19}$$

where $K_d = \text{diag}(k_{d_1}, k_{d_2}, \ldots, k_{d_n})$ with k_{d_i} is a positive constant gain, $i = 1, \ldots, n$, and,

$$\Phi = [\Phi_{fm}\ \Phi_{fl}\ \Phi_{plant}]$$
$$\hat{W} = [\hat{W}_{fm}\ \hat{W}_{fl}\ \hat{W}_{plant}]$$

$\hat{\bullet}$ denotes the parameter estimate vector and $\tilde{\bullet} = \bullet - \hat{\bullet}$ denotes the parameter estimate error vector.

Theorem 10 *Consider a nonlinear system in the form (6.2) under the assumption of high stiffness so that singular perturbation applies with the reference model (6.16) and the control law (6.19). The adaptive control law is asymptotically stable and the tracking error converges to zero with the following adaptation law :*

$$\dot{\hat{W}} = -\Gamma \, \Phi \, s$$

Proof 4 *Let's take the derivative of the error signal s in (6.17) :*

$$\dot{s} = \ddot{q} - \ddot{q}_r$$

$$M_t(q)\dot{s} = M_t(q)\ddot{q} - M_t(q)\ddot{q}_r$$

Substituting $M_t(q)\ddot{q}$ from (6.18) :

$$M_t(q)\dot{s} = r\tau_m - M_t(q)\ddot{q}_r - C(q,\dot{q})\dot{q}_r - G_t(q) - \tau_{Fr} - C(q,\dot{q})s$$

The linear in parameters property yields

$$M_t(q)\dot{s} = r\tau_m - \Phi^T W - C(q,\dot{q})s \tag{6.20}$$

Set

$$r\tau_m = \Phi^T \hat{W} - K_D s$$

Equation (6.20) becomes

$$M_t(q)\dot{s} = \Phi^T \tilde{W} - C(q,\dot{q})s - K_D s$$

Choose the Lyapunov candidate function :

$$V = \frac{1}{2}\{s^T M_t(q)s + \tilde{W}^T \Gamma^{-1} \tilde{W}\}$$

Take the time-derivative of V :

$$\dot{V} = s^T M_t(q)\dot{s} + \frac{1}{2} s^T \dot{M}_t(q) s + \tilde{W}^T \Gamma^{-1} \dot{\hat{W}}$$

Since the unknown parameters W are constants, $\dot{\tilde{W}} = \dot{\hat{W}}$.
Substituting for $M_t(q)\dot{s}$ *and set* $s^T \{(\dot{M}_t(q) - 2C(q,\dot{q}))\} s = 0$ *due to skew-symmetry property :*

$$\dot{V} = s^T \Phi^T \tilde{W} + \tilde{W}^T \Gamma^{-1} \dot{\hat{W}} - s^T K_d s$$

Set the adaptation law as :

$$\dot{\hat{W}} = -\Gamma\, \Phi\, s$$

Then,

$$\dot{V} = -s^T K_d\, s \leq 0$$

Therefore V ; and so s, $\tilde{W}$ *and* $\hat{W}$, *are bounded and converge to finite values. It implies from (6.17) that* e_q *and* $\dot{e}_q$, *and so,* q *and* $\dot{q}$ *are bounded. It follows from (6.19) that* τ_m *is bounded, which implies from (6.20) that* $\dot{s}$ *is bounded. Therefore,* $\ddot{V} = -2s^T K_D \dot{s}$ *is also bounded. Hence, from Barbalat's Lemma, it implies that* $\lim_{t \to \infty} \dot{V} = 0$. *Thus,* $\lim_{t \to \infty} s = 0$. *Since* $\dot{s}$ *is bounded,* $\ddot{e}_q$ *is also bounded. Barbalat's Lemma also shows that* $\lim_{t \to \infty} \dot{e}_q = 0$. *Thus,* $\lim_{t \to \infty} e_q = 0$.
Therefore, $\lim_{t \to \infty} q = q_d$ *and* $\lim_{t \to \infty} \dot{q} = \dot{q}_d$.

6.4.1 Setup

To demonstrate the performance of the proposed controller, a set of computer simulation runs is carried out on a planar flexible-joint manipulator (Fig. 6.2). Table 6.1 summarizes the manipulator's physical parameters along with their respective values. Both links are chosen to be identical. The joints stiffness coefficient and gear ratio are assumed to be $K = 9$ N·m/rad and $r = 1$.

TABLE 6.1 – Manipulator's physical parameters ($i = 1,2$)

Parameter	Link	Motor
rotational inertia (kg·m^2)		$J_{m_i} = 4 \cdot 10^{-3}$
viscous friction coefficient (N·m·s/rad)	$F_{vl_i} = 2 \cdot 10^{-2}$	$F_{vm_i} = 1 \cdot 10^{-2}$
Coulomb friction coefficient (N·m)	$F_{cl_i} = 5 \cdot 10^{-2}$	$F_{cm_i} = 2 \cdot 10^{-2}$
static friction coefficient (N·m)	$F_{sl_i} = 3 \cdot 10^{-2}$	$F_{sm_i} = 7 \cdot 10^{-2}$
static friction decreasing rate (rad/s)	$\eta_{sl_i} = 5 \cdot 10^{-2}$	$\eta_{sm_i} = 4 \cdot 10^{-2}$
link's mass (kg)	$m_i = 0.5$	
link's length (m)	$l_i = 0.4$	

The manipulator's dynamics is defined by (6.21), where $g = 9.8$ m/s^2 is the gravitational constant. The controller is set to obtain a bandwidth of 100 Hz and the load's desired angular positions and velocities in both joints are set as in Fig. 6.3.

$$M(q) = \begin{bmatrix} m_{12}l_1^2 + m_2 l_2^2 + 2ac_2 & m_2 l_2^2 + ac_2 \\ m_2 l_2^2 + ac_2 & m_2 l_2^2 \end{bmatrix} \tag{6.21a}$$

$$C(q,\dot{q}) = \begin{bmatrix} -2as_2\dot{q}_2 & -as_2\dot{q}_2 \\ as_2\dot{q}_1 & 0 \end{bmatrix} \tag{6.21b}$$

$$G(q) = \begin{bmatrix} m_{12}gl_1c_1 + m_2gl_2\cos(q_1+q_2) \\ m_2gl_2\cos(q_1+q_2) \end{bmatrix} \tag{6.21c}$$

$$J_m = \begin{bmatrix} J_{m_1} & 0 \\ 0 & J_{m_2} \end{bmatrix} \tag{6.21d}$$

$$m_{12} = m_1 + m_2, a = m_2 l_1 l_2 \tag{6.21e}$$

$$c_i = \cos(q_i), s_i = \sin(q_i),\ i = 1,2 \tag{6.21f}$$

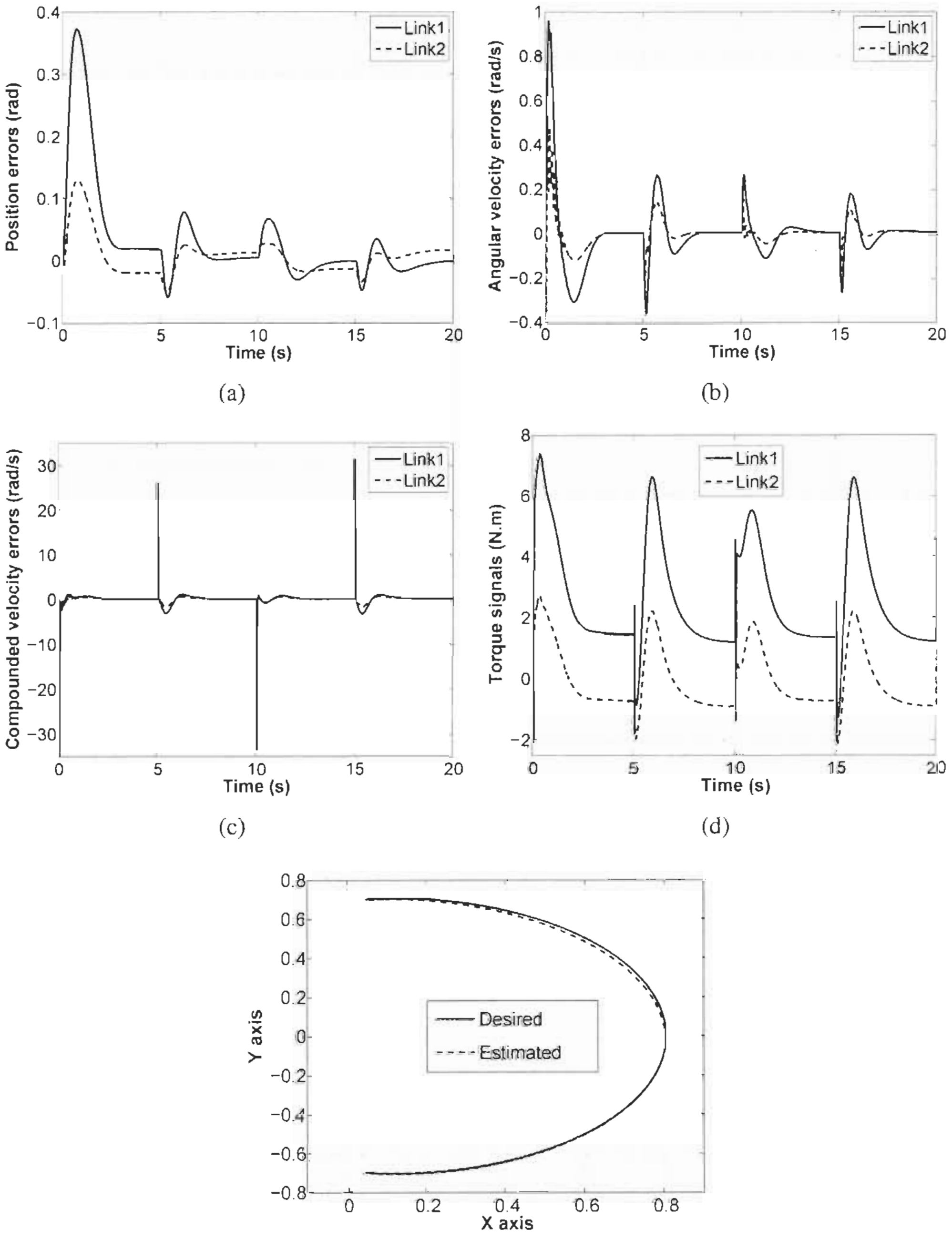
The regression matrix Φ_{plant} is given by (6.22) while the parameter vector is, $W_{Plant} = [m_i l_i^2 \; m_i l_i \; J_{m_i} r]$, with $i = 1,2$.

$$\Phi_{Plant} = \begin{bmatrix} (3+2c_2)\ddot{q}_1 + (1+c_2)\ddot{q}_2 - (2\dot{q}_1\dot{q}_2 + \dot{q}_2^2)s_2 & 2gc_1 + g\cos(q_1+q_2) & \ddot{q}_1 \\ (1+c_2)\ddot{q}_1 + \ddot{q}_2 + \dot{q}_1^2 s_2 & g\cos(q_1+q_2) & \ddot{q}_2 \end{bmatrix} \tag{6.22}$$

6.4.2 Results

The system's response is studied taking into account the manipulator's position and velocity errors, the system's stability, the controller's output torques τ_m for both links, and the end-effector's trajectory. The aforementioned nominal values are used to simulate the manipulator's dynamics with the two different friction compensation estimates for (6.12) and (6.14). In this case, the two friction models led to comparable performances due to the relatively low severity degree of the friction magnitude. As such, only the results of the full friction compensation method (6.12) are shown (Fig. 6.7). The manipulator's position and velocity errors for both links decay gradually before stabilizing within negligible amplitude while maintaining the system's internal stability by making the motor's speed converges to zero. On the other hand, it is noticeable how the controller generates a torque impulse whenever the actuator's velocity crosses zero to compensate for the highly nonlinear friction around the zero velocity neighborhood, which contributes the most to the error observed on the load's trajectory. A satisfactory end-effector's trajectory tracking is obtained. Figs. 6.8 and 6.9 reveal the parameter estimation performance. The convergence of the estimation parameters is inversely proportional to the value of Γ_i. Increasing Γ_i leads to a better global convergence at the expense of more noise in the estimates. Tracking the friction at each joint is a challenging task due to the severe nonlinearities associated to it. This is observed from the friction tracking performance of the two proposed friction compensation methods illustrated in Fig. 6.9. The superiority of the full friction compensation model over its simplified linear counterpart is evident.

Then, the friction is magnified 25 times to articulate the difference between the proposed friction compensation methods. As shown in Fig. 6.10, the full friction approximation method outperformed the simplified linear friction compensation technique. The latter overcompensated for the nonlinear friction terms that are not explicitly modeled, which resulted in severe



(a)

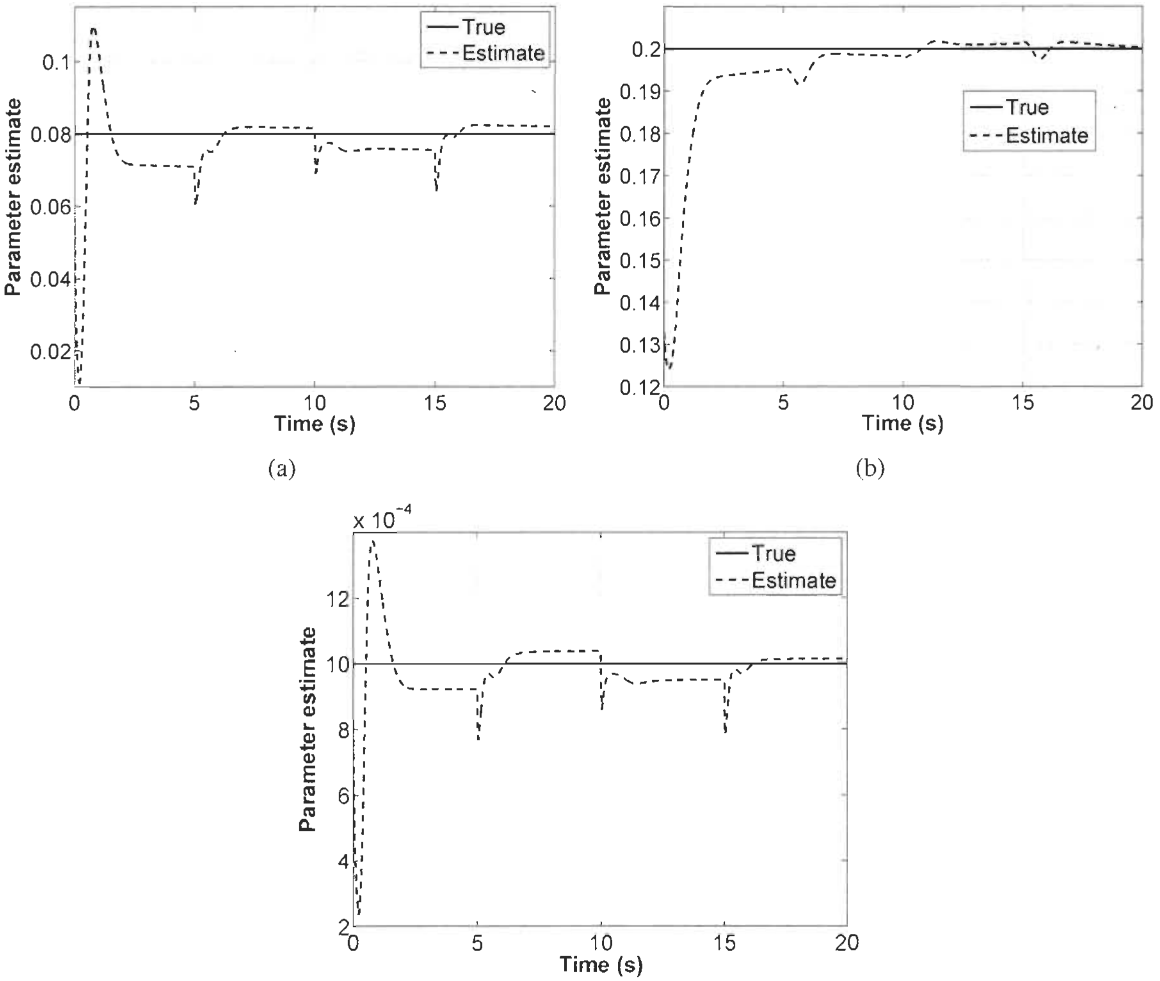
(b)

(c)

(d)

(e)

FIGURE 6.7 – Friction compensation response with nominal values : (a) manipulator's position error ; (b) manipulator's velocity error ; (c) manipulator's internal stability ; (d) controller's output torque τ_m ; and (e) end-effector's trajectory.

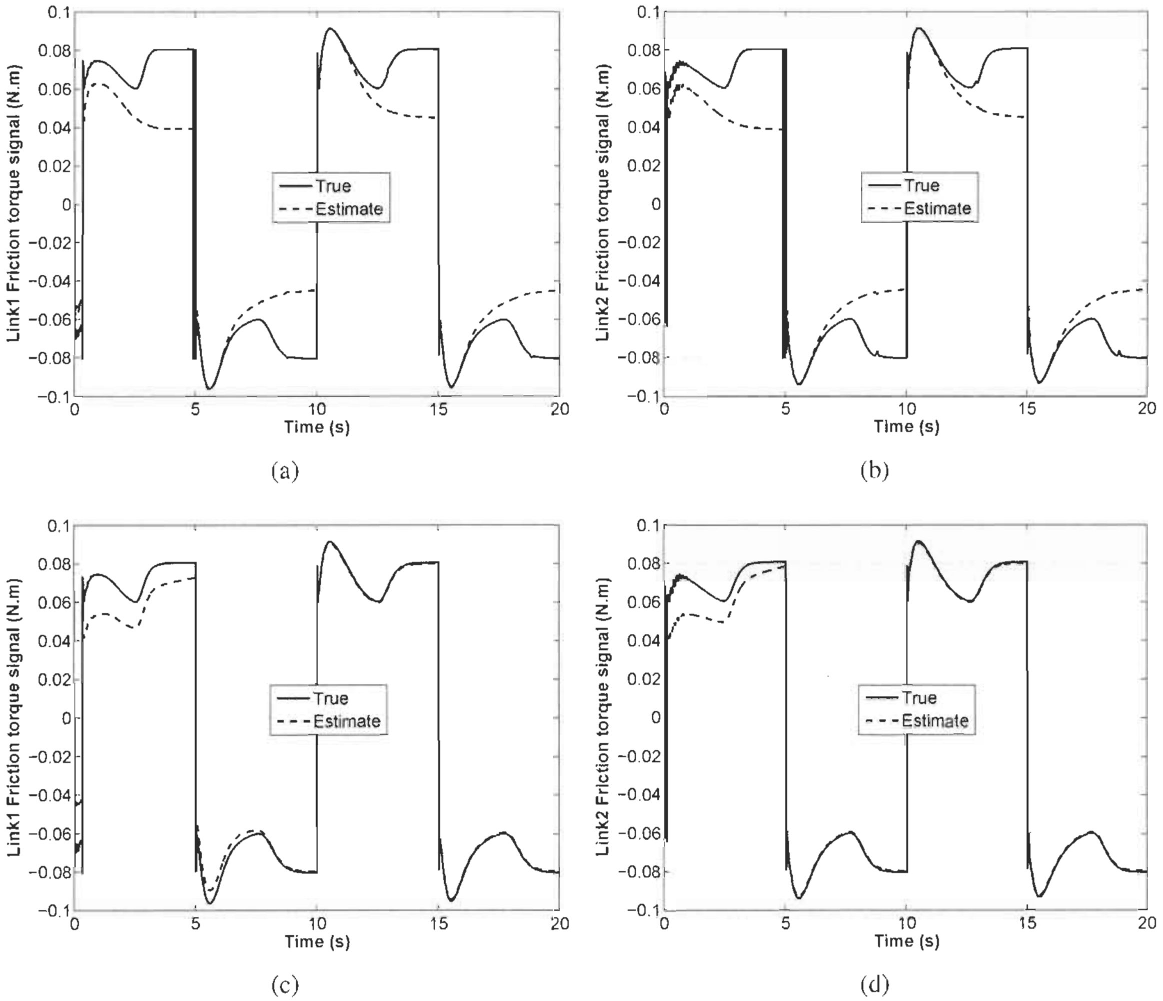


(a)

(b)

(c)

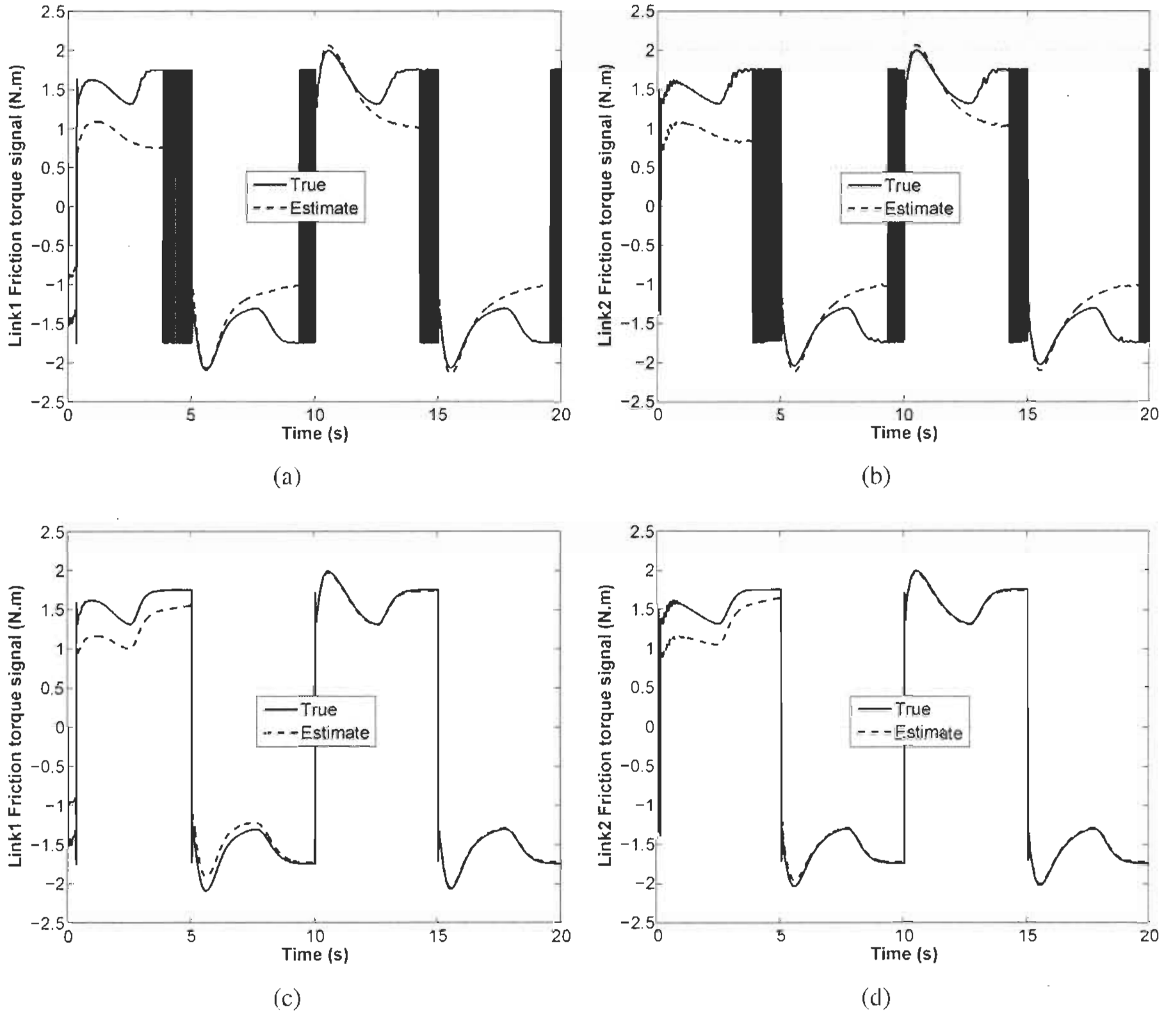
FIGURE 6.8 – Parameters estimate : (a) $\hat{W}_{plant1} \simeq ml^2$; (b) $\hat{W}_{plant2} \simeq ml$; and (c) $\hat{W}_{plant3} \simeq J_m r$.



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

FIGURE 6.9 – Friction estimate with nominal values : (a) and (b) partial friction compensation $\hat{\tau}_{Fr}$; and (c) and (d) full friction compensation $\hat{\tau}_{Fr}$.

oscillations in the actuator (Figs. 6.10(a) and 6.10(b)). On the other hand, no limit cycles were observed with the full friction compensation model (Figs. 6.10(c) and 6.10(d)), which yielded a smooth friction compensation control signal.



(a)

(b)

(c)

(d)

FIGURE 6.10 – Friction estimate with magnified friction : (a) and (b) partial friction compensation $\hat{\tau}_{Fr}$; and (c) and (d) full friction compensation $\hat{\tau}_{Fr}$.

The full friction compensation model is a key to achieve this control accuracy since the feedforward control signal alone, τ_{FF}, cannot compensate for these effects because the zero crossings of the motor velocity cannot be predicted from the load's desired trajectory. Hence, the full friction compensation model will be used in the rest of the simulations.

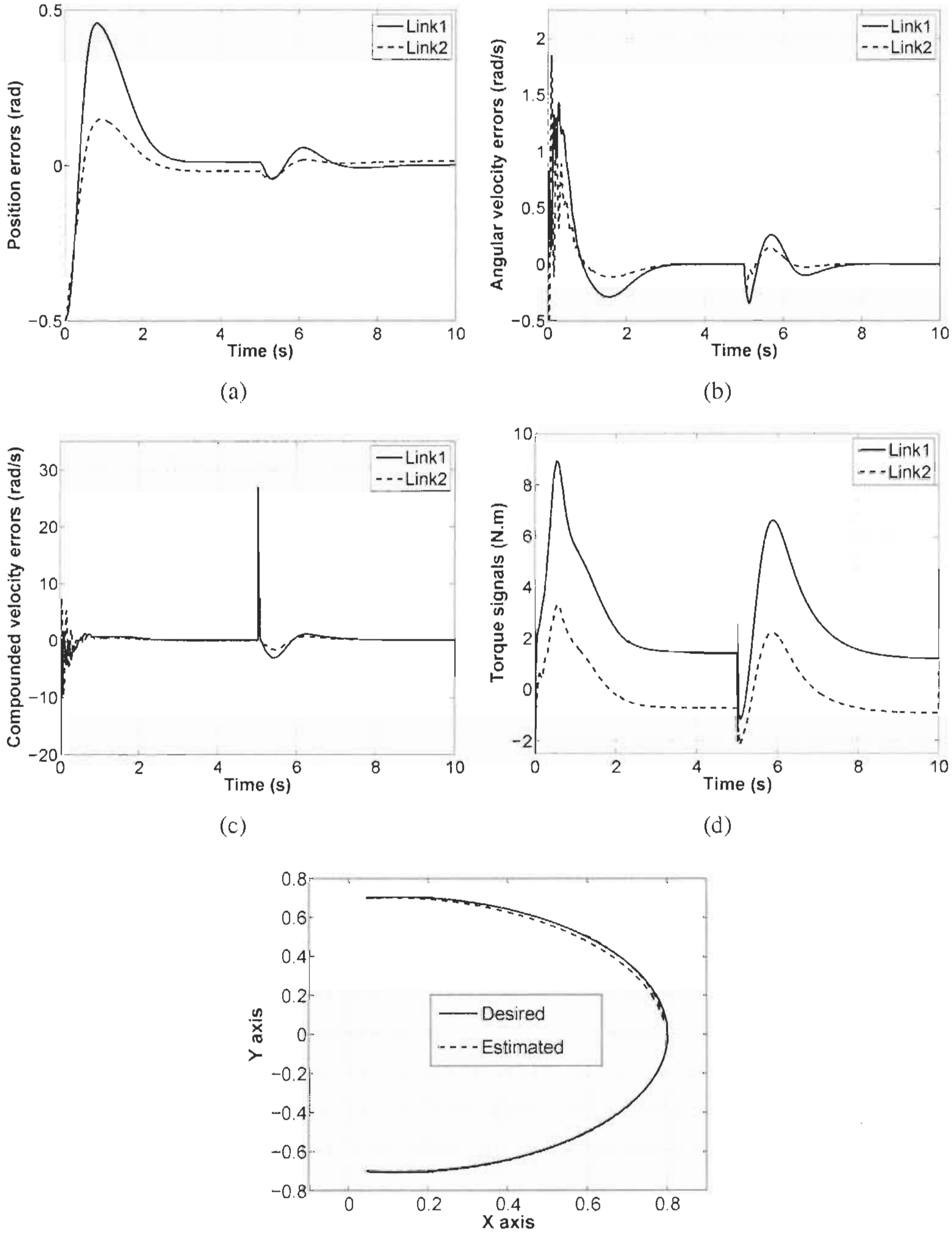
To study the effect of initial conditions on the proposed controller, an initial condition of $(q = 0.5, \theta = 0.25)$ rad is introduced. Note that these initial conditions implicitly imply a non-zero initial torque τ_t at the flexible transmission (see formulation (6.2)). The controller provides a satisfactory transient response. The manipulator's position and velocity errors as well as the motor's speed converge to zero soon after some time. As it can be observed in Fig. 6.11(d), The control signal is quite significant at the beginning of the simulation to compensate for the system's initial error. The controller's characteristics beyond the first unit step are similar to those of the nominal case (Fig. 6.7). The controller is also compensating for the highly nonlinear friction terms by providing a smooth control torque signal.

The link's nominal inertia is doubled to show the modularity of the proposed controller. The results are shown in Fig. 6.12. As it can be seen, the change in the manipulator's dynamics has little effect on the controller's performance. It maintains a similar behavior as in Fig. 6.7 but with slightly larger oscillations at the startup, which is expected due to the longer time needed for the adaptive parameters to settle down.

Although the better performance obtained by the proposed controller. It assumes a high stiffness so that singular perturbation applies and therefore, it remains incapable of dealing with high elasticity and disturbance, which are ignored in the control design. Thus, next section addresses these issues.

6.5 Adaptive Disturbance Compensation for Flexible-Joint Manipulators

The feedforward controller designed in the previous section is used to mimic the manipulator's inverse dynamics (6.2). Due to the high complexity order of the system's dynamical model, an adaptive neural network controller is used to alleviate the feedforward controller task by compensating for residual errors due to modeling and parametric uncertainties. A block diagram of the control structure is illustrated in Fig. 6.13. Therefore, we obtain the following



(a)

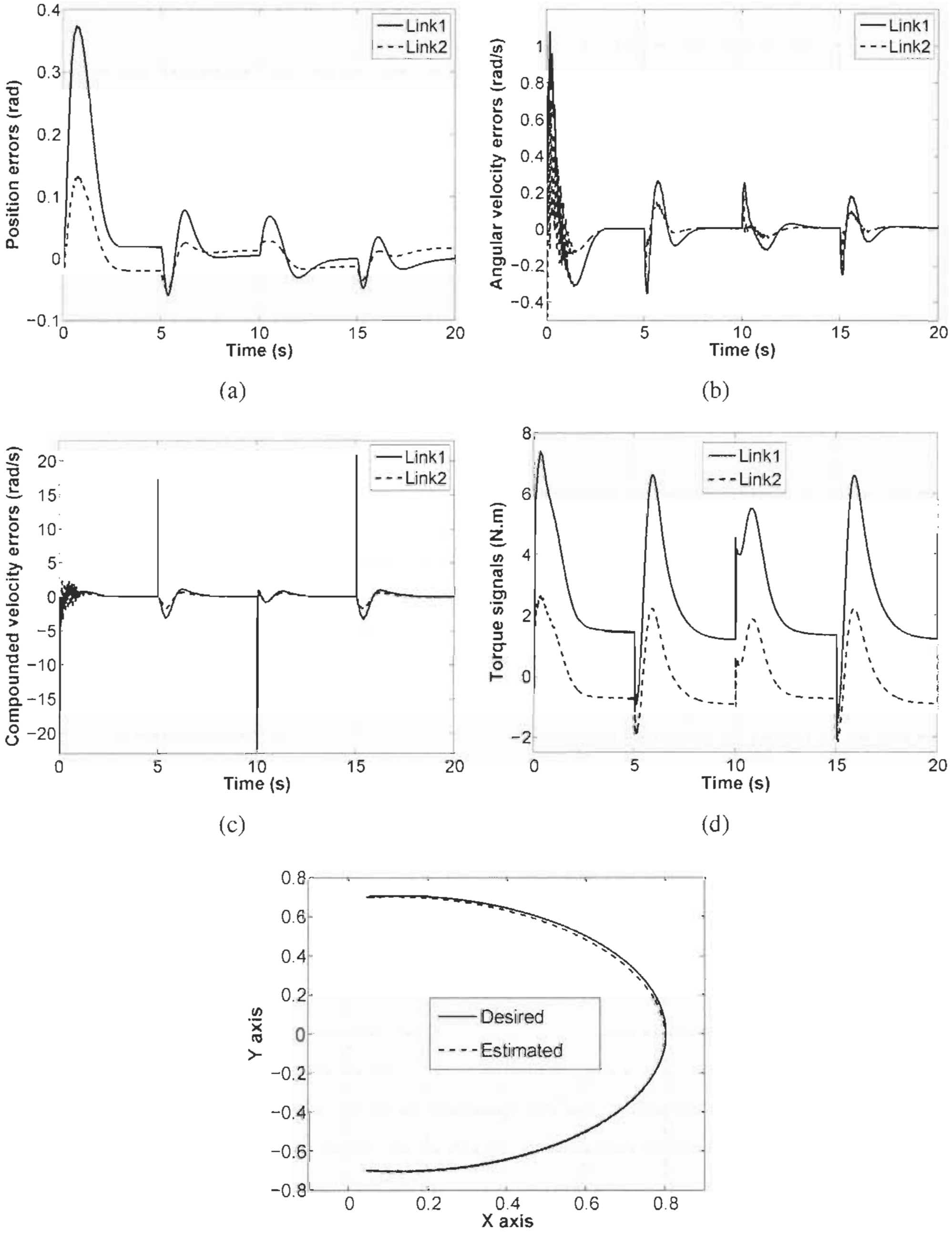
(b)

(c)

(d)

(e)

FIGURE 6.11 – Friction compensation response with initial conditions $(q = 0.5, \theta = 0.25)$ rad : (a) manipulator's position error ; (b) manipulator's velocity error ; (c) manipulator's internal stability ; (d) controller's output torque τ_m ; and (e) end-effector's trajectory.



(a)

(b)

(c)

(d)

(e)

FIGURE 6.12 – Friction compensation response with twofold motor inertia : (a) manipulator's position error ; (b) manipulator's velocity error ; (c) manipulator's internal stability ; (d) controller's output torque τ_m ; and (e) end-effector's trajectory.

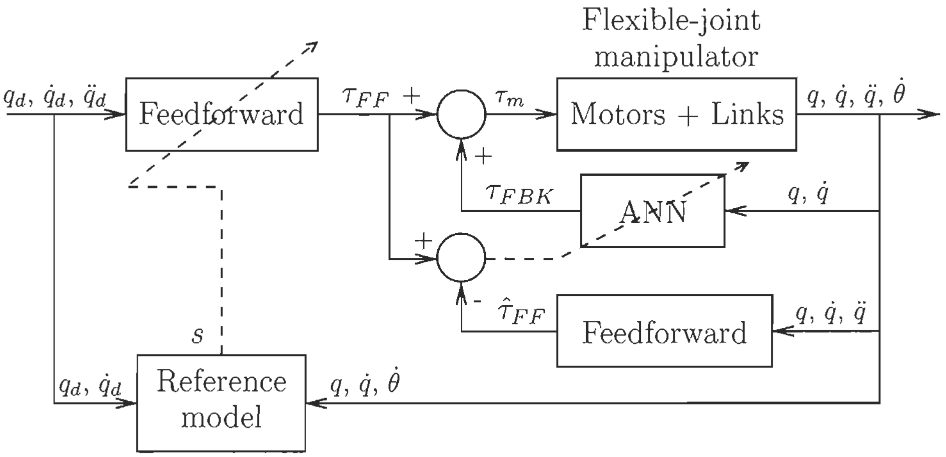

Flexible-joint manipulator
q_d, $\dot{q}_d$, $\ddot{q}_d$
Feedforward
τ_{FF} +
τ_m
Motors + Links
q, $\dot{q}$, $\ddot{q}$, $\dot{\theta}$
+
τ_{FBK}
ANN
q, $\dot{q}$
+
-
$\hat{\tau}_{FF}$
Feedforward
q, $\dot{q}$, $\ddot{q}$
s
q_d, $\dot{q}_d$
Reference model
q, $\dot{q}$, $\dot{\theta}$

FIGURE 6.13 – Disturbance compensation control scheme.

rigid model by using the singular perturbation method,

$$M_t(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)+\tau_{Frd}=r\tau_m \tag{6.23}$$

where,

$$M_t(q)=J_m r^2+M(q)$$

$$\tau_{Frd}=r\tau_{fm}+r\tau_{dm}+\tau_{fl}+\tau_{dl}$$

Using the linear in parameter property of the manipulator dynamics, we can write the model as a linear regression :

$$M_t(q)\ddot{q}_r+C(q,\dot{q})\dot{q}_r+G(q)=\Phi^T(\ddot{q}_r,\dot{q}_r,\dot{q},q)\,W$$

where $\Phi \in \mathbb{R}^{n\times m}$ is a matrix of known functions (regressor), and W is a m-dimensional vector of parameters. This representation is the same as the one used in the previous section for Φ_{plant} and W_{plant}. Therefore, the same control and adaptation laws are used and the term τ_{Frd} is approximated by the adaptive neural network feedback controller.

Let $\tilde{\tau}_{FF}$ be the feedforward torque error mainly due to τ_{Frd} be defined as follows :

$$\tilde{\tau}_{FF} = \hat{\tau}_{FF} - \tau_{FF}$$

This equation can be written as :

$$\tilde{\tau}_{FF} = \hat{M}_t(q)\dot{s} + \hat{C}(q,\dot{q})s$$

It is worth pointing out that the feedforward torque error $\tilde{\tau}_{FF}$ is expressed in terms of the error signal s and its derivative $\dot{s}$ that hold information about modeling and parametric uncertainties. Henceforth, a feedback controller is designed to provide τ_{FBK} by minimizing the feedforward torque error $\tilde{\tau}_{FF}$.

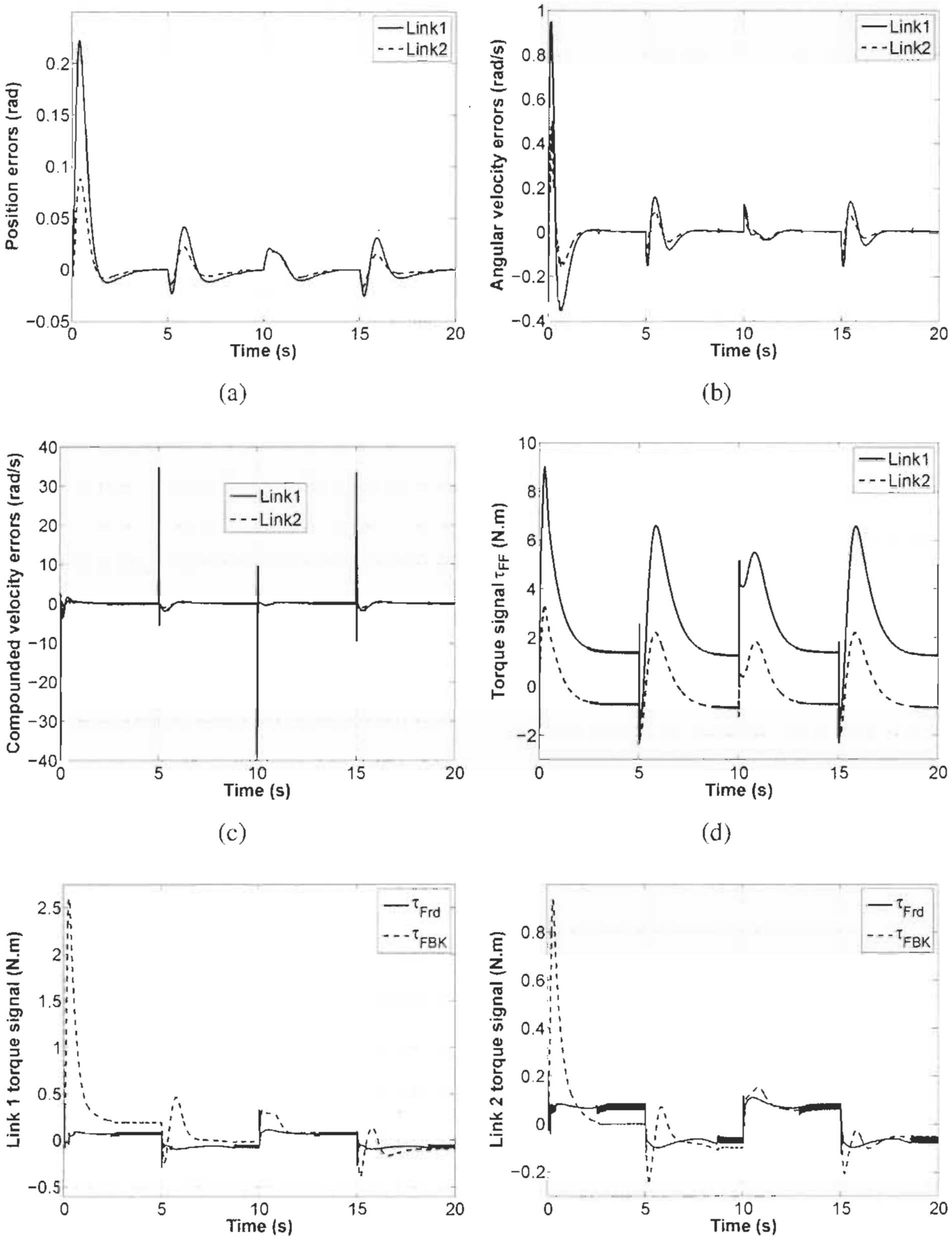
In this strategy, the feedforward controller is used as a robotic manipulaor's inverse model approximator. By using this model with the feedback signals, we obtain the estimated resultant torque $\hat{\tau}_{FF}$. The difference between this signal and the feedforward signal τ_{FF} is due to structured and unstructured uncertainties. In other words, should the exact inverse model have been available, a perfect tracking would be achieved and the feedforward control signal would be equal to the estimated torque $\hat{\tau}_{FF}$. As exact inverse model is not available for such system, a neural network based controller takes the difference between the two torque signals and estimates the feedback signal τ_{FBK} to compensate for residual errors mainly due to elasticity, friction and disturbance. In the proposed approach, the multilayered neural network controller is composed of three layers each : one input layer of two neurons, one hidden layer with six neurons and one neuron for the output layer. The sigmoid function is used as activation function for all neurons except for the output neuron which uses a linear function.

6.5.1 Results

The system's response is studied taking into account the manipulator's position and velocity errors, the system's stability, the controller's output torques τ_m for both links, and the force torque τ_{Frd} with the controller's feedback torque τ_{FBK} for both links. The aforementioned nominal values are used to simulate the manipulator's dynamics. As it is shown in Fig. 6.14, the manipulator's position and velocity errors for both links decay gradually before stabilizing within a negligible amplitude while maintaining the system's internal stability by making the motor's speed converge to zero. On the other hand, the neural network controller shows good performance in transient stage by dealing with parametric uncertainties and once the parameters converge to their respective values, it generates a torque impulse whenever the actuator's velocity crosses zero to compensate for the highly nonlinear friction around the zero velocity neighborhood, which contributes the most to the error observed on the load's trajectory. The feedforward control signal cannot compensate alone for these effects since the zero crossings of the motor velocity cannot be predicted from the load's desired trajectory. It is noteworthy that the adaptive neural network feedback controller also compensates for joints elasticity not explicitly modeled in the feedforward controller. Tracking the friction at each joint is a challenging task due to the severe nonlinearities associated to it. This is observed from the friction tracking performance of the adaptive neural network feedback controller illustrated in Fig. 6.14(e) and Fig. 6.14(f). The convergence of the parameters estimate is shown in Fig. 6.15.

In here, more joints elasticity is introduced by reducing the two links torsion constants K to the third of their nominal values by setting $K = 3$. The results are shown in Fig. 6.16. As it can be seen, the controller maintains a similar behavior as in Fig. 6.14 but with slightly larger torque pulses from the feedback, which is expected due to the larger control effort needed to compensate for the additional flexion. It is important to note that although the joints elasticity is added, the manipulator's position error remains in the same range as in Fig. 6.14. The joints elasticity compensation can be clearly seen in the feedback torque signal τ_{FBK} for both links.

An initial condition of (q_1, q_2)=$(-0.5, 1)$ rad is introduced for both links to study its effect on the proposed controller. Note that these initial conditions implicitly imply a non-zero initial torque τ_t at the flexible transmission (see formulation (6.2)). The controller provides a satisfactory transient response. The manipulator's position and velocity errors as well as the motor's



(a)

(b)

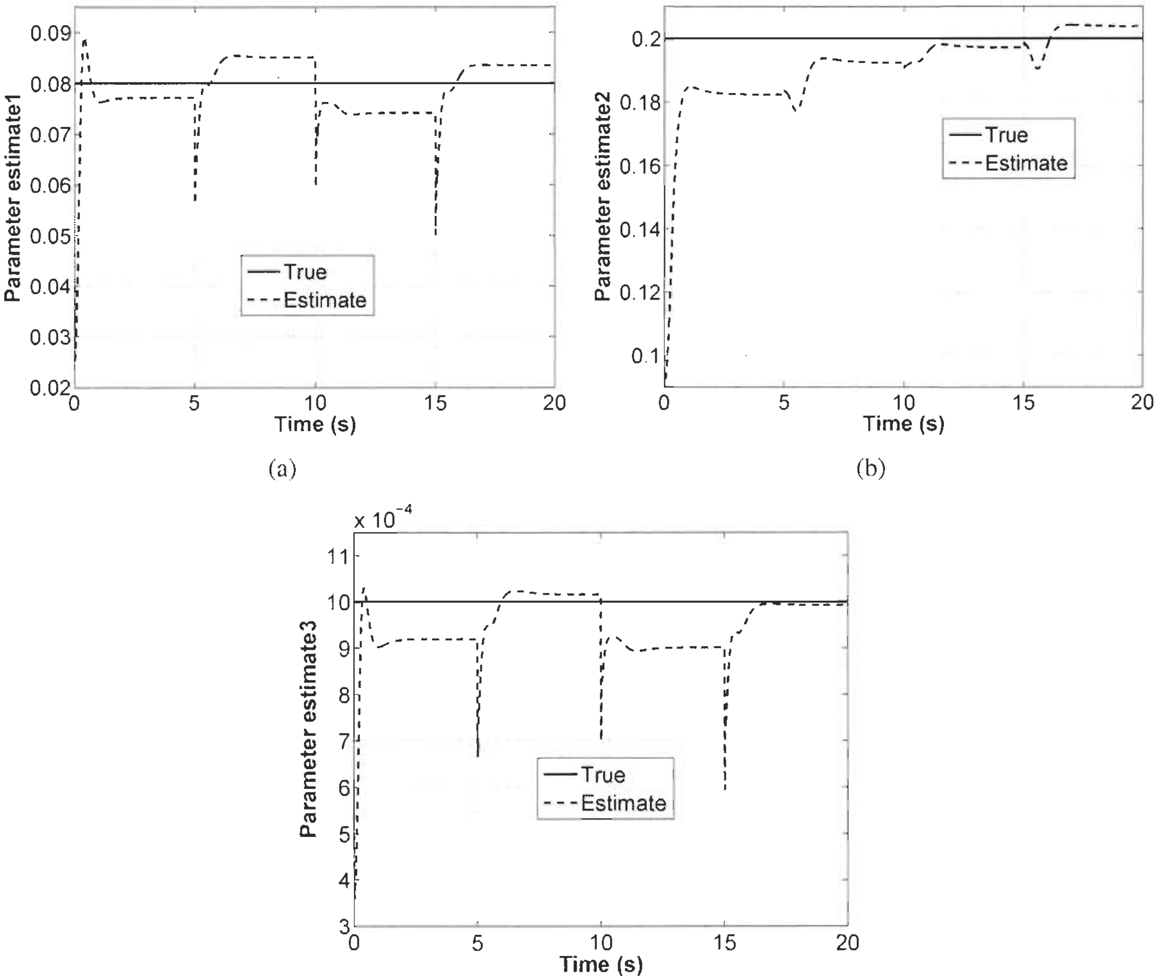
(c)

(d)

(e)

(f)

FIGURE 6.14 – Disturbance compensation response with nominal values : (a) manipulator's position error ; (b) manipulator's velocity error ; (c) manipulator's internal stability ; (d) controller's output torque τ_m ; (e) controller's torque τ_{FBK} for link1 ; and (f) controller's torque τ_{FBK} for link2.

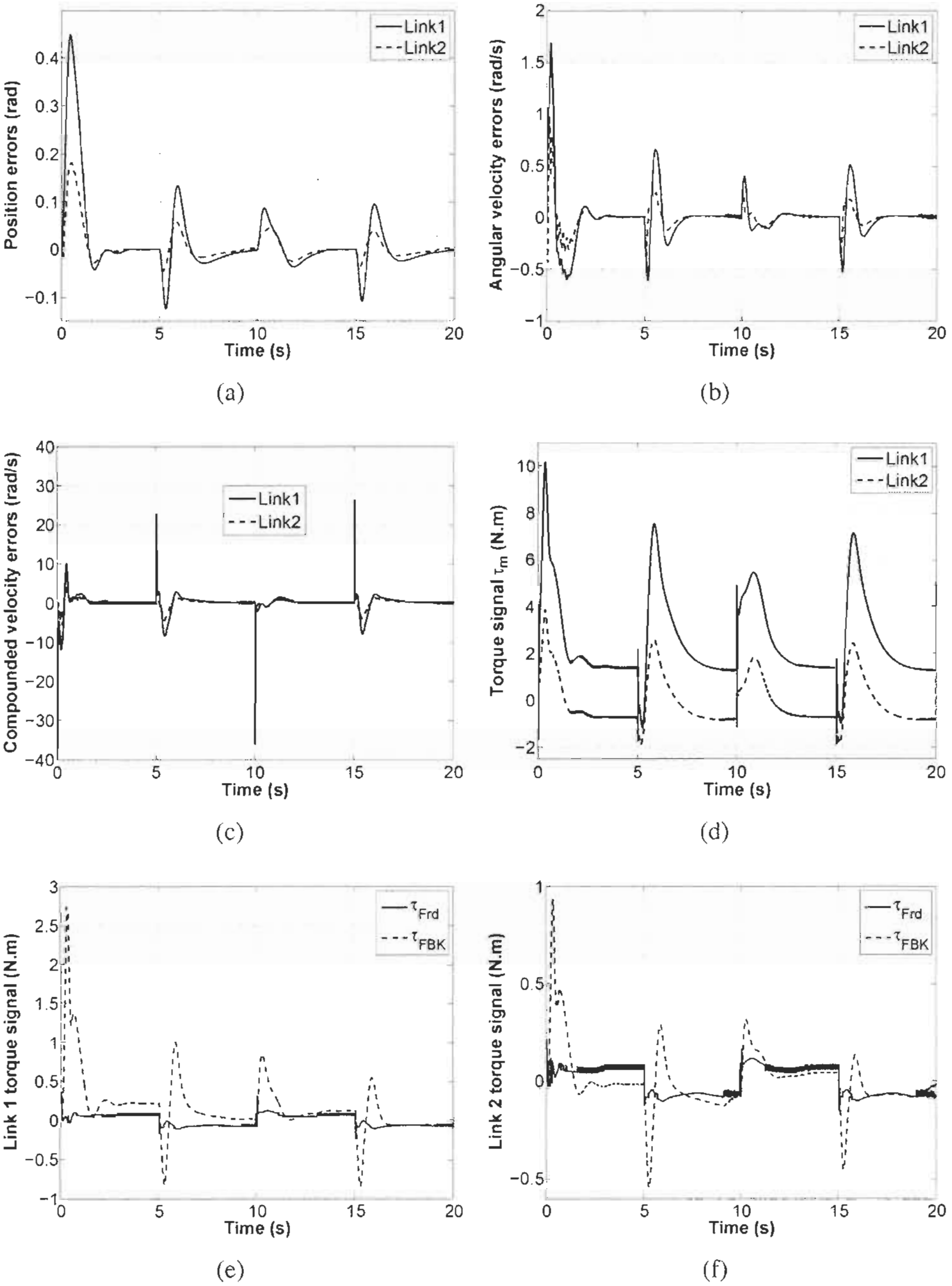


(a)

(b)

(c)

FIGURE 6.15 – Disturbance compensation parameters estimate : (a) $\hat{W} \simeq ml^2$; (b) $\hat{W} \simeq ml$; and (c) $\hat{W} \simeq J_m r$.



(a) (b) (c) (d) (e) (f)

FIGURE 6.16 – Disturbance compensation response with higher elasticity : (a) manipulator's position error ; (b) manipulator's velocity error ; (c) manipulator's internal stability ; (d) controller's output torque τ_m ; (e) controller's torque τ_{FBK} for link1 ; and (f) controller's torque τ_{FBK} for link2.

speed converge to zero. As it can be observed in Fig. 6.17(a), The controller compensates for the system's initial errors. The controller's characteristics beyond the first unit step are similar to those of Fig. 6.14. The controller is also compensating for the highly nonlinear friction terms by providing a smooth control torque signal.

In the following, an external disturbance is introduced. For that, the two links are being subjected to an external large magnitude disturbance $\tau_{dl} = 3\sin(0.2\pi t) + 1$, where t is the time index. It is worth pointing out the fact that the introduced external disturbance is not dependent on the system's measurable states $(q, \dot{q}, \dot{\theta})$, and hence it is not explicitly modeled in the design of the proposed controller. The controller's performance under such conditions is revealed in Fig. 6.18. Again, the end-effector and the links' position errors remain limited to the same range as in Fig. 6.14. The role of the feedback in annihilating the effect of the external disturbance is clearly shown in Fig. 6.18(e) and Fig. 6.18(f).

The end-effector trajectory in the world coordinate system for all cases is depicted in Fig. 6.19. A satisfactory end-effector's trajectory tracking is obtained for all cases.

6.6 Fuzzy Logic Control of Flexible-Joint Manipulators

The fuzzy control strategy is based on a human operator experience to interpret a situation and initiate its control action. A block diagram for the fuzzy controller is illustrated in Fig. 6.20. Given the desired control signals q_d and $\dot{q}_d$, the link's position error e_q and the compounded velocity error $\dot{e}_q$ are computed as in (6.15). The FLC takes these two inputs and provides a control action τ_{m} that is proportional to the input values. As shown in Fig. 6.21, these signals are quantized into 5 levels represented by a set of linguistic variables : Negative Large (NL), Negative Small (NS), Zero (Z), Positive Small (PS), and Positive Large (PL). In this study, triangular membership functions are used, mainly due to their high computational and performance efficiencies [62]. The fuzzy rules used by the controller are shown in Table 5.2 and can be refined by an expert to improve the control performance. Therefore, an empirical analysis for the fuzzy rules and the parameters of the input membership functions is performed to allow all rules of the fuzzy inference engine to fire. However, these procedures

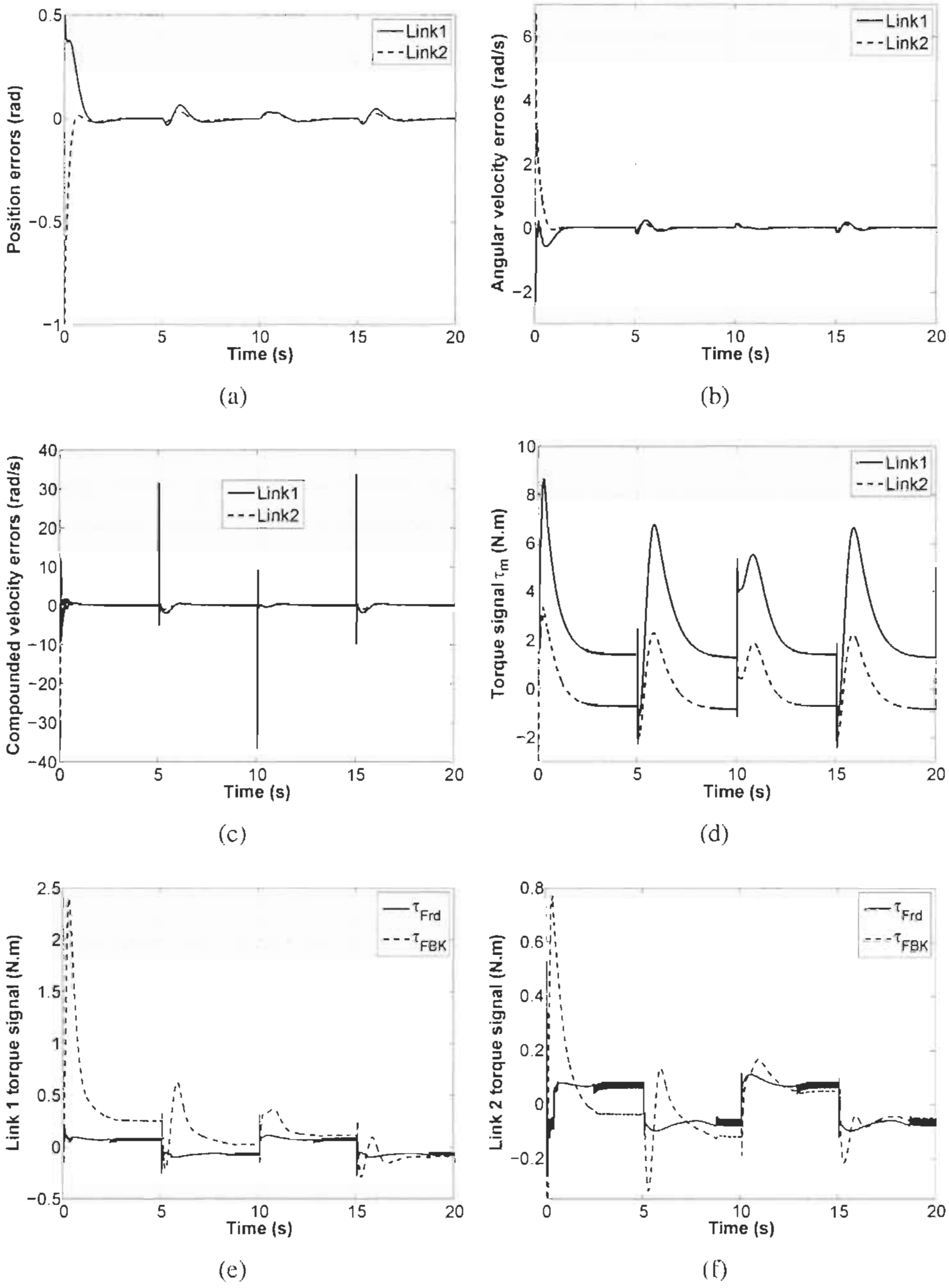
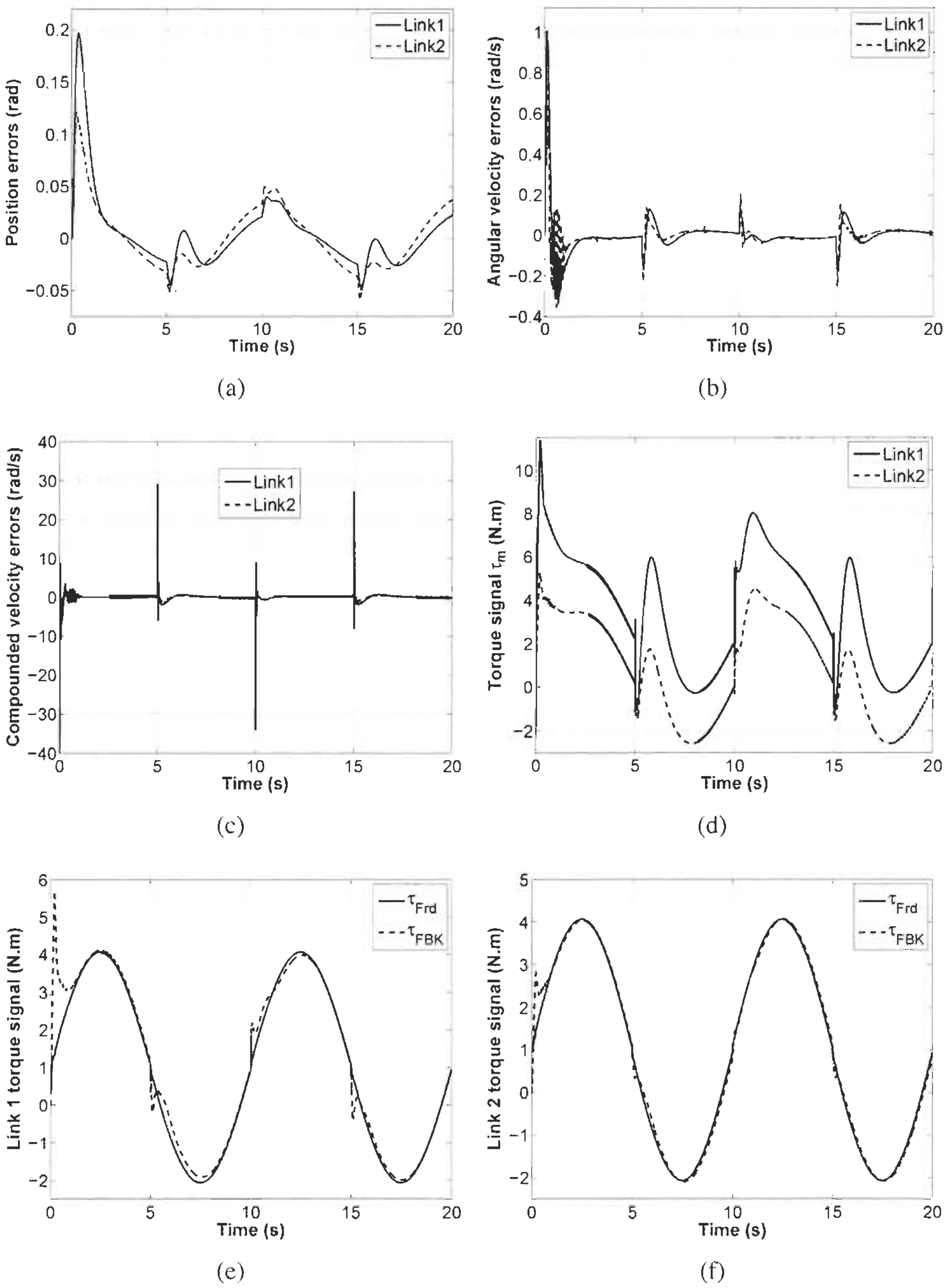


FIGURE 6.17 – Disturbance compensation response with initial conditions (q_1, q_2)=$(-0.5, 1)$: (a) manipulator's position error ; (b) manipulator's velocity error ; (c) manipulator's internal stability ; (d) controller's output torque τ_m ; (e) controller's torque τ_{FBK} for link1 ; and (f) controller's torque τ_{FBK} for link2.



(a) (b) (c) (d) (e) (f)

FIGURE 6.18 – Disturbance compensation response with higher disturbance magnitude : (a) manipulator's position error ; (b) manipulator's velocity error ; (c) manipulator's internal stability ; (d) controller's output torque τ_m ; (e) controller's torque τ_{FBK} for link1 ; and (f) controller's torque τ_{FBK} for link2.

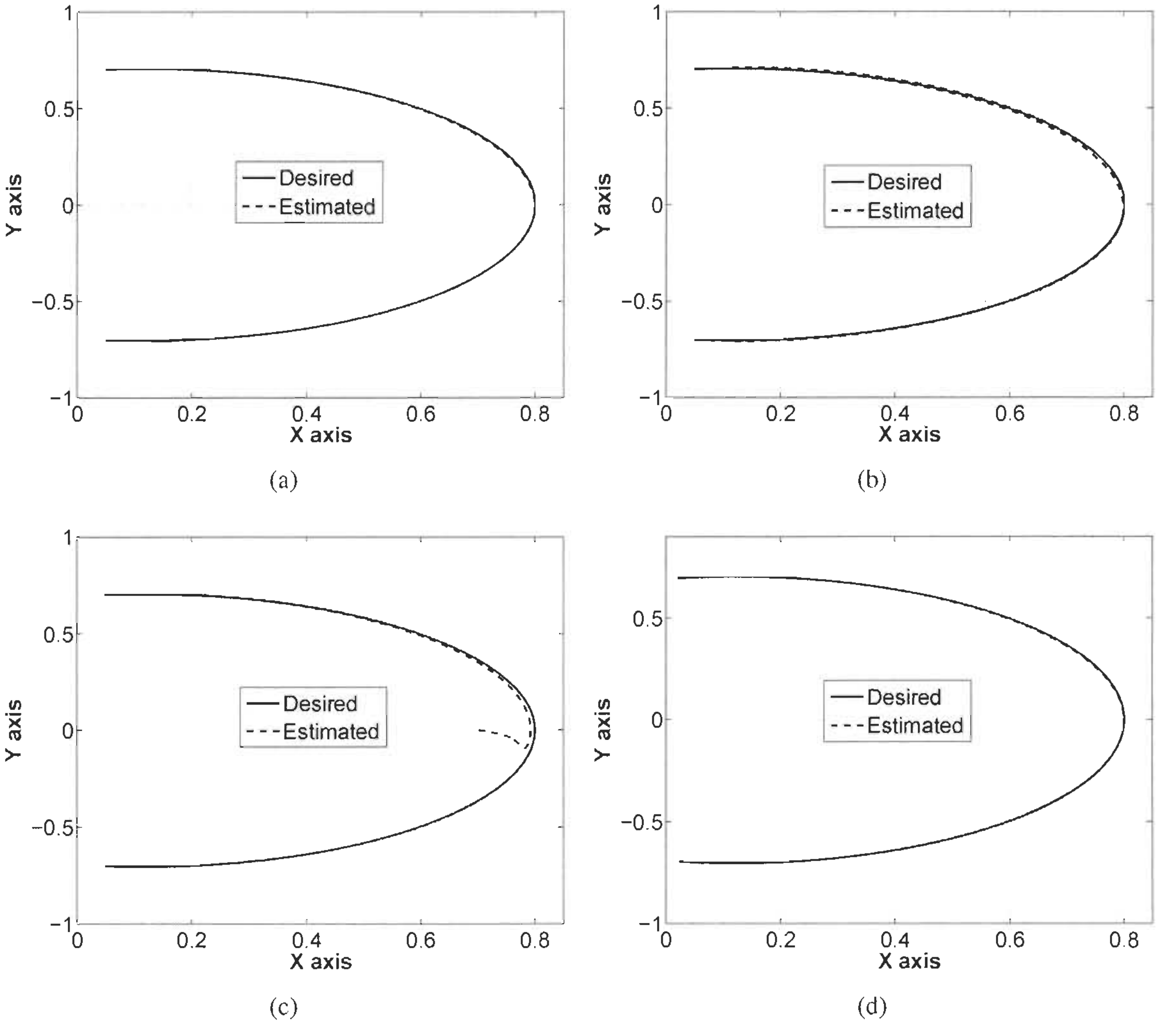


FIGURE 6.19 – Disturbance compensation end-effector trajectory for : (a) Fig. 6.14 ; (b) Fig. 6.16 ; (c) Fig. 6.17 ; and (d) Fig. 6.18.

are conducted ahead of time and are not part of the control cycle.

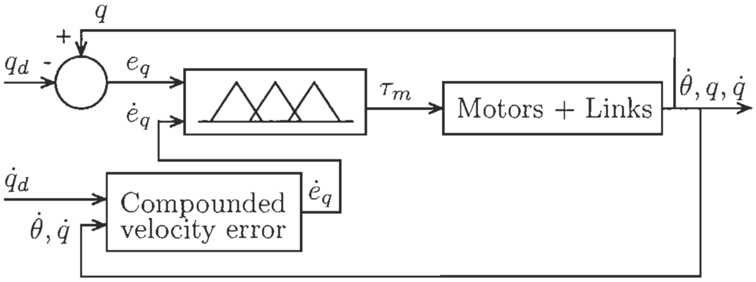


FIGURE 6.20 – Fuzzy logic control scheme.

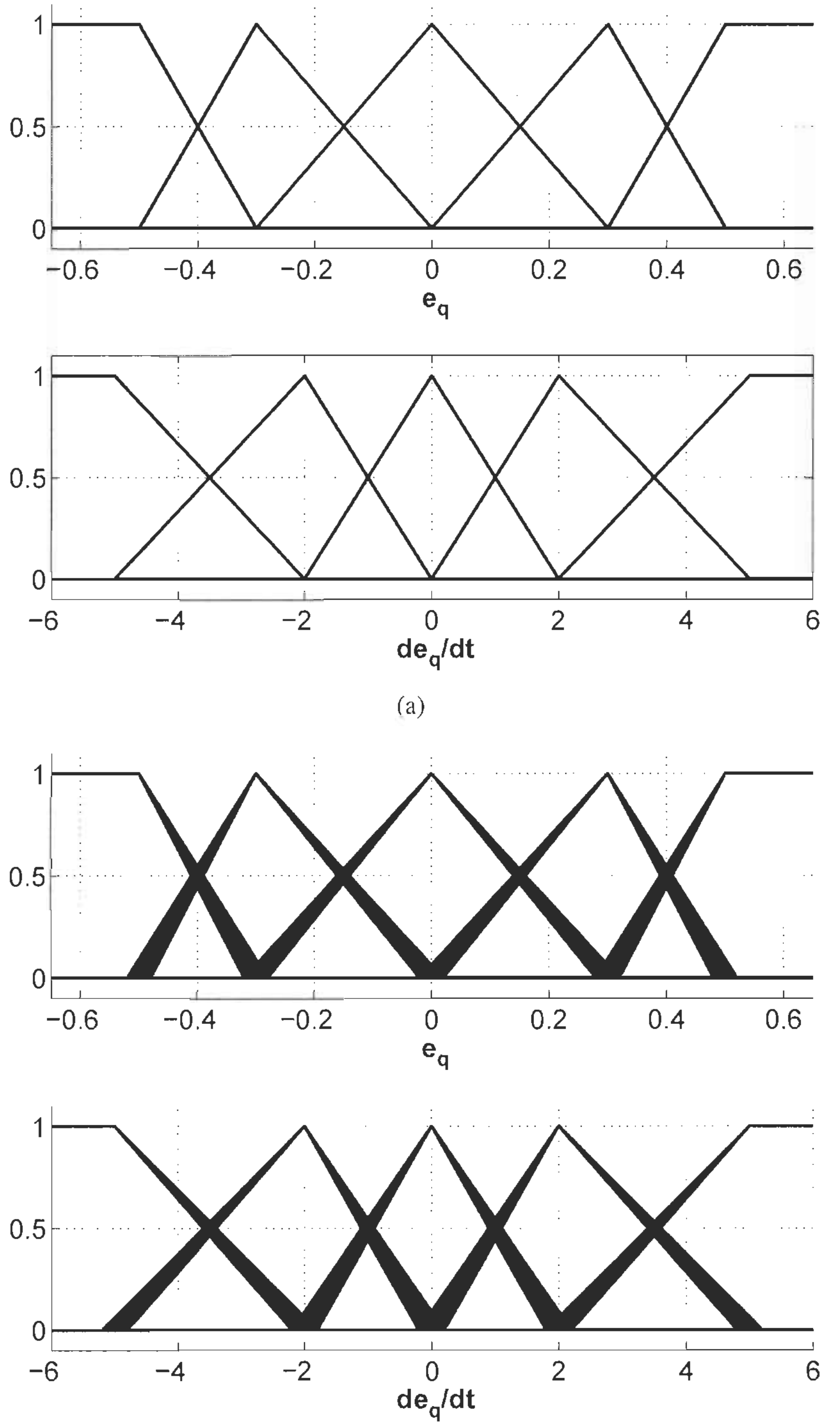
6.6.1 Results

To demonstrate the performance of the proposed controller, the controller is validated through simulation and by comparing type-2 against its type-1 counterpart. For that, five simulation sets are carried out on a single link flexible-joint manipulator. In all these simulations, the system's dynamics is assumed to be *a priori* unknown. The stiffness coefficient and gear ratio are set to be $K = 7$ N·m/rad and $r = 1$, respectively. The link's mass and length are taken as $m = 0.21$ kg and $l = 0.3$ m, respectively. The manipulator's dynamics in terms of its physical parameters is defined by : $M(q) = 5.05 \cdot 10^{-2}$ kg·m^2, $C(q, \dot{q}) = 0$, and $G(q) = mgl\sin(q)$, where $g = 9.8$ m/s^2 is the gravitational constant.

The manipulator's desired position and velocity trajectories are shown in Fig. 6.22. The control structure scheme and the system's model are implemented in Simulink$^{\text{TM}}$ while the fuzzy control engines are programmed in C.

In the first simulation, the aforementioned nominal values are used to simulate the manipulator's dynamics. The results are shown in Fig 6.23. As expected, both controllers lead to a satisfactory performance in this case as they both attenuate the position and velocity errors to zero within comparable time delays. However, this comparable performance is achieved by type-2 FLC with a much smoother control effort as compared to type-1 FLC.

Initial conditions are introduced in the second simulation to study their effect on the controllers' performances. The results are depicted in Fig 6.24. The performance of the type-1



(a)

(b)

FIGURE 6.21 – Fuzzy membership functions : (a) type-1 ; and (b) type-2.

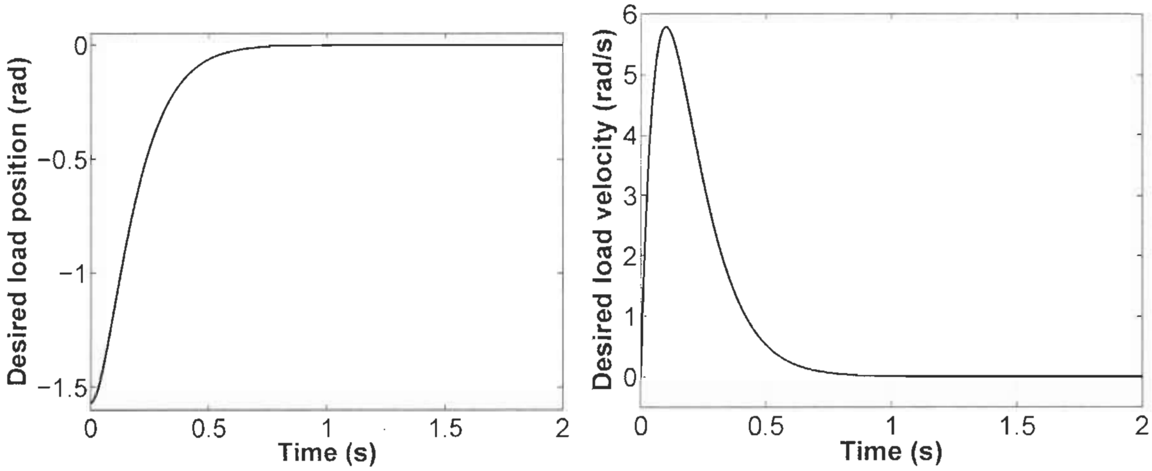
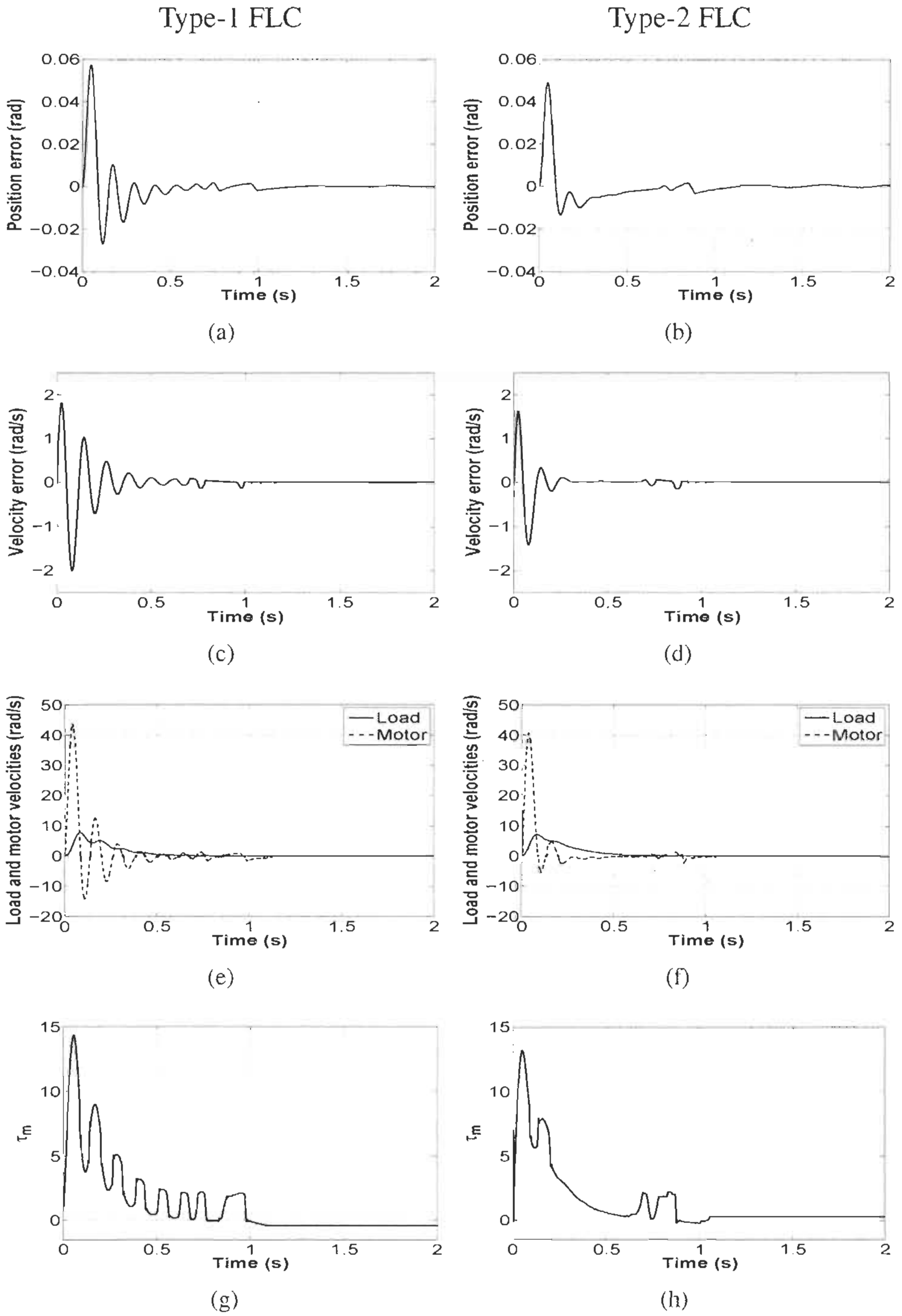


FIGURE 6.22 – Manipulator's position and velocity reference signals

FLC is not affected much by the different initial condition, in the sense that it could still decay the manipulator's position and velocity errors to zero in less than 1 sec. Nevertheless, the superiority of the type-2 FLC is revealed more clearly in this simulation. All error signals converge more steadily to zero under the type-2 fuzzy control scheme with almost half the control effort.

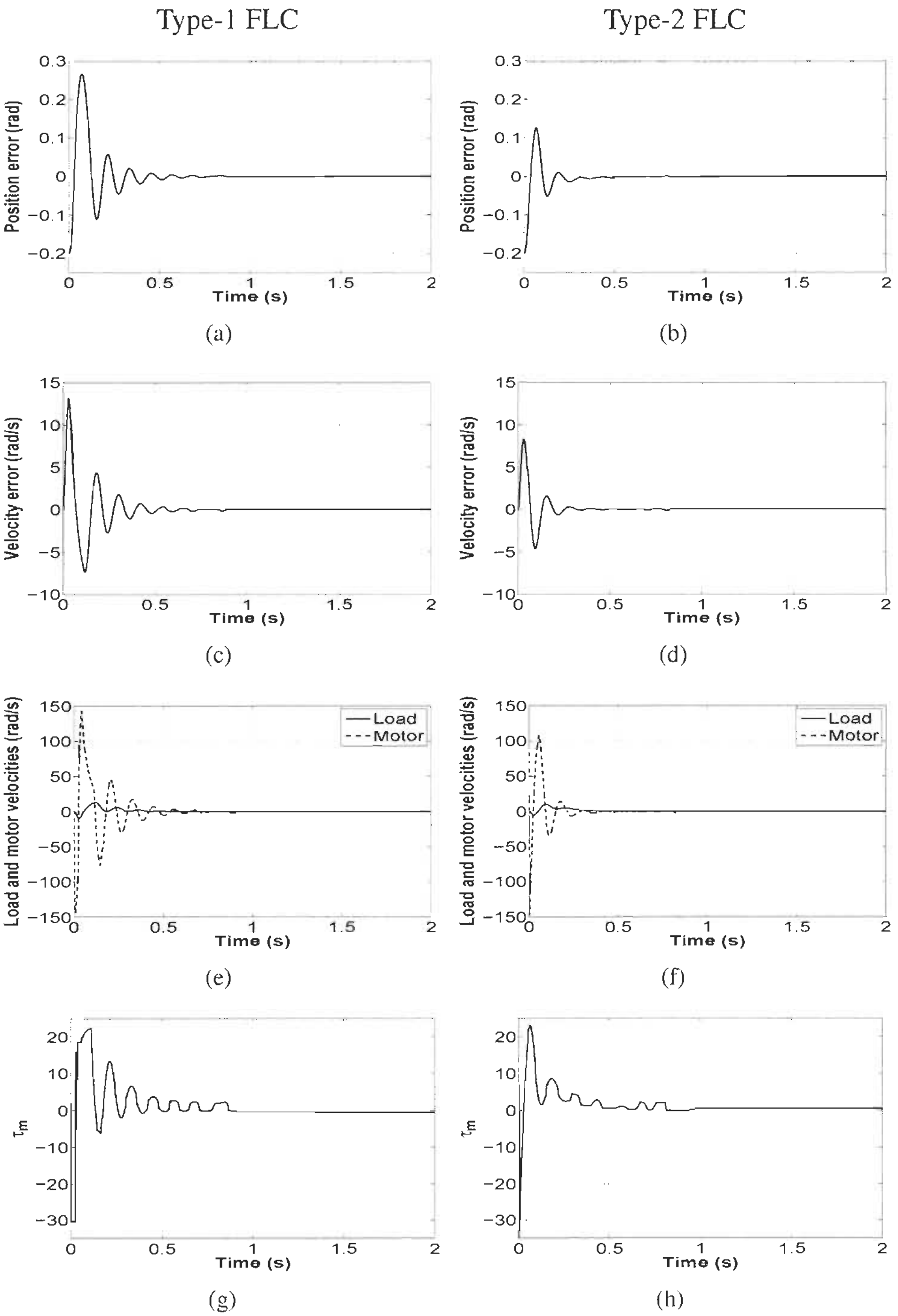
To study the ability of controllers to sustain various types and intensities of uncertainties, the following simulations are carried out. For this purpose, the load's inertia is doubled. The results are shown in Fig 6.25. The manipulator's position and velocity errors remain within an amplitude range similar to that of simulation 1 (see Fig 6.23). However, the error signals and the controller's output under type-1 FLC are fairly fluctuating as opposed to a smooth and steady convergence behavior with the type-2 FLC. It is quite important to notice here the degradation in the actuator's internal stability (Fig. 6.25(e)) under type-1 FLC despite the settling of the load's velocity. The superiority of the type-2 FLC in compensating for such a type of uncertainty is manifested with lower load position and speed errors, over its type-1 counterpart.

In the fourth simulation, all load friction terms are amplified 5 times with respect to their nominal values (Table 6.1). The results are illustrated in Fig 6.26. As can be seen, type-1 FLC is able to compensate for the system's uncertainty but at a lower performance measure than



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 6.23 – FLC response with nominal values : (a)–(b) position error ; (c)–(d) velocity error ; (e)–(f) motor's velocity vs. manipulator's velocity ; and (g)–(h) controller's torque (τ_m).



(a) (b)

(c) (d)

(e) (f)

(g) (h)

FIGURE 6.24 – FLC response with initial conditions : (a)–(b) position error ; (c)–(d) velocity error ; (e)–(f) motor's velocity vs. manipulator's velocity ; and (g)–(h) controller's torque (τ_m).

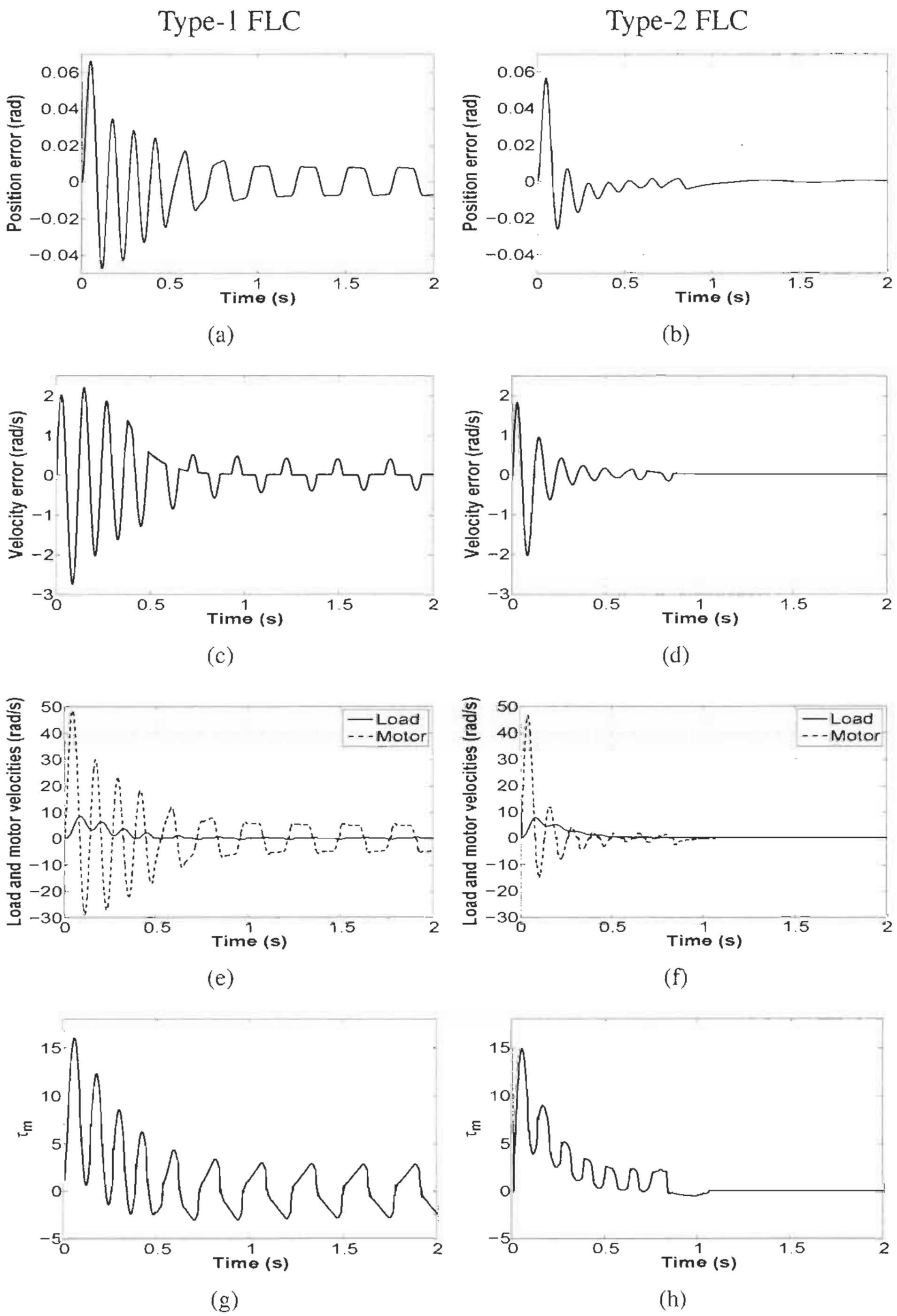


FIGURE 6.25 – FLC response with twofold inertia : (a)–(b) position error ; (c)–(d) velocity error ; (e)–(f) motor's velocity vs. manipulator's velocity ; and (g)–(h) controller's torque (τ_m).

that produced by type-2 FLC.

In the last simulation, an external time-varying disturbance $\tau_{dl} = 0.25\sin(30\pi t) + 0.4$ is introduced on the link, where t is the time index. It is worth pointing out the fact that the introduced external disturbance is not dependent on the system's measurable states $(q, \dot{q}, \dot{\theta})$, and hence it is not explicitly modeled in the design of the proposed controller. The controller's performance under such conditions is revealed in Fig 6.27. As in simulation 3, type-1 FLC is able to maintain bounded error signals but fail to make them converge to a practically insignificant margin. This is especially clear with the load's velocity error (Fig. 6.27(c)) and the motor's internal stability (Fig. 6.27(e)). On the other hand, we can see that type-2 FLC does indeed deem down these signals to practically zero. In addition, this is accomplished with a much smoother control signal τ_m and control effort than those generated by type-1 FLC.

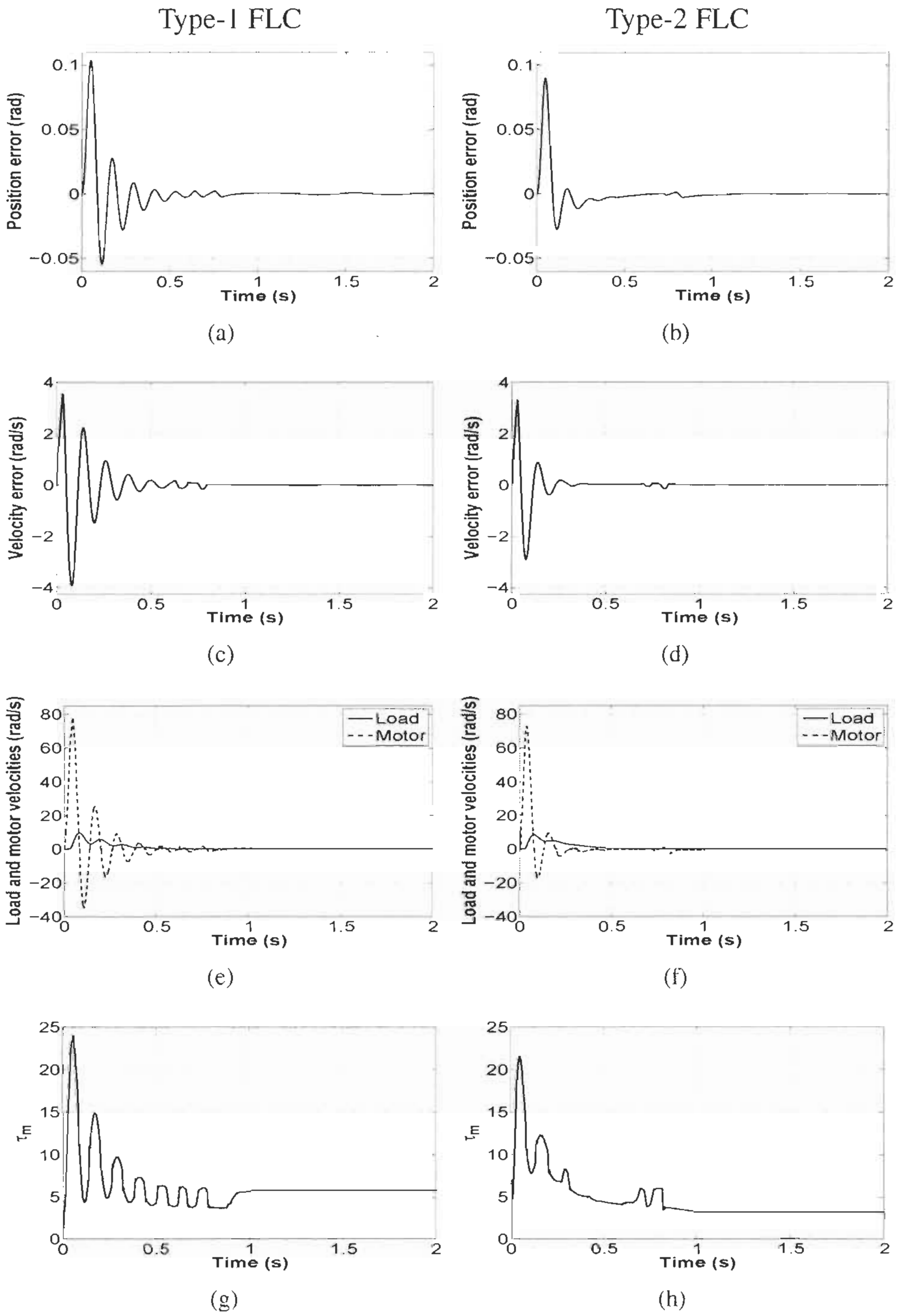
6.7 Adaptive Type-2 Fuzzy Logic Control for Flexible-Joint Manipulators

The adaptive fuzzy control scheme is illustrated in Fig. 6.28 and the adaptive type-2 FLC structure is depicted in Fig. 6.29. It consists of four layers. Input nodes and type-2 fuzzification nodes are shown in layer 1 and layer 2, respectively, forming the antecedent part of the fuzzy rules. Consequent parts are represented by layer 3 and 4, which are constructed with fuzzy rule nodes and output nodes. They are linked by interval weighting factors $[w^i_{lj}\ w^i_{rj}]$, $i = 1 \ldots m$ and $j = 1 \ldots z$.

The adaptive type-2 FLC's output can be written as,

$$Y = \Phi^T W + \varepsilon = \hat{\Phi}^T \hat{W}$$

where $\hat{W} \in \mathbb{R}^{z \times m}$ is a weight matrix, $\hat{\Phi}$ is a m-dimensional vector of known functions (regressor), and $\varepsilon = \hat{\Phi}^T \hat{W} - \Phi^T W$ is the adaptive type-2 FLC error.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

FIGURE 6.26 – FLC response with magnified friction : (a)–(b) position error ; (c)–(d) velocity error ; (e)–(f) motor's velocity vs. manipulator's velocity ; and (g)–(h) controller's torque (τ_m).

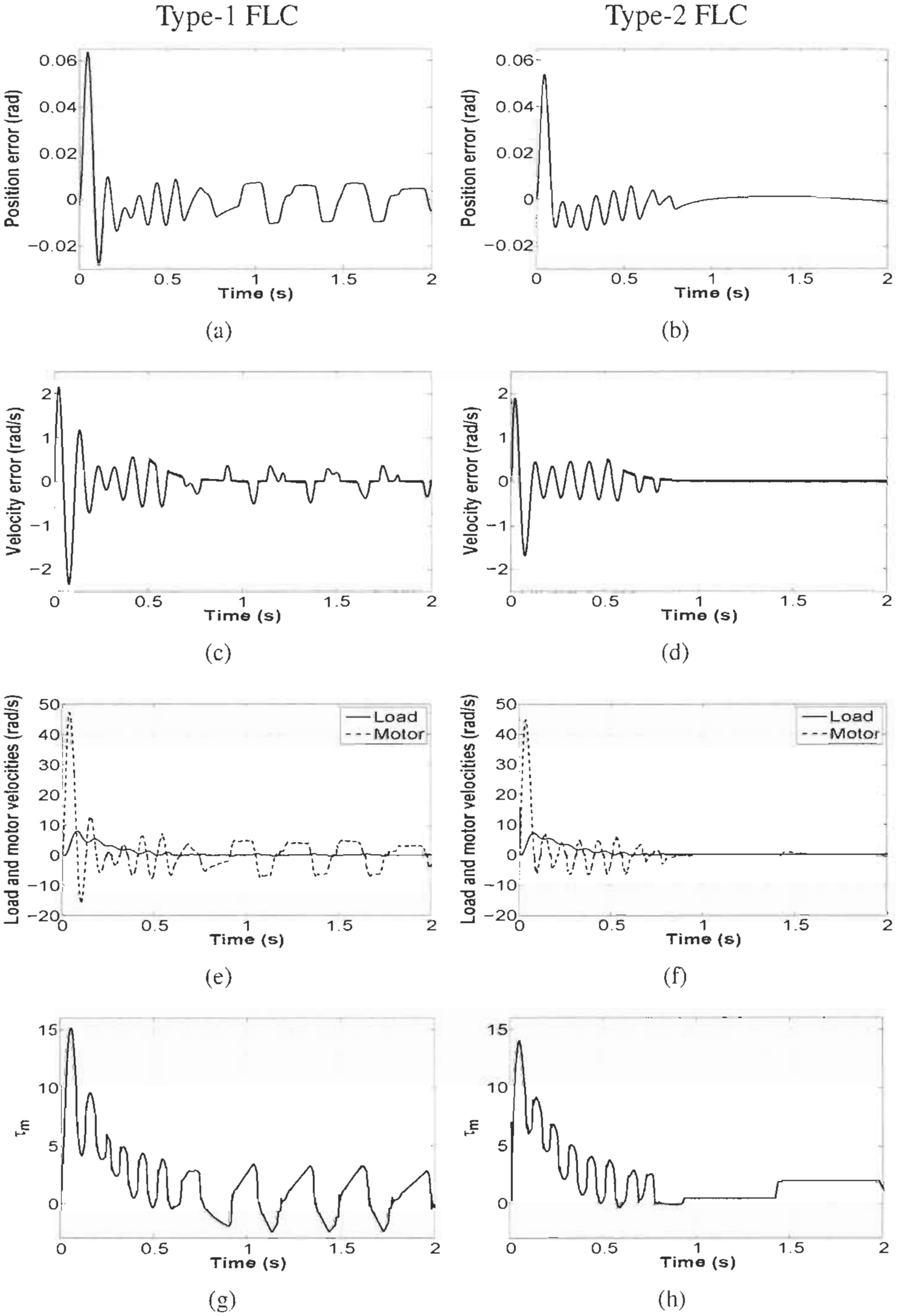


FIGURE 6.27 – FLC response with load disturbance : (a)–(b) position error ; (c)–(d) velocity error ; (e)–(f) motor's velocity vs. manipulator's velocity ; and (g)–(h) controller's torque (τ_m).

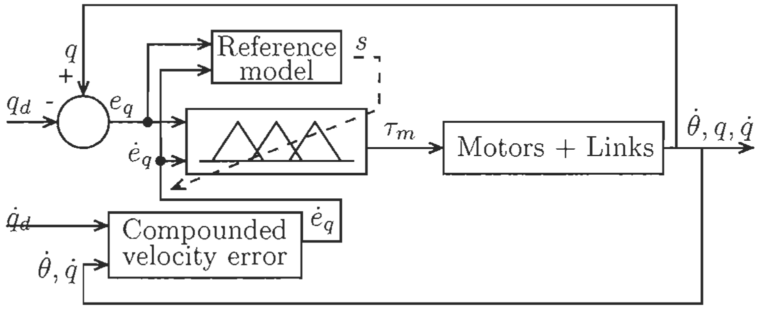


FIGURE 6.28 – Adaptive fuzzy logic control scheme.

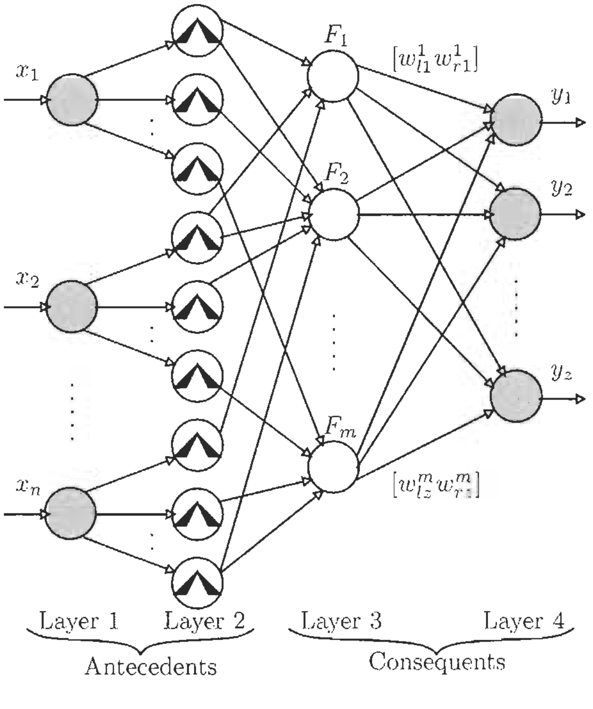


FIGURE 6.29 – Adaptive Type-2 fuzzy logic control structure

Recall the following model (6.23),

$$M_t(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_{Frd} = r\tau_m$$

where,

$$M_t(q) = J_m r^2 + M(q)$$

$$\tau_{Frd} = r\tau_{fm} + r\tau_{dm} + \tau_{fl} + \tau_{dl}$$

The model can be represented by a linear regression using the linear in parameters property of the manipulator dynamics,

$$M_t(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r + G(q) = \Phi(\ddot{q}_r, \dot{q}_r, \dot{q}, q)^T W$$

The control law is :

$$r\tau_m = \hat{\Phi}^T \hat{W} - K_D s \tag{6.24}$$

where, K_D is a positive diagonal matrix gain and the sign $\hat{\bullet}$ denotes the parameter estimate vector. It is noteworthy that the ability of type-2 FLCs to cope with uncertainties of larger magnitudes makes that no additional controller is required to damp the oscillations of the joint elasticity as in [52].

Theorem 11 *Consider the nonlinear system in (6.2) under the assumption of high stiffness so that singular perturbation applies with reference signal (6.15) and control law (6.24). The adaptive control law is stable in the sense of Lyapunov with the following adaptation law,*

$$\dot{\hat{W}} = -\Gamma\, \hat{\Phi}\, s$$

where $\Gamma = diag(\gamma_1, \gamma_2, \ldots, \gamma_n)$ and γ_i is a positive constant, $i = 1, \ldots, n$. Therefore, the manipulator's position q and velocity $\dot{q}$ converge to their pre-defined time-dependent desired values q_d and $\dot{q}_d$, respectively.

Proof 5 *Take the derivative of (6.17),*

$$\dot{s} = \ddot{q} - \ddot{q}_r$$

$$M_t(q)\dot{s} = M_t(q)\ddot{q} - M_t(q)\ddot{q}_r$$

Substituting $M_t(q)\ddot{q}$ from (6.23),

$$M_t(q)\dot{s} = r\tau_m - M_t(q)\ddot{q}_r - C(q,\dot{q})\dot{q}_r - G(q) - \tau_{Frd} - C(q,\dot{q})s$$

The linear in parameters property yields

$$M_t(q)\dot{s} = r\tau_m - \Phi^T W - \tau_{Frd} - C(q,\dot{q})s \tag{6.25}$$

Set,

$$r\tau_m = \hat{\Phi}^T \hat{W} - K_D s$$

Equation (6.25) becomes

$$M_t(q)\dot{s} = \varepsilon - \tau_{Frd} - C(q,\dot{q})s - K_D s$$

where, $\varepsilon = \hat{\Phi}^T \hat{W} - \Phi^T W$. Add and substract $\hat{\Phi}^T W$,

$$\varepsilon = \hat{\Phi}^T \hat{W} - \hat{\Phi}^T W + \hat{\Phi}^T W - \Phi^T W$$

Therefore,

$$\varepsilon = \hat{\Phi}^T \tilde{W} + \tilde{\Phi}^T W \tag{6.26}$$

where, $\tilde{\Phi} = \hat{\Phi} - \Phi$ and $\tilde{W} = \hat{W} - W$.

Choose the following Lyapunov candidate,

$$V = \frac{1}{2}\{s^T M_t(q)s + \tilde{W}^T \Gamma^{-1} \tilde{W}\}$$

Taking the time-derivative of V *:*

$$\dot{V} = s^T M_t(q)\dot{s} + \frac{1}{2} s^T \dot{M}_t(q)s + \tilde{W}^T \Gamma^{-1} \dot{\hat{W}}$$

Since the unknown weights W *are constants,* $\dot{\tilde{W}} = \dot{\hat{W}}$.

Substituting for $M_t(q)\dot{s}$ *and using the skew-symmetry property,*

$$\dot{V} = s^T \varepsilon - s^T \tau_{Frd} + \tilde{W}^T \Gamma^{-1} \dot{\hat{W}} - s^T K_D s$$

Substitute ε *from (6.26),*

$$\dot{V} = s^T \hat{\Phi}^T \tilde{W} + s^T \tilde{\Phi}^T W - s^T \tau_{Frd} + \tilde{W}^T \Gamma^{-1} \dot{\hat{W}} - s^T K_D s$$

Setting the adaptation law as,

$$\dot{\hat{W}} = -\Gamma \, \hat{\Phi} \, s$$

leads to,

$$\dot{V} = s^T \Upsilon - s^T K_D s$$

where $\Upsilon = \tilde{\Phi}^T W - \tau_{Frd}$ *is assumed to be upper bounded by a positive constant* δ*, i.e.,* $|\Upsilon| \leq \delta$*. Thus,*

$$\dot{V} \leq |s^T|\delta - s^T K_D s$$

Therefore, it is possible to choose K_D *so that* $\dot{V} \leq 0$*, except possibly in a neighborhood of* $s = 0$*. Then, the system is stable in the sense of Lyapunov. The neighborhood of* $s = 0$ *is a region defined by* δ *and gets smaller as* $\delta \to 0$.

6.7.1 Results

Two simulation sets are carried out on a single link flexible-joint manipulator to highlight the proposed type-2 FLC as opposed to type-1 in tolerating a higher degree of parametric and modeling uncertainties. In both simulations, the system's dynamics is assumed to be *a priori* unknown. For each simulation, the system's response is studied taking into account the manipulator's position and velocity errors, the joint's internal stability, and the controller's output torque, τ_m. The manipulator's desired position trajectory is taken as the step response of a critically damped second order system with a natural frequency of 3 rad/s, as shown in Fig. 6.30.

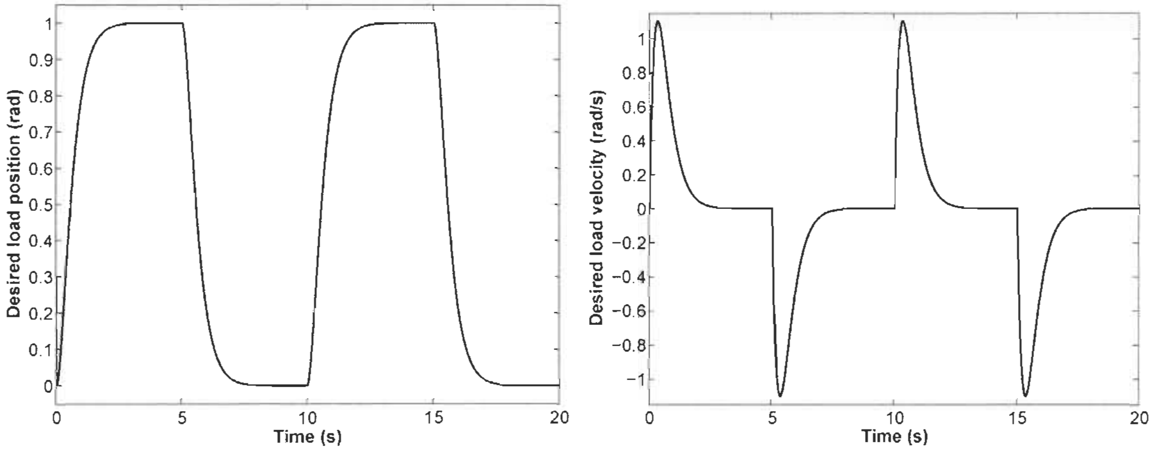


FIGURE 6.30 – Manipulator's position and velocity reference signals

The first simulation is meant to study the ability of controllers to sustain various types and magnitudes of load uncertainties. For this purpose, the load's inertia and the link's mass are both doubled abruptly at time 5 sec. and returned back to their original values at time 15 sec. of the simulation. The results are shown in Fig 6.31. A slight increase is noticed in the manipulator's position and velocity errors due to a heavier load. However, the error signals and the controller's output under type-1 FLC are fairly fluctuating as opposed to a smooth and steady convergence behavior with the type-2 FLC. It is quite important to notice here the degradation in the actuator's internal stability (Fig. 6.31(e)) under type-1 FLC despite the settling of the load's velocity. The superiority of the adaptive type-2 FLC in compensating for such a type of uncertainty is manifested with a better load position, speed accuracy and control effort performance, over its type-1 counterpart.

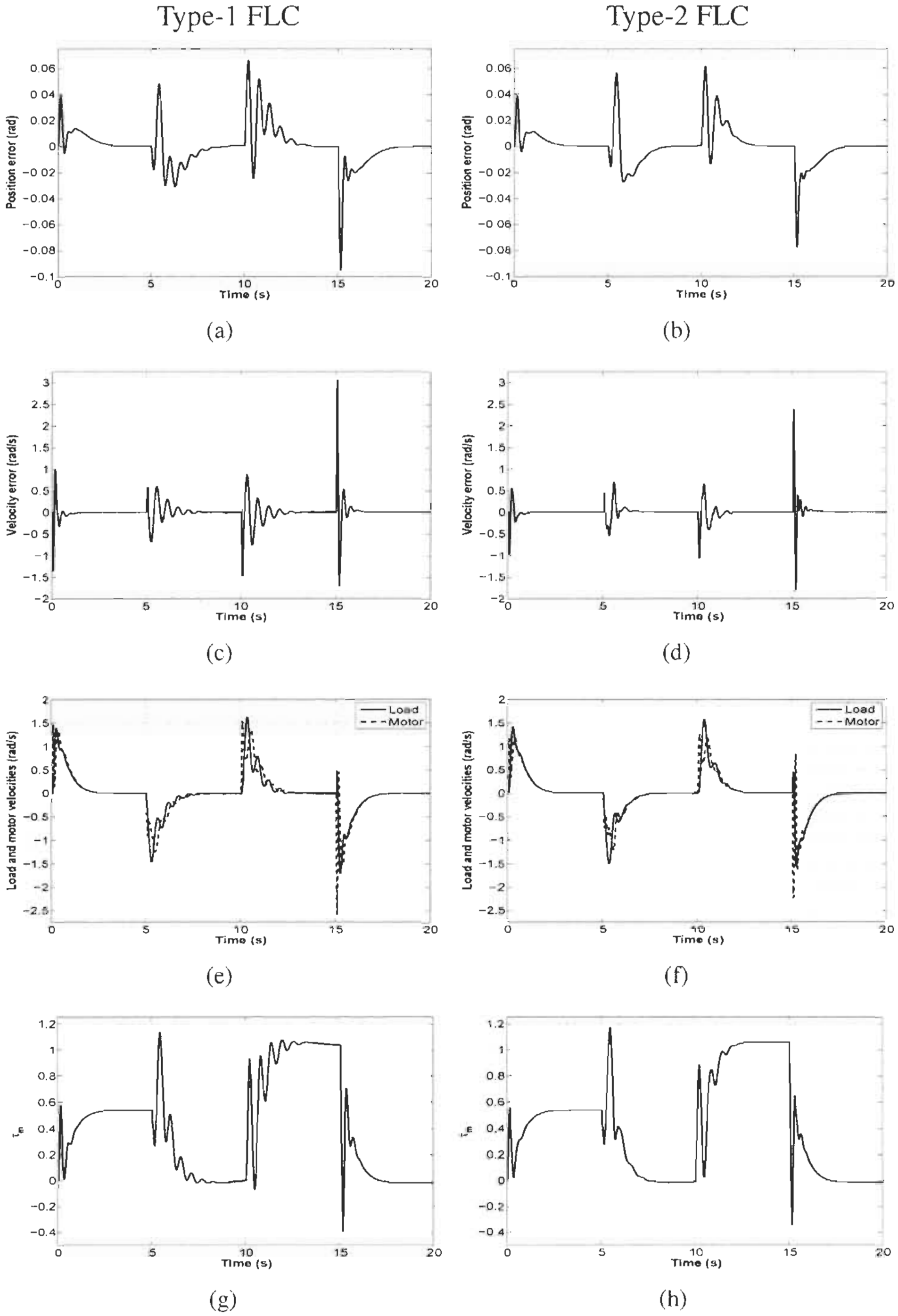
In the second simulation, the elastic joint's stiffness coefficient is changed abruptly to $K = 5$ N·m/rad at time 5 sec. and returned back to its original value ($K = 7$ N·m/rad) at time 15 sec. of the simulation. The controller's performance under such conditions is revealed in Fig 6.32. As in simulation 1, adaptive type-1 FLC is able to maintain bounded error signals but fails to make them converge smoothly. This is especially clear with the load's position and velocity errors (Fig. 6.32(c) and 6.32(a)) and the motor's internal stability (Fig. 6.32(e)). On the other hand, it is clear that adaptive type-2 FLC does indeed deem down these signals smoother and with less control effort than its type-1 counterpart.

6.8 Conclusion

Adaptive control for rigid robotic manipulators is considered as a starting point for the design of robust adaptive control techniques for flexible-joint manipulators. This has been made possible with the singular perturbation theory. However, it assumes high stiffness. Moreover, the presence of friction and disturbance affects significantly the performance and the stability. Under the assumption of high stiffness, a rigid-based feedforward controller approximates the inverse model of the flexible-joint manipulator. On the other hand, adaptive friction and disturbance compensators are used as a feedback to reduce the effect of residual errors left from the forward stage. As such, the proposed feedback controllers can be used with many conventional rigid manipulators approaches in the literature. Furthermore, a trade-off criterion strategy is also used to improve the manipulator's internal stability. Simulation results highlight the quality of compensation of flexibility, friction nonlinearities, and disturbance. Accurate tracking of the desired load trajectory is obtained and internal stability, a potential problem with such a system, is also achieved. Furthermore, the stability is proven by Lyapunov direct method.

On the other hand, a type-2 FLC and an adaptive type-2 FLC are developed for the control of flexible-joint manipulators in the presence of dynamical modeling and parametric uncertainties of various magnitudes. To alleviate the naturally inherited high computational complexity of type-2 FLCs, interval membership functions are adopted. Unlike other control methodologies suggested in the literature, the proposed controllers are independent of the noisy acceleration and torque signals. Both controllers are also compared to their type-1 counterpart in similar operating conditions. Simulation results show the superiority of type-2 FLC in damping the oscillations of the joint elasticity and compensating for high-magnitude uncertainties. This finding confirms the theoretical credentials associated to type-2 FLSs in their higher tolerance to the imprecise modeling of fuzzy controllers ; namely the fuzzy membership functions and knowledge base. Thus far, this work is one of the scarce attempts in designing and implementing type-2 fuzzy logic control architecture for flexible-joint manipulators.

Robotic manipulators are not only driven by DC motors, AC machines can also be used for such systems. However, their performance is also limited by friction nonlinearities as well as by load uncertainties. Therefore, several soft-computing based adaptive control techniques are presented in the next chapter for speed control and estimation of permanent magnet synchronous machines (PMSMs).



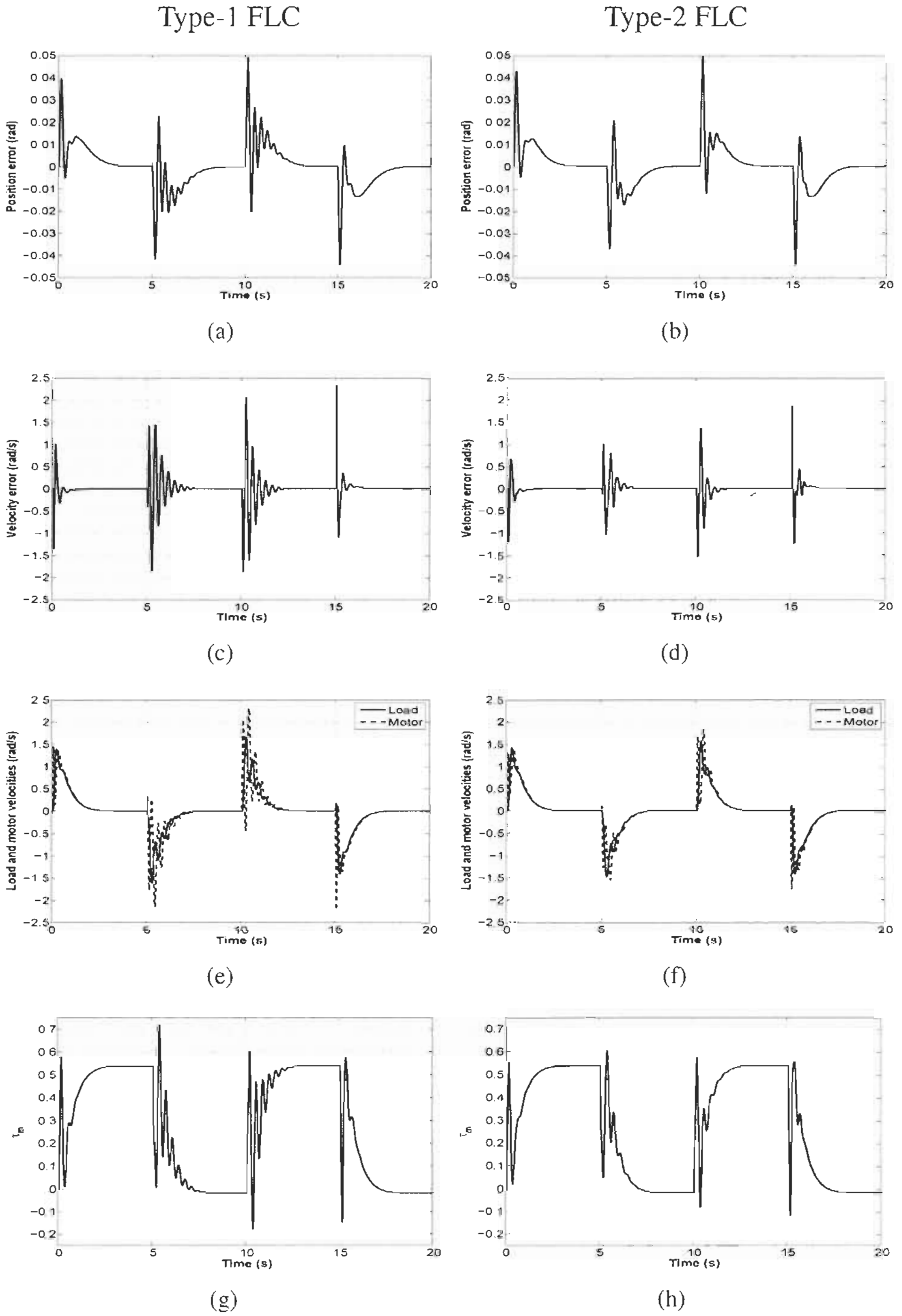
(a) (b)

(c) (d)

(e) (f)

(g) (h)

FIGURE 6.31 – Adaptive type-1 and type-2 FLC responses to varying load's mass and inertia : (a)–(b) position error ; (c)–(d) velocity error ; (e)–(f) motor's velocity vs. manipulator's velocity ; and (g)–(h) controller's torque (τ_m).



(a) (b)

(c) (d)

(e) (f)

(g) (h)

FIGURE 6.32 – Adaptive type-1 and type-2 FLC responses to varying stiffness : (a)–(b) position error ; (c)–(d) velocity error ; (e)–(f) motor's velocity vs. manipulator's velocity ; and (g)–(h) controller's torque (τ_m).

Chapitre 7

Lyapunov-Based Control of Permanent Magnet Synchronous Machines

7.1 Introduction

Permanent magnet synchronous machines (PMSMs) are widely used in many industrial applications, thanks to their compact size, high efficiency, high power density, large torque to inertia ratio and low rotor losses [63, 64]. These advantages make them good candidates for high-performance applications, such as electric vehicles, wind turbines and robotics. However, in order to operate efficiently, high resolution rotor angle encoder is required to generate smooth torque. These high resolution position sensors attached to the shaft of the rotor add length to the machine, raise cost, increase rotor inertia and require additional cabling. Recently, sensorless PMSM drives have received increasing interest for industrial applications where there are limitations on the use of a position sensor. Furthermore, sensorless control for motor drives reduces susceptibility to noise and vibration, cost, size and maintenance while increasing the overall system's reliability and robustness. However, controlling such systems still faces numerous challenges that need to be addressed such as, varying operating conditions, structured and unstructured dynamical uncertainties, and external disturbances.

Over the years, researchers attempted various estimation techniques. The rotor flux linkage estimation method [65] is used for its simplicity. This estimation method integrates the machine terminal voltages and the stator resistance drop to derive the rotor position from these quantities. However, integration introduces a drift problem and poor precision at low speed. Furthermore, stator resistance is known to vary with temperature. Extended Kalman filters (EKF) [66, 67] have been successfully and intensively used for rotor speed and position estimation in sensorless drives. Although EKF is a well-known approach, they are computationally intensive with the calculation of Jacobian matrices and require proper initialization to avoid instability which can occur due to linearization. This approach, as many methods based on state observers, uses a linearization of the system model around operating points. However, linearization of the nonlinear equations describing the drive's behavior along the nominal state trajectory does not guarantee the overall stability. Model-based observers [68, 69] have been widely used, however, they suffer from sensitivity to mechanical parameters such as load torque, friction and inertia parameters that are often changing dynamically or are unknown. Moreover, the technique requires a preliminary rotor alignment. The model reference adaptive system (MRAS) is used widely [69] for position and speed estimation. MRAS uses reference and parameter adjustable models. The two models have the same output and the error of two outputs is used to adjust, by the adaptation law, the model parameters. Thus, the actual output of the reference model follows the one of the plant. A sliding mode speed observer is proposed in [70]. This method suffers from poor low-speed performance, and robustness to parameter variations and uncertain load disturbances is obtained only when sliding mode truly occurs. In principle, sliding mode is achieved by discontinuous control and switching at infinite frequency. However, switching frequency is limited in real life applications and results in discretization chattering problem. To overcome this problem, the boundary solution replaces the discontinuous control with a saturation function that approximates the sign function in a boundary layer of sliding mode manifold. This solution preserves partially the invariance property of sliding mode where states are confined to a small vicinity of the manifold, and convergence to zero cannot be guaranteed.

In recent years, several solutions have been proposed for PMSM speed control, including classical, robust and adaptive control laws, such as vector and sliding mode control but they are generally consider (structured) parametric uncertainties only. The field-oriented vector control technique [71] emulates a separately excited compensated dc motor and its principle consists of controlling the angle and amplitude of the stator field. This conventional control strategy capitalizes on the fact that PMSM torque is proportional to the q-axis current in the synchronous reference frame. Hence, vector control is achieved by decoupling the d-q axes. This transformation allows d-q axes currents to be controlled independently, usually with proportional integral controllers. However, faster torque dynamic response is achieved with Direct Torque Control (DTC) [72] for permanent magnet drives at the expense of current distortion, and torque ripple. These negative effects have been addressed in [73–75], and recently, a compromise between ripple and switching frequency has been studied in [76]. The method consists of either current or flux prediction and selects, consequently, the inverter voltage vector that produces the fastest possible transient. On the other hand, a Model Predictive Control (MPC) has been developed for a permanent-magnet synchronous motor drive in [77] where speed and current controllers are combined together. This technique offers an alternative to the conventional cascade control structure.

Although these controllers work quite well in theory, their performance degrades in the presence of varying operating conditions, structured and unstructured dynamical uncertainties, and external disturbances. These control methods assume a precise mathematical system model. In the real life applications, deriving such model for complex industrial processes might be a difficult task to undertake. In addition, other factors might be unpredictable, such as load dynamics, noise, temperature, and parameter variations. Hence, the system's dynamics cannot be efficiently based on presumably accurate mathematical models.

Moreover, these controllers have been proposed with no analytical solution for stability, which has not been given much attention in the literature. Furthermore, stability based on linearization around an operating point of the states cannot apply for the overall system. A stability analysis has been provided in [78] for a model reference adaptive controller that copes with parameter uncertainties for PMSM currents regulation. However, this analysis is limited to the inner currents loop and does not include the anti-wind up PI used as speed controller. Finally, considering unknown system's dynamics, an interval type-2 fuzzy neural network controller is developed in [9] to approximate the nonlinear PMSM dynamics. Unlike other controllers, the adaptive algorithms are derived using the Lyapunov stability theorem. Several artificial neural network and fuzzy logic models have been proposed for speed and position estimation and control of PMSMs [3, 11], providing an alternative to conventional control techniques.

This chapter presents different advanced intelligent control strategies for Permanent Magnet Synchronous Machine (PMSMs). Adaptive control theory is used to design an adaptive vector control technique with a guaranteed stability. This control scheme uses many controllers and tuning is not trivial. Therefore, this structure is simplified by combining two adaptive controllers. The adaptive control strategies use a speed and disturbance observer. Then, under the uncertain dynamics constraint, an adaptive controller is proposed by assuming less knowledge about the machine's dynamics, which yields simpler control scheme. Therefore, stability can also be guaranteed even with uncertain dynamics. Henceforth, this result is extended to the unknown dynamics case and stability is also achieved for intelligent adaptive controllers, which yields better robustness to unstructured uncertainties. Next, the ANN's approximation property is exploited to design a reduced complexity control scheme able to cope with parametric uncertainties. An ANN-based speed observer is also proposed. However, the estimation accuracy is sensitive to flux variations. This problem is addressed by proposing an ANN-based nonlinear speed observer assuming a no *a priori* dynamics knowledge. Thus, robustness to higher degrees of nonlinearities is achieved. Finally, a single adaptive fuzzy logic controller is proposed to achieve robustness to both structured and unstructured uncertainties, which reduces the system's complexity compared to classical cascaded-based control structures. Moreover, a Lyapunov stability-based adaptation technique is used as an alternative to the conventional heuristic tuning methods.

7.2 Modeling

The PMSM dynamic mathematical model in the d-q axes rotational reference frame can be described by the following equations :

$$v_d = R\, i_d + L_d\, \frac{d}{dt} i_d - L_q\, p\, \omega\, i_q \tag{7.1a}$$

$$v_q = R\, i_q + L_q\, \frac{d}{dt} i_q + L_d\, p\, \omega\, i_d + p\, \lambda\, \omega \tag{7.1b}$$

$$\tau = \frac{3}{2}\, p\, [(L_d - L_q)\, i_d\, i_q + \lambda\, i_q] \tag{7.1c}$$

The mechanical equations of motion can be expressed by :

$$\frac{d}{dt}\omega = \frac{1}{J}\,(\tau - \tau_F - \tau_L) \tag{7.2a}$$

$$\frac{d}{dt}\theta = p\, \omega \tag{7.2b}$$

where,

v_d, v_q	voltage in d-q axes
i_d, i_q	current in d-q axes
L_d, L_q	inductance in d-q axes
R	armature winding resistance
p	number of pole pairs
λ	flux linkage
τ, τ_L, τ_F	motor, load and friction torques
J	rotor and load inertia
θ	rotor electrical position
ω	rotor electrical speed

7.3 Observer-based Adaptive Vector Control

Let $e_\omega = \hat{\omega} - \omega^*$ denote the machine speed error, with $\hat{\omega}$ and ω^* being the speed estimate and its desired time-dependent signal, respectively. Then, let $e_d = i_d - i_d^*$ and $e_q = i_q - i_q^*$ denote the d-q axes current errors, with i_d^* and i_q^* being the desired time-dependent d-q axes current signals. The adaptive control strategy uses quadrature voltage v_q to achieve speed tracking and keeps the exciting current i_d constant to zero. The resultant control scheme is illustrated in Fig. 7.1. When $i_d \to 0$, a maximum torque-current ratio is reached, which is called, the Maximum Torque Per Ampere (MTPA) method for surface-mount PMSMs. As such, an adaptive controller keeps i_d constant to zero with a second adaptive controller tracking the q-axis current error e_q to zero. On the other hand, a third adaptive controller achieves precise speed tracking by machine's inverse dynamics approximation. As depicted in Fig. 7.1, the controllers deliver desired voltages v_d^*, and v_q^*, which are then fed to a Space Vector Pulse Width Modulation (SVPWM) algorithm to produce the proper duty cycles for the inverter.

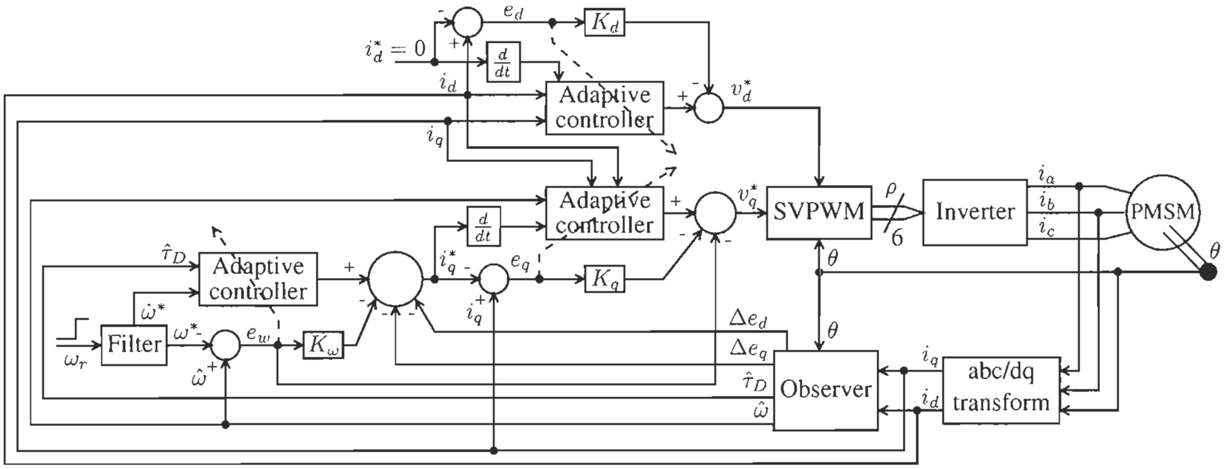


FIGURE 7.1 – Adaptive vector control scheme

Recall the PMSM formulation (7.1) and (7.2),

$$\frac{d}{dt}\omega = \frac{1}{J}\left(\tau - \tau_F - \tau_L\right)$$

$$\tau = \frac{3}{2}\, p\, \left[(L_d - L_q)\, i_d\, i_q + \lambda\, i_q\right]$$

Substitute τ into the mechanical equation and since the permanent magnet machine torque depends mainly on the quadrature current i_q, it is more convenient to set the direct reference current i_d^* to zero, which will minimize the torque vs. current ratio and increase the motor efficiency. Therefore, we obtain for following model :

$$i_q = \mu \dot{\omega} + \sigma \tau_D - \Delta e_d \tag{7.3}$$

with, $\mu = J\sigma$, and $\sigma = \frac{2}{3p\lambda}$. $\tau_D = \tau_F + \tau_L$ is the disturbance (friction+load) torque, which is estimated using the above observer and Δe_d is the uncertainty due to the direct current error e_d described by,

$$\Delta e_d = \frac{(L_d - L_q)i_q}{\lambda}\, e_d$$

It is noteworthy that for surface-mount PMSMs, $\Delta e_d = 0$ since $L_d = L_q$. In this work, the control design is done regardless of the PMSM type and hence, Δe_d is not taken to be equal to zero. The desired dynamics for (7.3) can be represented by the following linear regression,

$$\mu \dot{\omega}^* + \sigma \hat{\tau}_D = \Phi_\omega^T W_\omega \tag{7.4}$$

Recall the PMSM formulation (7.1),

$$v_d = R\, i_d + L_d\, \frac{d}{dt} i_d - L_q\, p\, \omega\, i_q \tag{7.5a}$$

$$v_q = R\, i_q + L_q\, \frac{d}{dt} i_q + L_d\, p\, \omega\, i_d + p\, \lambda\, \omega \tag{7.5b}$$

Therefore, we can also represent the desired dynamics for (7.5) by the following linear regression,

$$R\, i_d + L_d\, \frac{d}{dt} i_d^* - L_q\, p\, \hat{\omega}\, i_q = \Phi_d^T W_d \tag{7.6a}$$

$$R\, i_q + L_q\, \frac{d}{dt} i_q^* + L_d\, p\, \hat{\omega}\, i_d + p\, \lambda\, \hat{\omega} = \Phi_q^T W_q \tag{7.6b}$$

Take the time-derivative of e_ω,

$$\dot{e}_\omega = \dot{\hat{\omega}} - \dot{\omega}^*$$

Add and substract $\dot{\omega}$,

$$\dot{e}_\omega = \dot{\omega} - \dot{\omega}^* + \Delta\dot{\omega}$$

with, $\Delta\omega = \hat{\omega} - \omega$. Multiply both sides by μ and substitute $\mu\dot{\omega}$ from (7.3) and use the linear regression (7.4),

$$\mu\dot{e}_\omega = i_q - \Phi_\omega^T W_\omega + \mu\Delta\dot{\omega} + \sigma e_D + \Delta e_d \tag{7.7}$$

with, $e_D = \hat{\tau}_D - \tau_D$.

Using (7.3) and the observer estimates $\hat{\omega}$ and $\hat{\tau}_D$ we get,

$$\hat{i}_q = \mu\dot{\hat{\omega}} + \sigma\hat{\tau}_D - \Delta e_d \tag{7.8}$$

Subtract (7.8) from (7.3),

$$\Delta e_q = \mu\Delta\dot{\omega} + \sigma e_D$$

with, $\Delta e_q = \hat{i}_q - i_q$. Substitute Δe_q in (7.7),

$$\mu\dot{e}_\omega = i_q - \Phi_\omega^T W_\omega + \Delta e_d + \Delta e_q \tag{7.9}$$

Take the time-derivative of the error signals e_d, and e_q,

$$\dot{e}_d = \frac{d}{dt}i_d - \frac{d}{dt}i_d^*$$
$$\dot{e}_q = \frac{d}{dt}i_q - \frac{d}{dt}i_q^*$$

Multiply the equations by L_d, and L_q, respectively,

$$L_d \dot{e}_d = L_d \frac{d}{dt} i_d - L_d \frac{d}{dt} i_d^*$$
$$L_q \dot{e}_q = L_q \frac{d}{dt} i_q - L_q \frac{d}{dt} i_q^*$$

Substitute $L_d \frac{d}{dt} i_d$, and $L_q \frac{d}{dt} i_q$ from (7.5) and use the linear regression (7.6),

$$L_d \dot{e}_d = v_d - \Phi_d^T W_d \tag{7.10a}$$
$$L_q \dot{e}_q = v_q - \Phi_q^T W_q \tag{7.10b}$$

Since the inverter operates at a smaller period than the motor electrical time constant, a sensible practical assumption is that the inverter reproduces accurately the reference voltages v_d^*, and v_q^*, which reduces the number of sensors and makes $v_d^* = v_d$, and $v_q^* = v_q$.
Therefore, the control law can be defined as,

$$i_q^* = \Phi_\omega^T \hat{W}_\omega - K_\omega e_\omega - \Delta e_d - \Delta e_q \tag{7.11a}$$
$$v_d^* = \Phi_d^T \hat{W}_d - K_d e_d \tag{7.11b}$$
$$v_q^* = \Phi_q^T \hat{W}_q - K_q e_q - e_\omega \tag{7.11c}$$

where K_ω, K_d, and K_q are positive gains.

Theorem 12 *Consider the nonlinear system in (7.1)-(7.2) with the control law (7.11). The control law is asymptotically stable and the convergence of the errors to zero is guaranteed with the following adaptation law :*

$$\dot{\hat{W}}_\omega = -\Gamma_\omega \, \Phi_\omega \, e_\omega \tag{7.12a}$$
$$\dot{\hat{W}}_d = -\Gamma_d \, \Phi_d \, e_d \tag{7.12b}$$
$$\dot{\hat{W}}_q = -\Gamma_q \, \Phi_q \, e_q \tag{7.12c}$$

where $\Gamma_x = diag(\gamma_1, \gamma_2, \ldots, \gamma_n)$ and γ_i is a positive constant.

Proof 6 *Consider the Lyapunov candidate function,*

$$V = \frac{1}{2}\{\mu e_\omega^2 + L_d e_d^2 + L_q e_q^2 + \tilde{W}_\omega^T \Gamma_\omega^{-1} \tilde{W}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \tilde{W}_d + \tilde{W}_q^T \Gamma_q^{-1} \tilde{W}_q\}$$

Take the time-derivative of V:

$$\dot{V} = \mu \dot{e}_\omega e_\omega + L_d \dot{e}_d e_d + L_q \dot{e}_q e_q + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d + \tilde{W}_q^T \Gamma_q^{-1} \dot{\hat{W}}_q$$

The parameters vectors W_ω, W_d, and W_q are assumed to be constant, i.e., $\dot{\tilde{W}} = \dot{\hat{W}}$. Substitute $\mu \dot{e}_\omega$ from (7.9), and $L_d \dot{e}_d$, $L_q \dot{e}_q$ from (7.10),

$$\dot{V} = \{i_q - \Phi_\omega^T W_\omega + \Delta e_d + \Delta e_q\} e_\omega + \{v_d - \Phi_d^T W_d\} e_d + \{v_q - \Phi_q^T W_q\} e_q$$
$$+ \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d + \tilde{W}_q^T \Gamma_q^{-1} \dot{\hat{W}}_q$$

Setting,

$$i_q = i_q^* + e_q$$
$$v_d = v_d^* + \Delta v_d$$
$$v_q = v_q^* + \Delta v_q$$

with $\Delta v_d = v_d - v_d^* = 0$ and $\Delta v_q = v_q - v_q^* = 0$, yields,

$$\dot{V} = \{i_q^* - \Phi_\omega^T W_\omega + \Delta e_d + \Delta e_q + e_q\} e_\omega + \{v_d - \Phi_d^T W_d\} e_d + \{v_q - \Phi_q^T W_q\} e_q$$
$$+ \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d + \tilde{W}_q^T \Gamma_q^{-1} \dot{\hat{W}}_q$$

Setting the control law as in (7.11) yields,

$$\begin{aligned}\dot{V} &= \Phi_\omega^T \tilde{W}_\omega e_\omega + \Phi_d^T \tilde{W}_d e_d + \Phi_q^T \tilde{W}_q e_q + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega \\ &+ \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d + \tilde{W}_q^T \Gamma_q^{-1} \dot{\hat{W}}_q - K_\omega e_\omega^2 - K_d e_d^2 - K_q e_q^2\end{aligned}$$

Setting the adaptation law as in (7.12) implies that,

$$\dot{V} = -K_\omega e_\omega^2 - K_d e_d^2 - K_q e_q^2 < 0$$

Hence, the adaptive control system is asymptotically stable in the sense of Lyapunov.

The above adaptive control strategy requires the speed measurement and the disturbance estimate $\hat{\tau}_D$, which are estimated by a state observer. Recall the PMSM formulation (7.1) and (7.2),

$$\begin{aligned}\frac{d}{dt}\theta &= p\ \omega \\ \frac{d}{dt}\omega &= \frac{1}{J}\ (\tau - \tau_F - \tau_L) \\ \tau &= \frac{3}{2}\ p\ [(L_d - L_q)\ i_d\ i_q + \lambda\ i_q]\end{aligned}$$

Substitute τ,

$$\begin{aligned}\frac{d}{dt}\theta &= p\ \omega \\ \frac{d}{dt}\omega &= \frac{1}{J}\ (\frac{3}{2}\ p\ [(L_d - L_q)\ i_d\ i_q + \lambda\ i_q] - \tau_D)\end{aligned}$$

This can be written in a state space form,

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

where $x \in \mathbb{R}^3 = [\theta,\ \omega,\ \tau_D]^T$ is the state vector, and $u \in \mathbb{R}^3 = [i_d,\ i_q,\ i_d i_q]^T$ is the input vector. $A \in \mathbb{R}^{3\times 3}$, $B \in \mathbb{R}^{3\times 3}$, and $C \in \mathbb{R}^3$ are given by,

$$A = \begin{bmatrix} 0 & a_1 & 0 \\ 0 & 0 & a_2 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b_1 & b_2 \\ 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1\,0\,0 \end{bmatrix}$$

with, $a_1 = p$, $a_2 = -\frac{1}{J}$, $b_1 = \frac{3p\lambda}{2J}$, and $b_2 = \frac{3p}{2J}(L_d - L_q)$. Therefore, the observer is defined as,

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

with, G being the observer gain matrix. Since the pair (C, A) is completely observable, it is possible to choose G such as $(A + GC)$ is a Hurwitz matrix. The observer gain matrix G can be found by solving the algebraic Riccati equation or by using a pole placement technique. Therefore, the stability of the observer is guaranteed.

7.3.1 Setup

To demonstrate the performance of the proposed observer, a set of computer simulation runs is carried out on an Interior Permanent Magnet Synchronous Motor (IPMSM) model, which has been validated experimentally [79] and has been used to design industrial controllers for PMSM-based products, such as, hydroelectric and wind turbines. Table 7.1 summarizes the motor's parameters along with their respective values. The nominal dc-link voltage is 800 V, and the switching and sampling frequencies are both set to 5 kHz. The dc-link voltage drops are ignored, and the inverter is taken as being ideal, i.e., no dead-time compensation is used.

TABLE 7.1 – PMSM's parameters

Parameter	Value
Nominal power (kW)	$P_n = 26$
Nominal torque (N·m)	$\tau_n = 416$
Nominal speed (RPM)	$w_n = 600$
Inductance in d-axis (H)	$L_d = 15.9 \cdot 10^{-3}$
Inductance in q-axis (H)	$L_q = 24.88 \cdot 10^{-3}$
Armature winding resistance (Ω)	$R = 361.45 \cdot 10^{-3}$
Flux linkage (Wb)	$\lambda = 1.6504$
Coulomb friction coefficient (N·m)	$F_c = 1$
Viscous friction coefficient (N·m·s/rad)	$F_v = 2$
Static friction coefficient (N·m)	$F_s = 7 \cdot 10^{-1}$
Static friction decreasing rate (rad/s)	$\eta_s = 5 \cdot 10^{-2}$
Rotor and load inertia (kg·m^2)	$J = 5$
Number of pole pairs	$p = 5$

7.3.2 Results

Four simulation runs are carried out to study the controller's performance. For each simulation, the controller's initial parameter estimate vectors $\hat{W}_\omega$, $\hat{W}_d$, and $\hat{W}_q$ are set to zero and the system's response is studied taking into account the machine's tracking and estimation speed errors, the d-q axes current errors (e_d, e_q) and voltages (v_d, v_q), the disturbance estimate $\hat{\tau}_D$, and the adaptive parameter estimate vectors $\hat{W}_\omega$, $\hat{W}_d$, and $\hat{W}_q$. The desired rotor speed is shown in Fig. 7.2.

In this simulation, the nominal values (Table 7.1) are used to simulate the machine's dynamics. As shown in Fig. 7.3, the tracking speed error decays gradually before stabilizing within negligible amplitude while accurate speed and disturbance estimation is obtained with

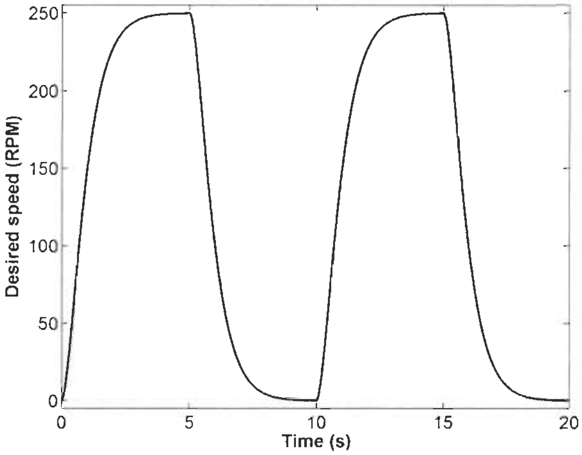


FIGURE 7.2 – Adaptive control speed reference signal

the proposed observer. On the other hand, both controllers provide smooth d-q axes currents and voltages. Moreover, the adaptive control scheme achieves high tracking precision and the observer delivers good performance during fast changes in electromagnetic torque. It is noteworthy the ability of the observer to estimate accurately the friction torque as shown in Fig. 7.3(e).

In the second simulation, a 100 (N·m) load torque is applied at time t=10s to evaluate the adaptive controller's performance to a disturbance torque. As shown in Fig. 7.4, when the drive is subjected to a disturbance torque, it compensates for this unexpected change by adjusting its parameters. Consequently, the control structure was successful in coping with the load torque variation (Fig. 7.4(e)). Moreover, the speed tracking error remains small, which yielded smooth currents and control signal. However, a speed estimation error increase is noticed and higher torque is required for the heavier load.

This simulation is meant to show the ability of the proposed controller in compensating for friction nonlinearities of different magnitudes. For that, the nonlinear Coulomb friction term F_c is magnified 10 times and the machine operates in motoring and regenerating modes to enable zero velocity crossing (i.e., Coulomb friction effect) as shown in Fig. 5.3. The results are shown in Fig. 7.5. As it can be seen, the change in the speed reference signal has no effect on the overall control performance. The speed estimation error starts increasing during

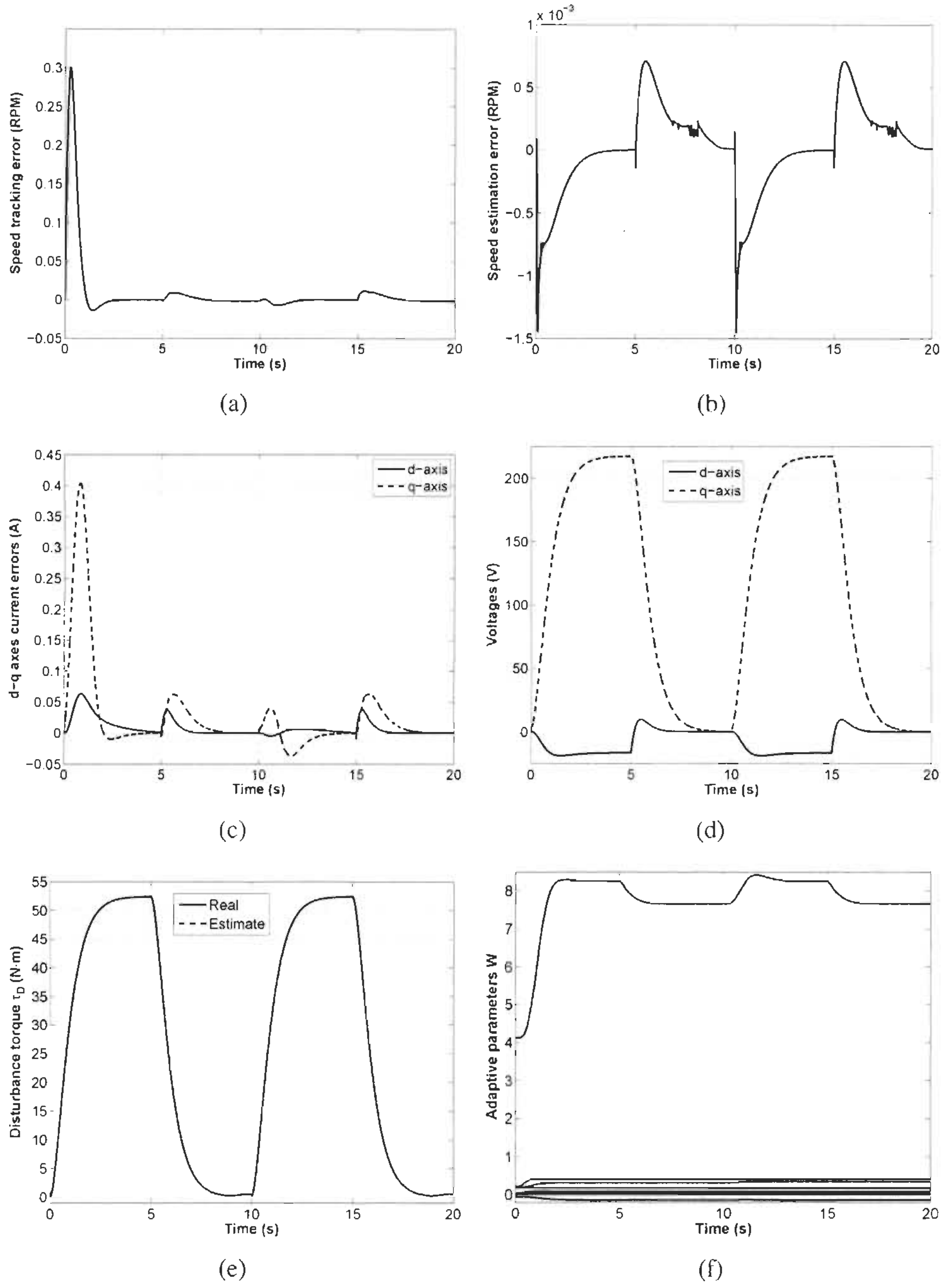
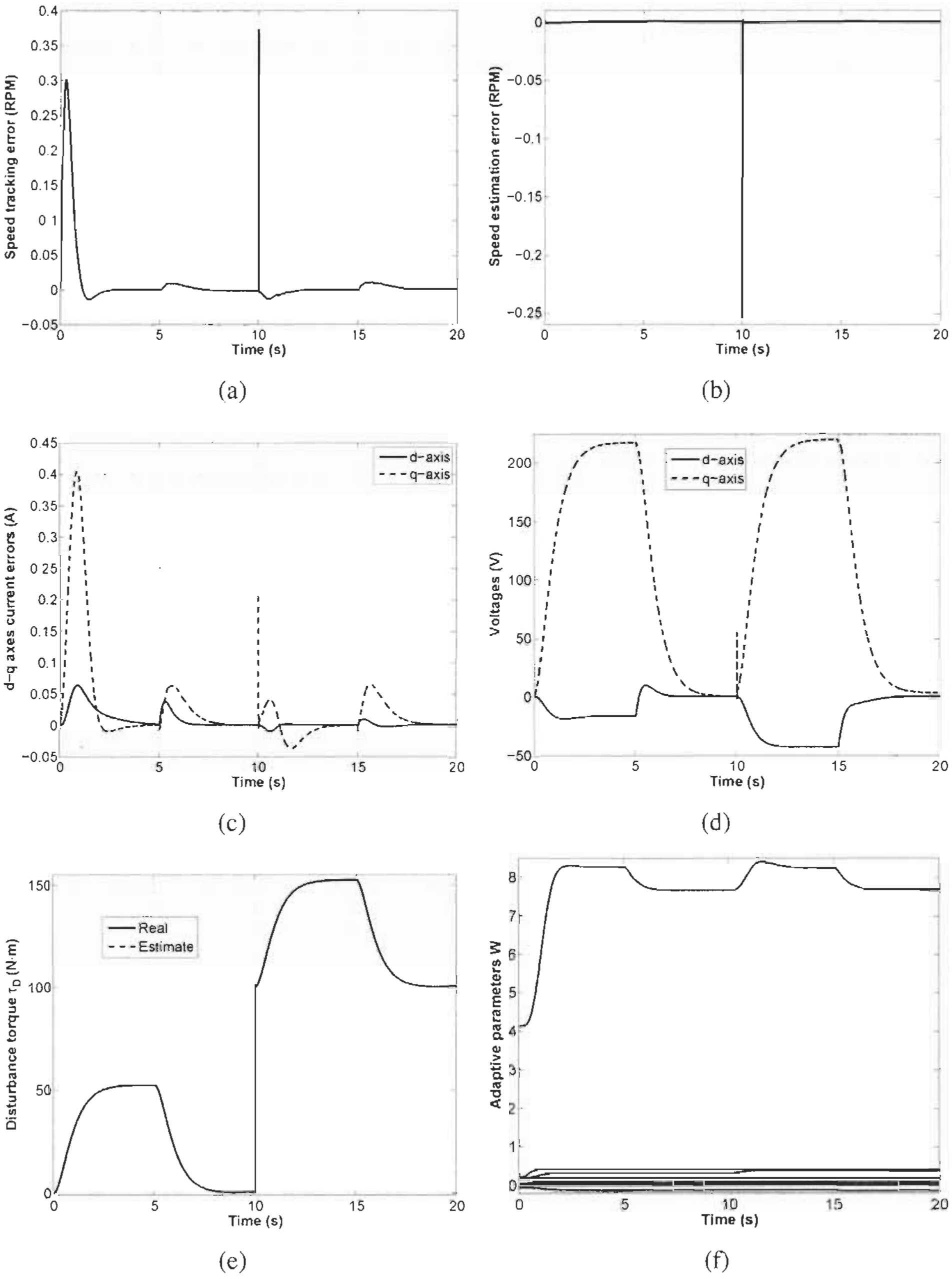


FIGURE 7.3 – Adaptive vector control response with nominal values : (a) speed tracking error ; (b) speed estimation error ; (c) d-q axes current errors ; (d) d-q axes voltages ; (e) disturbance ; and (f) Adaptive parameters $\hat{W}$.



(a)

(b)

(c)

(d)

(e)

(f)

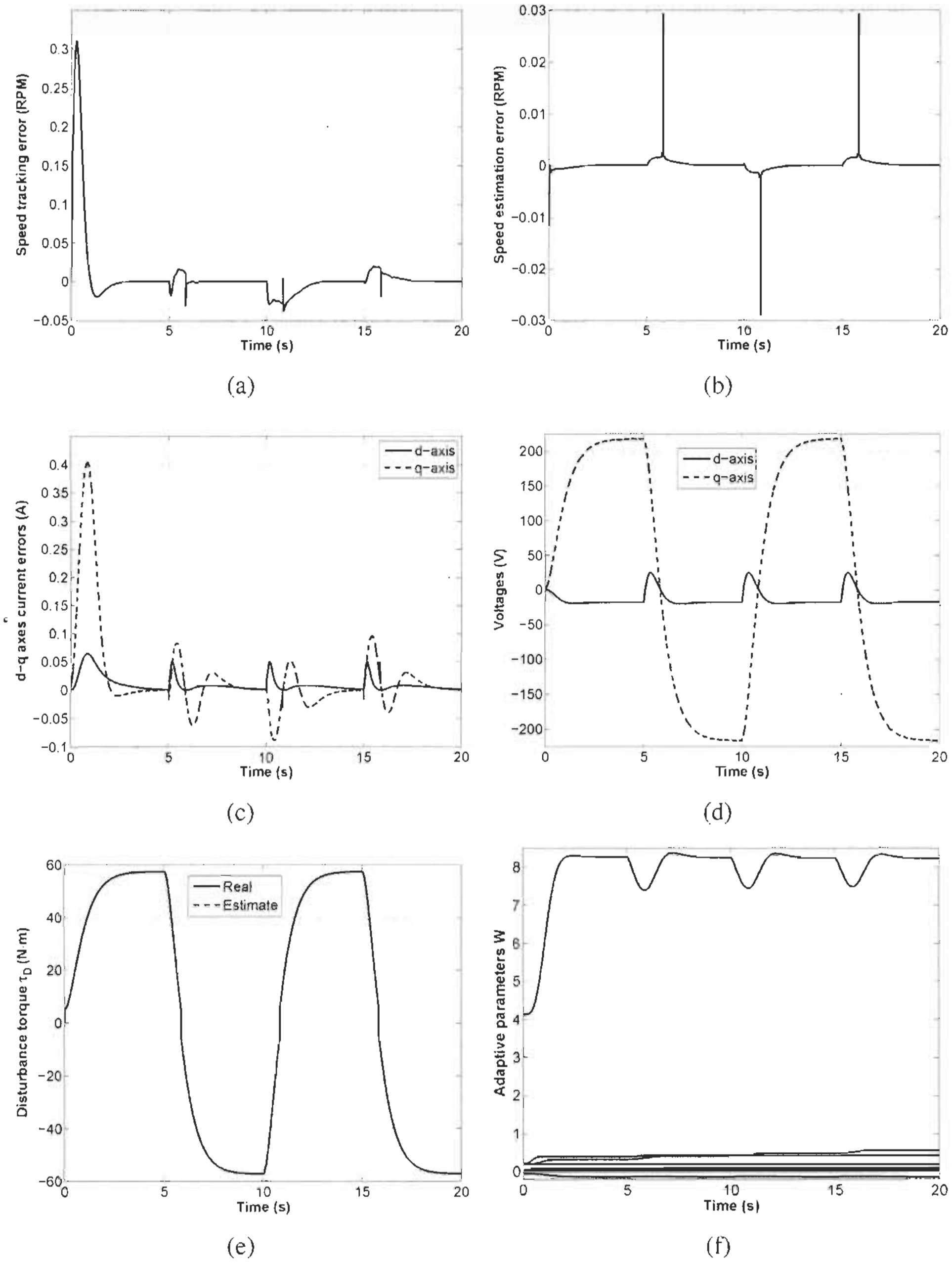
FIGURE 7.4 – Adaptive vector control response with load variations : (a) speed tracking error ; (b) speed estimation error ; (c) d-q axes current errors ; (d) d-q axes voltages ; (e) disturbance ; and (f) Adaptive parameters $\hat{W}$.

fast accelerations but stays in an acceptable range. Conventional control techniques tend to overcompensate for these effects and lead to severe tracking errors, limit cycles, chattering, excessive noise, and even instability [17]. As shown in Fig. 7.5, the nonlinearities around zero speed cause a tracking error. However, the controller is able to compensate for the nonlinear friction and no oscillation is observed (Fig. 7.5(e)), which yielded accurate speed tracking in both motoring and regenerating modes.

Since the machine's parameters are time-varying, a simulation is carried out to study the proposed observer's ability to sustain various parametric uncertainties. The controller's performance is similar to the nominal case, which is expected since the controller is adaptive and hence is independent of the machine's mechanical and electrical parameters. However, the observer is based on parameters J, L_d, L_q, λ, and p. The parameter p is the number of pairs of poles and is constant. Therefore, the machine's parameters, J, L_d, L_q, and λ are increased one at a time by 50% their nominal values. The observer's performance is depicted in Fig. 7.6. It is noteworthy that from the observer dynamics, $\frac{3}{2J}p[(L_d - L_q)i_d + \lambda]$ is the most significant term. Thus, the variation of L_d, and L_q does not have much effect on the observer's performance as it is shown in Fig. 7.6(b), and Fig. 7.6(c) since setting i_d^* to zero makes the observer less sensitive to inductance variations. On the other hand, the inertia J and the flux λ variations cause a slight increase in the estimation error as it is observed in Fig. 7.6(a) and Fig. 7.6(d). It remains however in an acceptable range unlike other estimation techniques.

7.4 Observer-based Adaptive Control

The above adaptive vector control strategy uses three adaptive controllers to drive the errors e_ω, e_d, and e_q to zero. Since the goal is to achieve trajectory tracking, i.e., $e_\omega \to 0$, the speed controller can be combined with the q-axis current controller, which yields less complex control scheme. As such, an adaptive controller keeps i_d constant to zero while a second adaptive controller achieves precise speed tracking by machine's inverse dynamics approximation. The resultant control scheme is illustrated in Fig. 7.7. Let $e_q = \hat{i}_q - i_q$ denote the observer error,



(a)

(b)

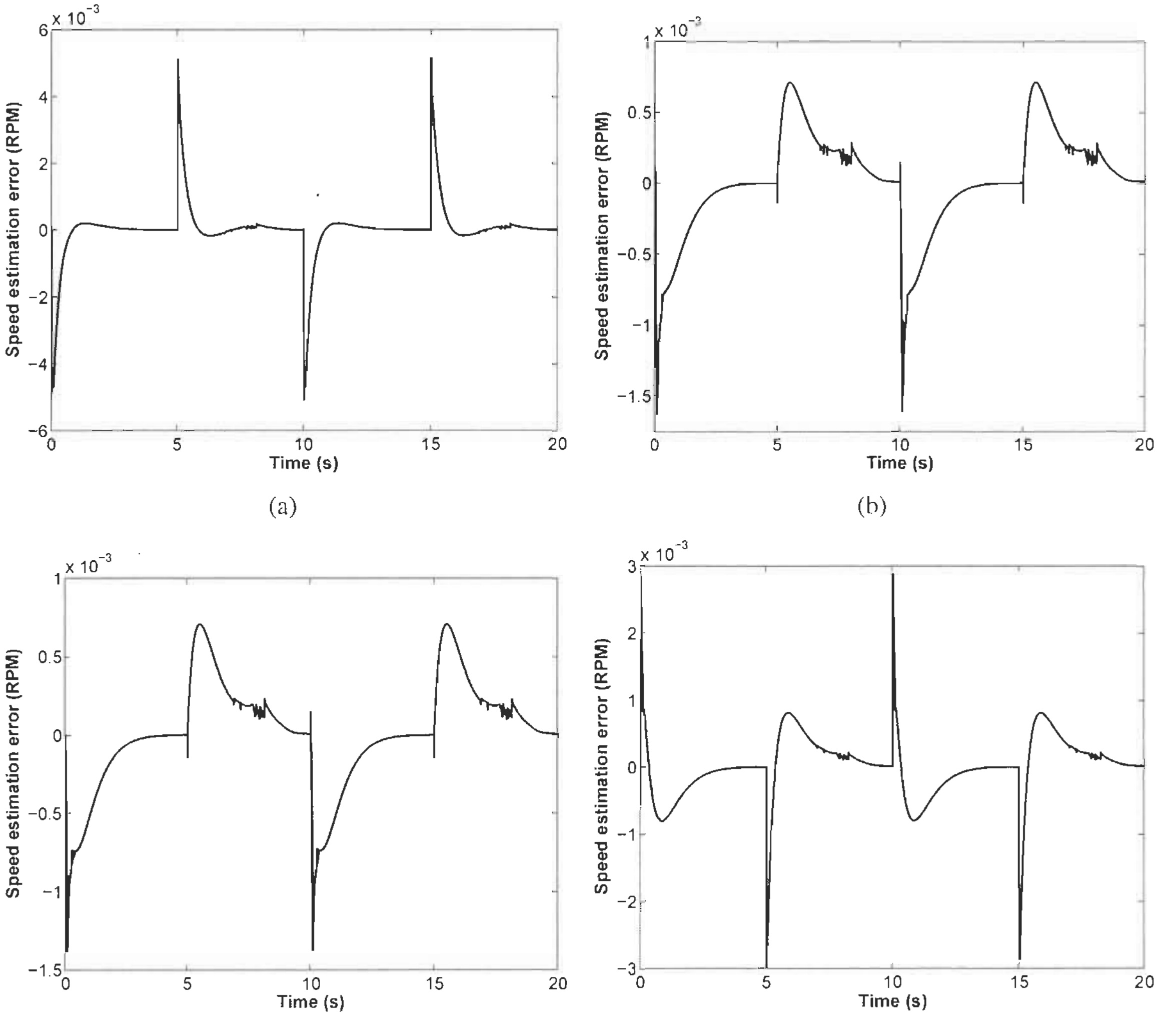
(c)

(d)

(e)

(f)

FIGURE 7.5 – Adaptive vector control response with magnified nonlinear friction : (a) speed tracking error ; (b) speed estimation error ; (c) d-q axes current errors ; (d) d-q axes voltages ; (e) disturbance ; and (f) Adaptive parameters $\hat{W}$.



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

FIGURE 7.6 – Adaptive vector control with variation of : (a) inertia J ; (b) inductance L_d ; (c) inductance L_q ; and (d) flux λ.

with $\hat{i}_q$ being the q-axis current estimate signal. As depicted in Fig. 7.7, the observer presented earlier is used for accurate speed and disturbance estimation.

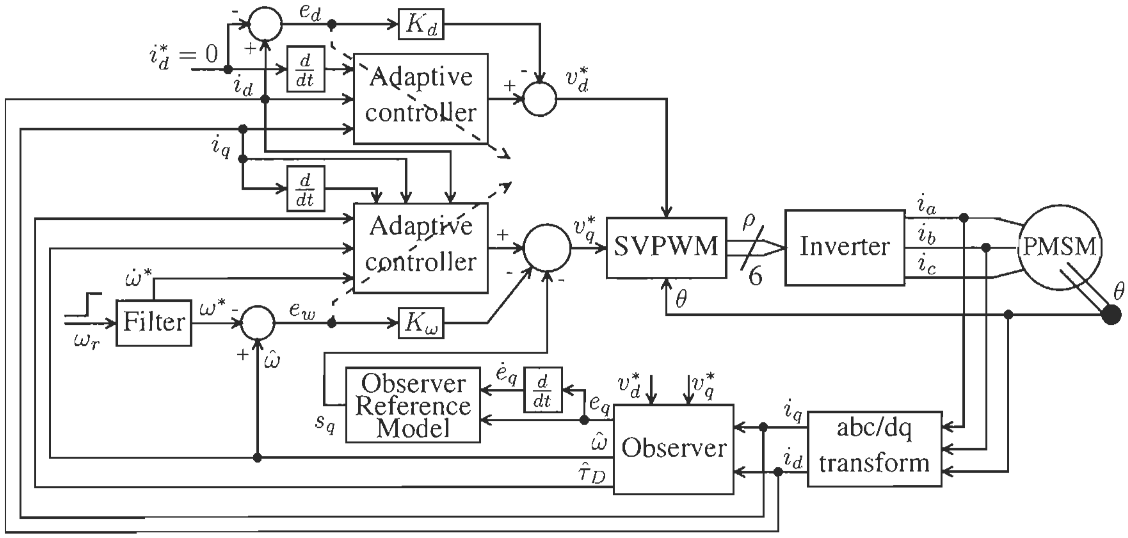


FIGURE 7.7 – Adaptive control scheme

Recall the PMSM formulation (7.1) and (7.2),

$$\frac{d}{dt}\omega = \frac{1}{J}\,(\tau - \tau_F - \tau_L)$$

$$\tau = \frac{3}{2}\,p\,[(L_d - L_q)\,i_d\,i_q + \lambda\,i_q]$$

Substitute for τ :

$$\dot{\omega} = \frac{1}{J}\,(\frac{3}{2}\,p\,[(L_d - L_q)\,i_d\,i_q + \lambda\,i_q] - \tau_F - \tau_L)$$

Multiply both sides by $\mu = \frac{2RJ}{3p\lambda}$,

$$\mu\,\dot{\omega} = \frac{R}{\lambda}\,(L_d - L_q)\,i_d\,i_q + R\,i_q - \frac{\mu}{J}\,(\tau_F + \tau_L) \tag{7.13}$$

Recall the PMSM formulation (7.1b) :

$$v_q = R\,i_q + L_q\,\frac{d}{dt}i_q + L_d\,p\,\omega\,i_d + p\,\lambda\,\omega$$

Substitute $R\, i_q$ in (7.13) :

$$\mu\,\dot{\omega} = \frac{R}{\lambda}\,(L_d - L_q)\; i_d\, i_q + v_q - L_q\,\frac{d}{dt} i_q - L_d\; p\;\omega\; i_d - p\;\lambda\;\omega - \frac{\mu}{J}\,(\tau_F + \tau_L)$$

Therefore, we obtain for following model :

$$v_q = \mu\,\dot{\omega} - \frac{R}{\lambda}\,(L_d - L_q)\; i_d\, i_q + L_q\,\frac{d}{dt} i_q + L_d\; p\;\omega\; i_d + p\;\lambda\;\omega + \frac{\mu}{J}\;\tau_F + \frac{\mu}{J}\;\tau_L \tag{7.14}$$

Setting $\sigma = p(L_d i_d + \lambda)$ and $\tau_D = \tau_F + \tau_L$ as disturbance (friction+load) torque, which is estimated using an observer, yields,

$$v_q = \mu\,\dot{\omega} - \frac{R}{\lambda}\,(L_d - L_q)\; i_d\, i_q + L_q\,\frac{d}{dt} i_q + \sigma\;\omega + \frac{\mu}{J}\;\tau_D \tag{7.15}$$

Recall the PMSM formulation (7.1),

$$v_d = R\; i_d + L_d\,\frac{d}{dt} i_d - L_q\; p\;\omega\; i_q \tag{7.16}$$

The desired dynamics for (7.15), and (7.16) can be represented by the following linear regression,

$$\mu\,\dot{\omega}^* - \frac{R}{\lambda}\,(L_d - L_q)\; i_d\, i_q + L_q\,\frac{d}{dt} i_q + \sigma\;\hat{\omega} + \frac{\mu}{J}\;\hat{\tau}_D = \Phi_\omega^T W_\omega \tag{7.17a}$$

$$R\; i_d + L_d\,\frac{d}{dt} i_d^* - L_q\; p\;\omega\; i_q = \Phi_d^T W_d \tag{7.17b}$$

Take the time-derivative of e_ω,

$$\dot{e}_\omega = \dot{\hat{\omega}} - \dot{\omega}^*$$

Add and substract $\dot{\omega}$,

$$\dot{e}_\omega = \dot{\omega} - \dot{\omega}^* + \Delta\dot{\omega}$$

with, $\Delta\omega = \hat{\omega} - \omega$. Multiply both sides by μ and substitute $\mu\dot{\omega}$ from (7.15) and use the linear regression (7.17a),

$$\mu \dot{e}_{\omega} = v_q - \Phi_{\omega}^T W_{\omega} + \sigma \Delta\omega + \mu \Delta\dot{\omega} + \frac{\mu}{J} e_D \tag{7.18}$$

with, $e_D = \hat{\tau}_D - \tau_D$.

Using (7.15) and the observer estimates $\hat{\omega}$ and $\hat{\tau}_D$ we get,

$$L_q \frac{d}{dt}\hat{i}_q = v_q - \mu\ \dot{\hat{\omega}} + \frac{R}{\lambda}\ (L_d - L_q)\ i_d\ \hat{i}_q - \sigma\ \hat{\omega} - \frac{\mu}{J}\ \hat{\tau}_D \tag{7.19}$$

Subtract (7.19) from (7.15),

$$L_q \dot{e}_q = \delta e_q - \sigma \Delta\omega - \mu \Delta\dot{\omega} - \frac{\mu}{J} e_D$$

with, $\delta = \frac{R}{\lambda}(L_d - L_q) i_d$.

$$\delta e_q - L_q \dot{e}_q = \sigma \Delta\omega + \mu \Delta\dot{\omega} + \frac{\mu}{J} e_D$$

Substitute $\delta e_q - L_q \dot{e}_q$ in (7.18),

$$\mu \dot{e}_{\omega} = v_q - \Phi_{\omega}^T W_{\omega} + \delta e_q - L_q \dot{e}_q \tag{7.20}$$

Define the observer reference model signal as,

$$s_q = K_{q1} e_q - K_{q2} \dot{e}_q \tag{7.21}$$

where K_{q1}, and K_{q2} are positive gains.

Take the time-derivative of the error signal e_d,

$$\dot{e}_d = \frac{d}{dt} i_d - \frac{d}{dt} i_d^*$$

Multiply both sides by L_d,

$$L_d \dot{e}_d = L_d \frac{d}{dt} i_d - L_d \frac{d}{dt} i_d^*$$

Substitute $L_d \frac{d}{dt} i_d$ from (7.16) and use the linear regression (7.17b),

$$L_d \dot{e}_d = v_d - \Phi_d^T W_d \tag{7.22}$$

Since the inverter operates at a smaller period than the motor electrical time constant, a sensible practical assumption is that the inverter reproduces accurately the reference voltages v_d^*, and v_q^*, which reduces the number of sensors and makes $v_d^* = v_d$, and $v_q^* = v_q$.
Therefore, the control law can be defined as,

$$v_q^* = \Phi_\omega^T \hat{W}_\omega - K_\omega e_\omega - s_q \tag{7.23a}$$

$$v_d^* = \Phi_d^T \hat{W}_d - K_d e_d \tag{7.23b}$$

where K_ω, and K_d are positive gains.

Theorem 13 *Consider the nonlinear system in (7.1)-(7.2) with the control law (7.23). The control law is asymptotically stable and the convergence of the errors to zero is guaranteed with the following adaptation law :*

$$\dot{\hat{W}}_\omega = -\Gamma_\omega \, \Phi_\omega \, e_\omega \tag{7.24a}$$

$$\dot{\hat{W}}_d = -\Gamma_d \, \Phi_d \, e_d \tag{7.24b}$$

where $\Gamma_x = diag(\gamma_1, \gamma_2, \ldots, \gamma_n)$ and γ_i is a positive constant.

Proof 7 *Consider the Lyapunov candidate function,*

$$V = \frac{1}{2}\{\mu e_\omega^2 + L_d e_d^2 + \tilde{W}_\omega^T \Gamma_\omega^{-1} \tilde{W}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \tilde{W}_d\}$$

Take the time-derivative of V :

$$\dot{V} = \mu \dot{e}_\omega e_\omega + L_d \dot{e}_d e_d + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d$$

The parameters vectors W_ω, and W_d are assumed to be constant, i.e., $\dot{\tilde{W}} = \dot{\hat{W}}$. Substitute $\mu \dot{e}_\omega$ from (7.20), and $L_d \dot{e}_d$ from (7.22),

$$\dot{V} = \{v_q - \Phi_\omega^T W_\omega + \delta e_q - L_q \dot{e}_q\} e_\omega + \{v_d - \Phi_d^T W_d\} e_d + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d$$

Setting the control law as in (7.23) yields,

$$\dot{V} = \Phi_\omega^T \tilde{W}_\omega e_\omega + \Phi_d^T \tilde{W}_d e_d + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d + \{\delta e_q - L_q \dot{e}_q - s_q\} e_\omega - K_\omega e_\omega^2 - K_d e_d^2$$

Substitute s_q from (7.21). Set $K_{q1} = \delta$, and $K_{q2} = L_q$,

$$\dot{V} = \Phi_\omega^T \tilde{W}_\omega e_\omega + \Phi_d^T \tilde{W}_d e_d + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d - K_\omega e_\omega^2 - K_d e_d^2$$

Setting the adaptation law as in (7.24) implies that,

$$\dot{V} = -K_\omega e_\omega^2 - K_d e_d^2 < 0$$

Hence, the adaptive control system is asymptotically stable in the sense of Lyapunov.

7.4.1 Results

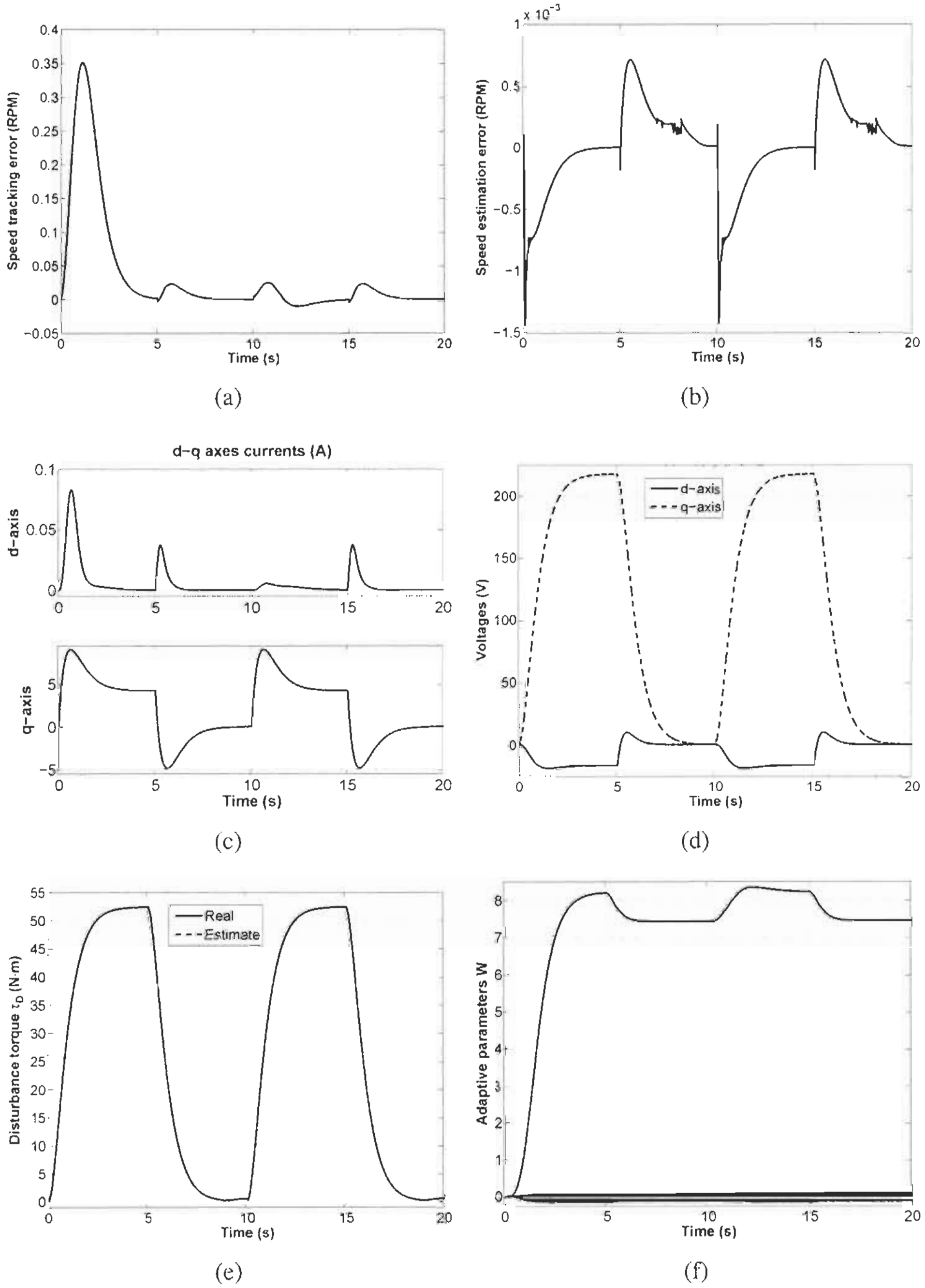
Four simulation runs are carried out to study the controller's performance. For each simulation, the controller's initial parameter estimate vectors $\hat{W}_\omega$ and $\hat{W}_d$ are set to zero and the system's response is studied taking into account the machine's tracking and estimation speed errors, the d-q axes currents and voltages, the disturbance estimate, and the adaptive parameter estimate vectors $\hat{W}_\omega$ and $\hat{W}_d$. The desired rotor speed is shown in Fig. 7.2.

In this simulation, the nominal values (Table 7.1) are used to simulate the machine's dynamics. As shown in Fig. 7.8, the tracking speed error decays gradually before stabilizing within negligible amplitude while accurate speed and disturbance estimation is obtained with the proposed observer. On the other hand, both controllers provide smooth d-q axes currents and voltages. Moreover, the adaptive control scheme achieves high tracking precision and the observer delivers good performance during fast changes in electromagnetic torque. It is noteworthy the ability of the observer to estimate accurately the friction torque as shown in Fig. 7.8(e).

In the second simulation, a 100 (N·m) load torque is applied at time t=10s to evaluate the adaptive controller's performance to a disturbance torque. As shown in Fig. 7.9, when the drive is subjected to a disturbance torque, it compensates for this unexpected change by adjusting its parameters. Consequently, the control structure was successful in coping with the load torque variation (Fig. 7.9(e)). Moreover, the speed tracking error remains small, which yielded smooth currents and control signal. However, a speed estimation error increase is noticed and higher torque is required for the heavier load.

The next simulation is meant to show the ability of the proposed controller in compensating for friction nonlinearities of different magnitudes. For that, the nonlinear Coulomb friction term F_c is magnified 10 times and the machine operates in motoring and regenerating modes to enable zero velocity crossing (i.e., Coulomb friction effect) as shown in Fig. 5.3. The results are shown in Fig. 7.10. As it can be seen, the change in the speed reference signal has no effect on the overall control performance. The speed estimation error starts increasing during fast accelerations but stays in an acceptable range. Conventional control techniques tend to overcompensate for these effects and lead to severe tracking errors, limit cycles, chattering, excessive noise, and even instability [17]. As shown in Fig. 7.10, the nonlinearities around zero speed cause a tracking error. However, the controller is able to compensate for the nonlinear friction and no oscillation is observed (Fig. 7.10(e)), which yielded accurate speed tracking in both motoring and regenerating modes.

Since the observer is taken from the adaptive vector control strategy, the observer's performance is expected to be similar to the one depicted in Fig. 7.6. Therefore, the machine's parameters, J, L_d, L_q, and λ are increased one at a time by 50% their nominal values. The observer's performance is depicted in Fig. 7.11. It is noteworthy that the observer's performance remains less sensitive to parameter variations.



(a)

(b)

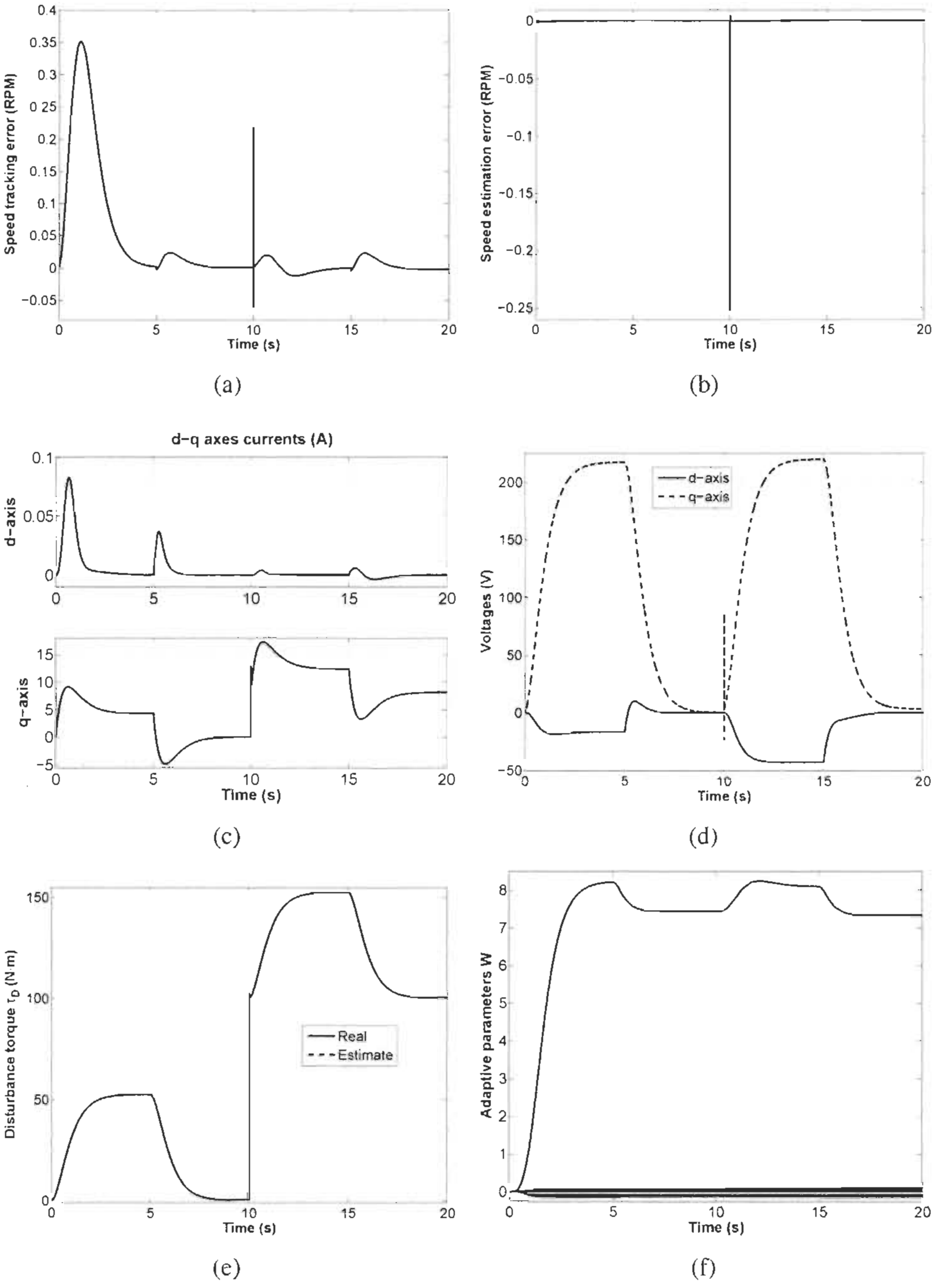
(c)

(d)

(e)

(f)

FIGURE 7.8 – Adaptive control response with nominal values : (a) speed tracking error ; (b) speed estimation error ; (c) d-q axes currents ; (d) d-q axes voltages ; (e) disturbance ; and (f) Adaptive parameters $\hat{W}$.



(a)

(b)

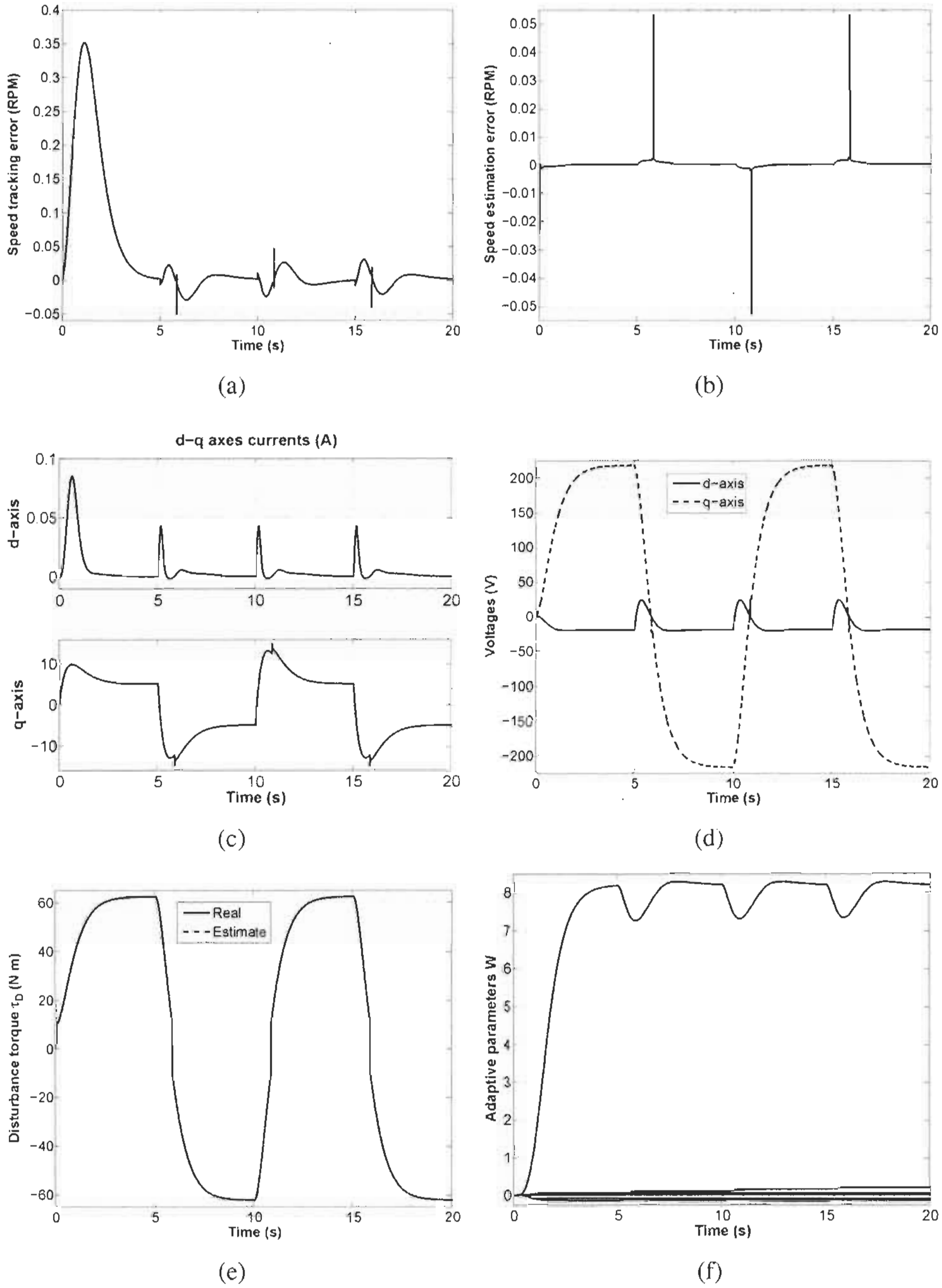
(c)

(d)

(e)

(f)

FIGURE 7.9 – Adaptive control response with load variations : (a) speed tracking error ; (b) speed estimation error ; (c) d-q axes currents ; (d) d-q axes voltages ; (e) disturbance ; and (f) Adaptive parameters $\hat{W}$.



(a)

(b)

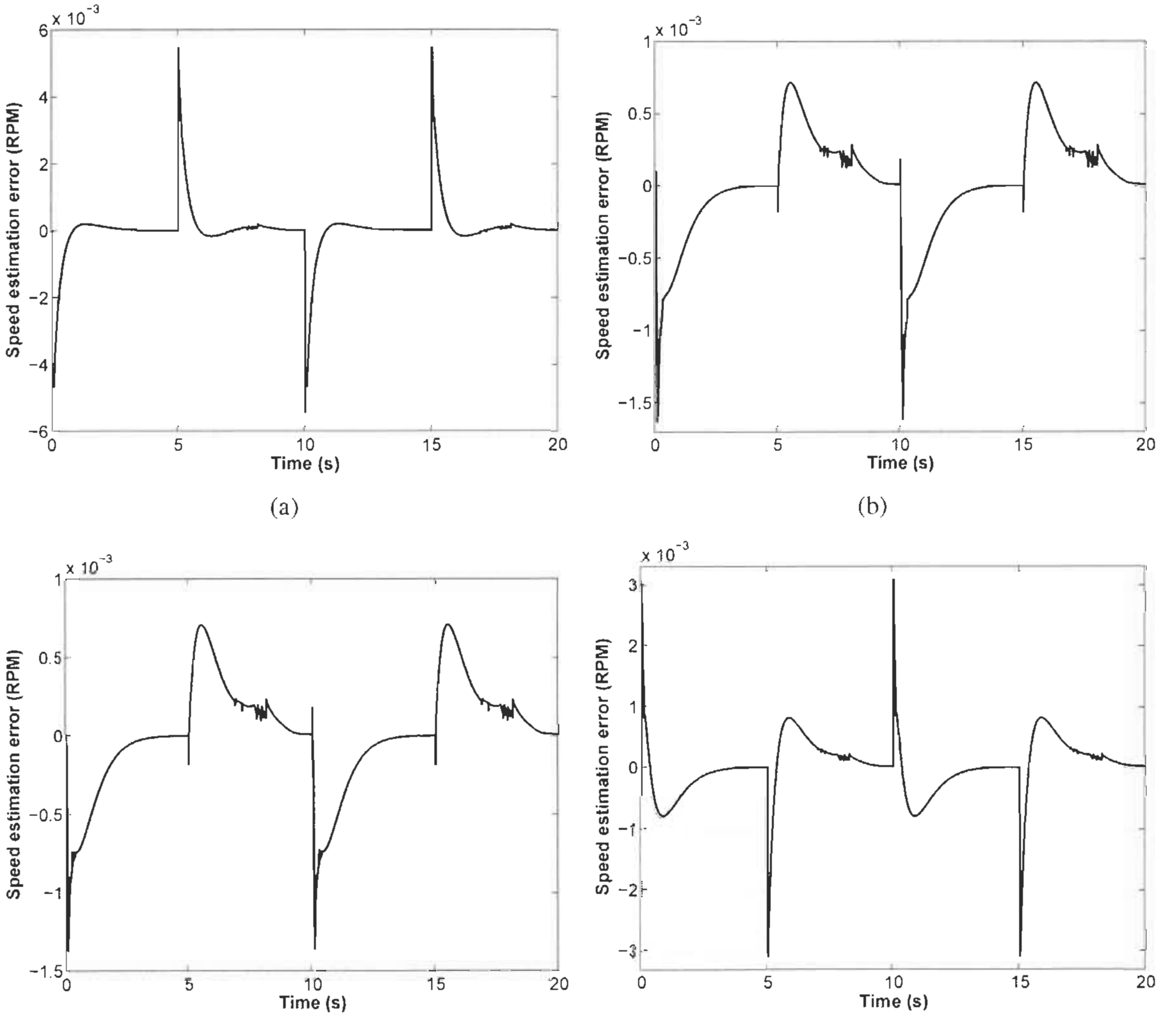
(c)

(d)

(e)

(f)

FIGURE 7.10 – Adaptive control response with magnified nonlinear friction : (a) speed tracking error ; (b) speed estimation error ; (c) d-q axes currents ; (d) d-q axes voltages ; (e) disturbance ; and (f) Adaptive parameters $\hat{W}$.



(a)

(b)

(c)

(d)

FIGURE 7.11 – Adaptive control with variation of : (a) inertia J ; (b) inductance L_d ; (c) inductance L_q ; and (d) flux λ.

7.5 Adaptive Control with Uncertain Dynamics

The adaptive controllers presented thus far use the machine's dynamics to achieve speed tracking. Under the uncertain dynamics constraint, the proposed adaptive controller assumes less knowledge about the machine's dynamics, which yields simpler control scheme. As such, a PI controller keeps i_d constant to zero while the adaptive controller achieves precise speed tracking by machine's inverse dynamics approximation as shown in Fig. 7.12. On the other hand, a state observer is used as feedback for disturbance estimation.

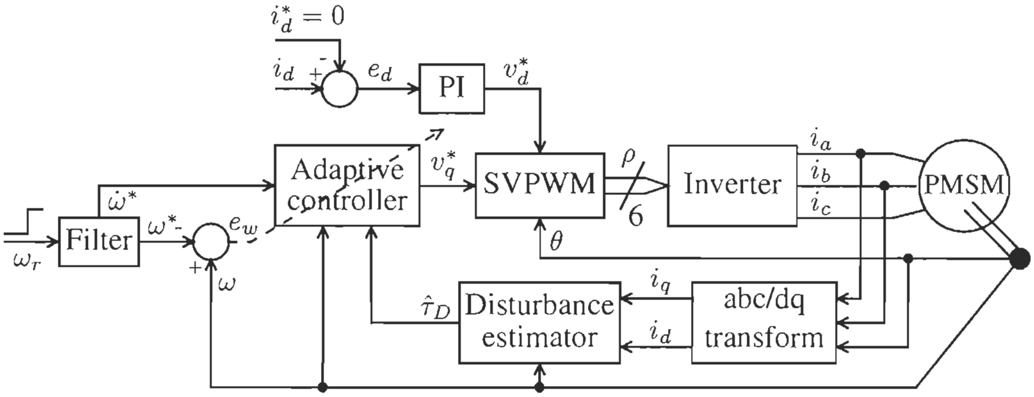


FIGURE 7.12 – Disturbance estimation scheme

Recall the formulation (7.15),

$$v_q = \mu\ \dot{\omega} - \frac{R}{\lambda}\ (L_d - L_q)\ i_d\ i_q + L_q\ \frac{d}{dt} i_q + \sigma\ \omega + \frac{\mu}{J}\ \tau_D \tag{7.25}$$

where $\sigma = p(L_d i_d + \lambda)$ and $\tau_D = \tau_F + \tau_L$ is the disturbance (friction+load) torque, which is estimated using a disturbance estimator. It is noteworthy from (7.25) that $\sigma\omega + \frac{\mu}{J}\tau_D$ is the most significant term in steady state since all derivatives vanish and setting i_d^* to zero makes the contribution of $\frac{R}{\lambda}(L_d - L_q) i_d i_q$ negligible in the control law. During transients, $L_q \frac{d}{dt} i_q$ is negligible compared to $\mu\dot{\omega}$. Therefore, (7.25) can be approximated by,

$$v_q \approx \mu\ \dot{\omega} + \sigma\ \omega + \frac{\mu}{J}\ \tau_D \tag{7.26}$$

It can also be represented in a linear form $\Phi_{\omega}^{T} W_{\omega}$. Therefore, the desired dynamics of (7.26) satisfy to the following linear regression,

$$\mu\ \dot{\omega}^{*} + \sigma\ \omega + \frac{\mu}{J}\ \tau_{D} = \hat{\Phi}_{\omega}^{T} W_{\omega} \tag{7.27}$$

where $\hat{\Phi}_{\omega}$ is the approximation of Φ_{ω}. This approximation introduces an uncertainty that can be represented as,

$$\varepsilon = \hat{\Phi}_{\omega}^{T} \hat{W}_{\omega} - \Phi_{\omega}^{T} W_{\omega}$$

where $\hat{W}_{\omega}$ is the parameter estimate vector. Take the time-derivative of e_{ω},

$$\dot{e}_{\omega} = \dot{\omega} - \dot{\omega}^{*}$$

Multiply both sides by μ and substitute $\mu\dot{\omega}$ from (7.26) and use the linear regression (7.27),

$$\mu \dot{e}_{\omega} = v_{q} - \Phi_{\omega}^{T} W_{\omega} \tag{7.28}$$

Since the inverter operates at a smaller period than the motor electrical time constant, a sensible practical assumption is that the inverter reproduces accurately the reference voltages v_{d}^{*}, and v_{q}^{*}, which reduces the number of sensors and makes $v_{q}^{*} = v_{q}$. Therefore, the control law can be defined as,

$$v_{q}^{*} = \hat{\Phi}_{\omega}^{T} \hat{W}_{\omega} - K_{\omega} e_{\omega} \tag{7.29}$$

where K_{ω} is a positive robustness gain introduced to cope with the uncertainty ε.

Theorem 14 *Consider the nonlinear system in (7.1)-(7.2) with the control law (7.29). The closed-loop system's stability is achieved with the following adaptation law :*

$$\dot{\hat{W}}_{\omega} = -\Gamma_{\omega} \hat{\Phi}_{\omega} e_{\omega} \tag{7.30}$$

where $\Gamma_{\omega} = [\gamma_{1}, \gamma_{2}, \ldots, \gamma_{n}]$ and γ_{i} is a positive constant.

Proof 8 *Consider the Lyapunov candidate function,*

$$V = \frac{1}{2}\{\mu e_\omega^2 + \tilde{W}_\omega^T \Gamma_\omega^{-1} \tilde{W}_\omega\}$$

Take the time-derivative of V :

$$\dot{V} = \mu \dot{e}_\omega e_\omega + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega$$

The parameters vector W_ω is assumed to be constant, i.e., ($\dot{\tilde{W}}_\omega = \dot{\hat{W}}_\omega$). Substitute $\mu \dot{e}_\omega$ from (7.28),

$$\dot{V} = \{v_q - \Phi_\omega^T W_\omega\} e_\omega + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega$$

Setting the control law as in (7.29) yields,

$$\dot{V} = \varepsilon_\omega e_\omega + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega - K_\omega e_\omega^2$$

Substitute ε_ω from (6.26),

$$\dot{V} = \hat{\Phi}_\omega^T \tilde{W}_\omega e_\omega + \tilde{\Phi}_\omega^T W_\omega e_\omega + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega - K_\omega e_\omega^2$$

Setting the adaptation law as in (7.30) implies that,

$$\dot{V} = \tilde{\Phi}_\omega^T W_\omega e_\omega - K_\omega e_\omega^2$$

Recall Young's inequality [80],

$$2ab \leq \frac{1}{\alpha} a^2 + \alpha b^2 \qquad \forall a, b \in \mathbb{R} \text{ and } \forall \alpha > 0$$

Therefore,

$$\dot{V} \leq \frac{1}{2\alpha}(\tilde{\Phi}_{\omega}^{T} W_{\omega})^{2} + \frac{\alpha}{2} e_{\omega}^{2} - K_{\omega} e_{\omega}^{2}$$

Setting,

$$K_{\omega} = \frac{\alpha}{2} + \beta$$

where β is positive gain yields,

$$\dot{V} \leq \frac{1}{2\alpha}\Psi - \beta e_{\omega}^{2}$$

with,

$$\Psi = (\tilde{\Phi}_{\omega}^{T} W_{\omega})^{2}$$

So, it is possible to choose $\alpha > 0$ and $\beta > 0$ so that $\dot{V} \leq 0$. Therefore, the system is stable in the sense of Lyapunov and converges to a small neighborhood of the origin, which is a region defined by the approximation error $\tilde{\Phi}$. This region gets smaller as $\tilde{\Phi} \rightarrow 0$. Theoretically, it can be made arbitrarily small by increasing the gains α and β.

The above adaptive control strategy requires the knowledge of the disturbance τ_D, which is estimated by a state observer. Recall the system's dynamics (7.1) and (7.2),

$$\begin{aligned} \frac{d}{dt}\omega &= \frac{1}{J}\,(\tau - \tau_F - \tau_L) \\ \tau &= \frac{3}{2}\,p\,[(L_d - L_q)\,i_d\,i_q + \lambda\,i_q] \end{aligned}$$

Substitute τ,

$$\frac{d}{dt}\omega = \frac{1}{J}\,(\frac{3}{2}\,p\,[(L_d - L_q)\,i_d\,i_q + \lambda\,i_q] - \tau_D)$$

This can be written in a state space form,

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

where $x \in \mathbb{R}^2 = [\omega,\ \tau_D]^T$ is the state vector, and $u \in \mathbb{R}^3 = [i_d,\ i_q,\ i_d i_q]^T$ is the input vector. $A \in \mathbb{R}^{2\times 2}$, $B \in \mathbb{R}^{3\times 2}$, and $C \in \mathbb{R}^2$ are given by,

$$A = \begin{bmatrix} 0 & a \\ 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & b_1 & b_2 \\ 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1\,0 \end{bmatrix}$$

with, $a = -\frac{1}{J}$, $b_1 = \frac{3p\lambda}{2J}$, and $b_2 = \frac{3p}{2J}(L_d - L_q)$. Therefore, the observer is defined as,

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + G(C\hat{x} - y) \\ \hat{y} &= C\hat{x} \end{aligned}$$

with, G being the observer gain matrix. Since the pair (C, A) is completely observable, it is possible to choose G such as $(A + GC)$ is a Hurwitz matrix. The observer gain matrix G can be found by solving the algebraic Riccati equation or by using a pole placement technique. Therefore, the stability of the observer is guaranteed.

7.5.1 Results

Three simulation runs are carried out to study the controller's performance. For each simulation, the controller's initial parameter estimate vector $\hat{W}_\omega$ is set to zero and the system's response is studied taking into account the machine's tracking speed error e_ω, the d-q axes currents and voltages, the output torque τ, the disturbance estimate $\hat{\tau}_D$, and the adaptive parameter estimate vector $\hat{W}_\omega$. The desired rotor speed is shown in Fig. 7.2.

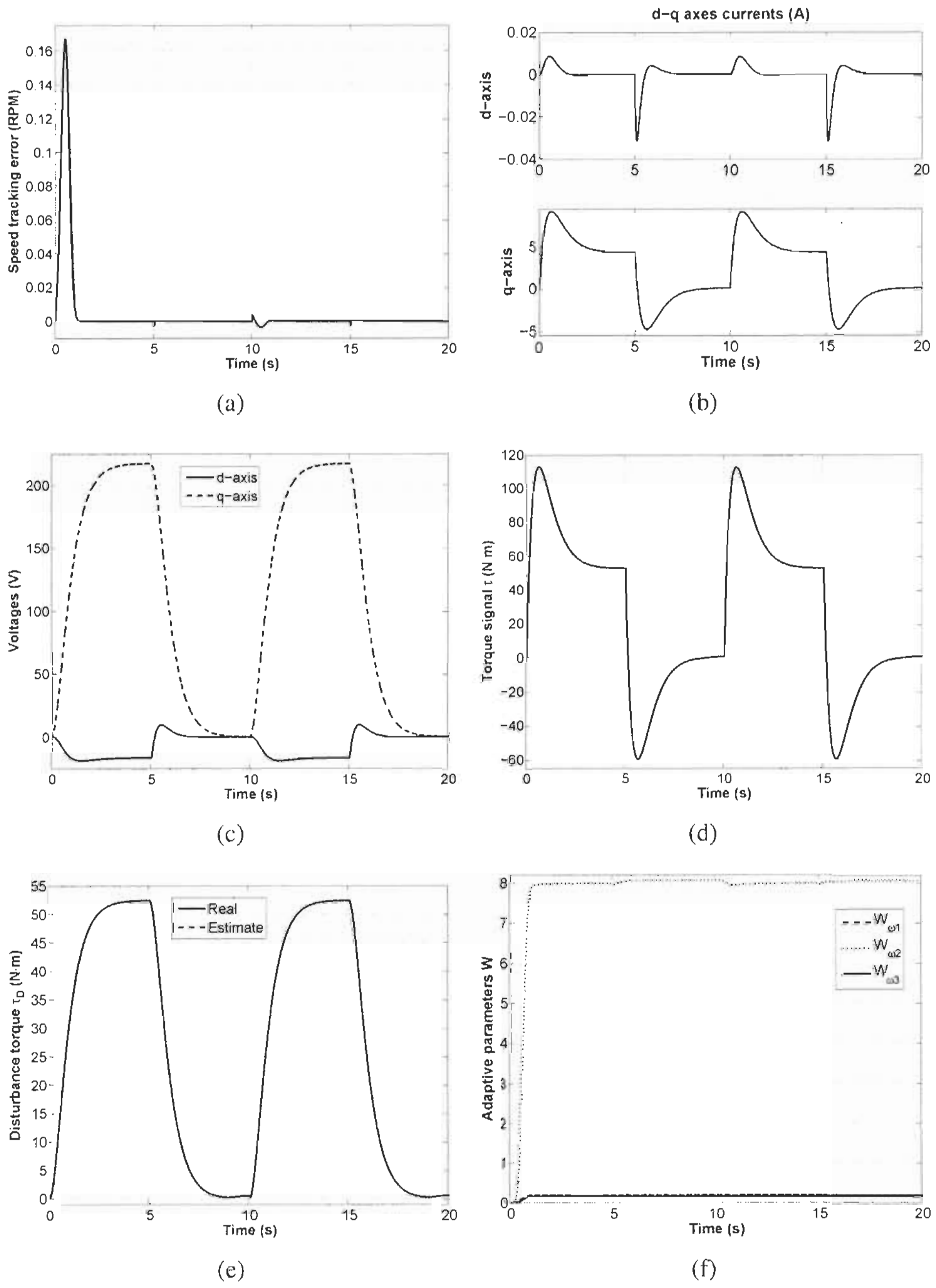
The nominal values (Table 7.1) are used to simulate the machine's dynamics. As shown in Fig. 7.13, the tracking speed error decays gradually before stabilizing within negligible amplitude while accurate disturbance estimation is obtained with the proposed estimation technique. On the other hand, the adaptive control scheme achieves high tracking precision while providing smooth d-q axes currents and voltages. Moreover, the estimation strategy delivers good performance during fast changes in electromagnetic torque by estimating accurately the friction torque as shown in Fig. 7.13(e).

Next, a 100 (N·m) load torque is applied at time t=10s to evaluate the adaptive controller's performance to a disturbance torque. As shown in Fig. 7.14, when the drive is subjected to a disturbance torque, it compensates for this unexpected change by adjusting its parameters. Consequently, the control structure was successful in coping with the load torque variation (Fig. 7.14(e)). Moreover, the speed tracking error remains small, which yielded smooth currents and control signals.

In order to show the ability of the proposed controller in compensating for friction nonlinearities of different magnitudes, the nonlinear Coulomb friction term F_c is magnified 20 times and the machine operates in motoring and regenerating modes to enable zero velocity crossing (i.e., Coulomb friction effect). The results are shown in Fig. 7.15. As it can be seen, the change in the speed reference signal has no effect on the overall control performance. The speed tracking error starts increasing during fast accelerations but stays in an acceptable range. Conventional control techniques tend to overcompensate for these effects and lead to severe tracking errors, limit cycles, chattering, excessive noise, and even instability [17]. As shown in Fig. 7.15, the nonlinearities around zero speed cause a tracking error. However, the controller is able to compensate for the nonlinear friction and no oscillation is observed (Fig. 7.15(e)), which yielded accurate speed tracking in both motoring and regenerating modes.

7.6 Sensorless ANN-Based Adaptive Control

The above adaptive controller shows that stability can be guaranteed even with uncertain dynamics. Therefore, it is possible to extend this result to the unknown dynamics case. Thus, stability can also be achieved for intelligent adaptive controllers, which yields better robustness



(a)

(b)

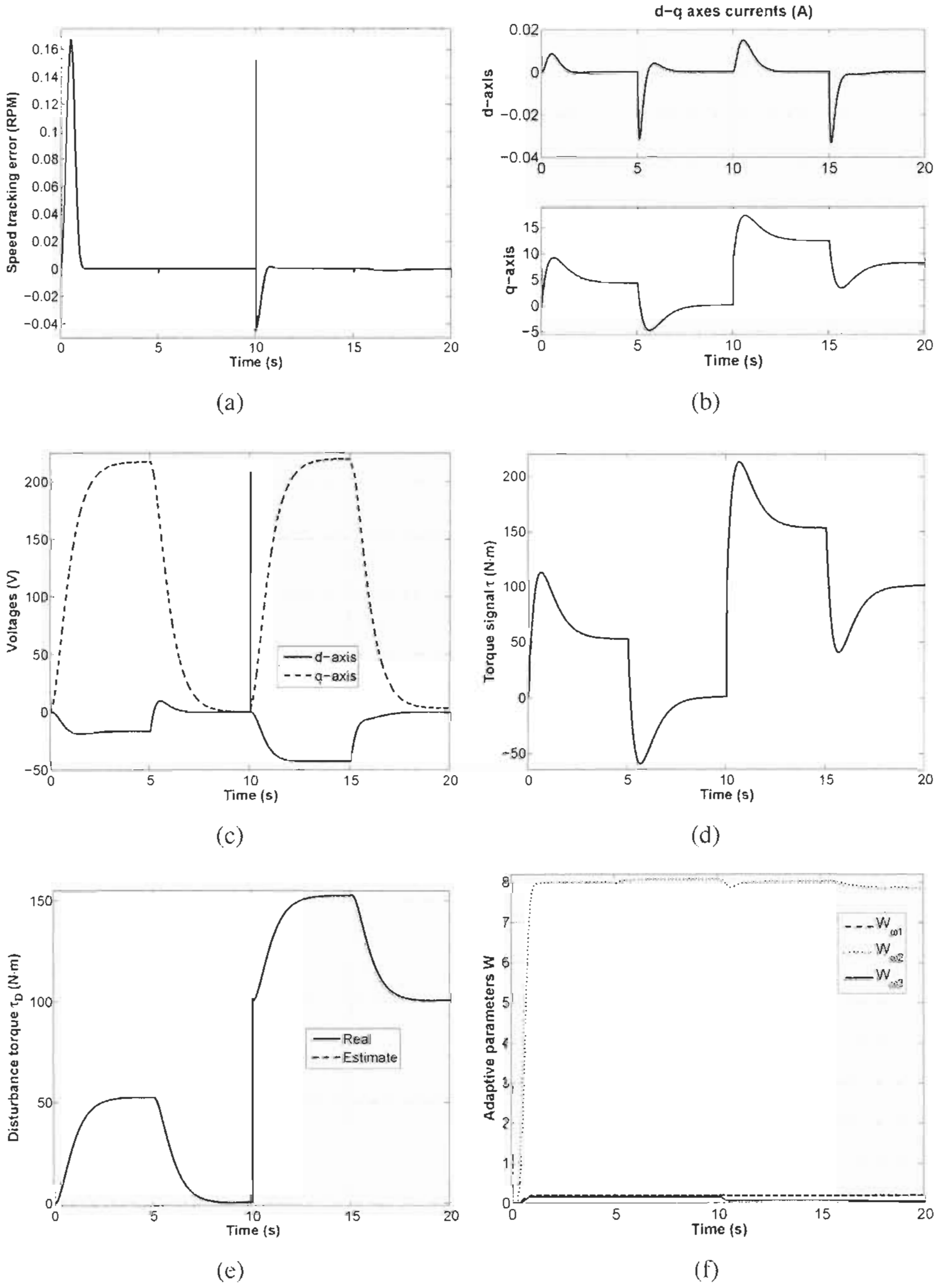
(c)

(d)

(e)

(f)

FIGURE 7.13 – Disturbance estimation response with nominal values : (a) speed tracking error ; (b) d-q axes currents ; (c) d-q axes voltages ; (d) output torque ; (e) disturbance ; and (f) Adaptive parameters $\hat{W}$.



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 7.14 – Disturbance estimation response with load variations : (a) speed tracking error ; (b) d-q axes currents ; (c) d-q axes voltages ; (d) output torque ; (e) disturbance ; and (f) Adaptive parameters $\hat{W}$.

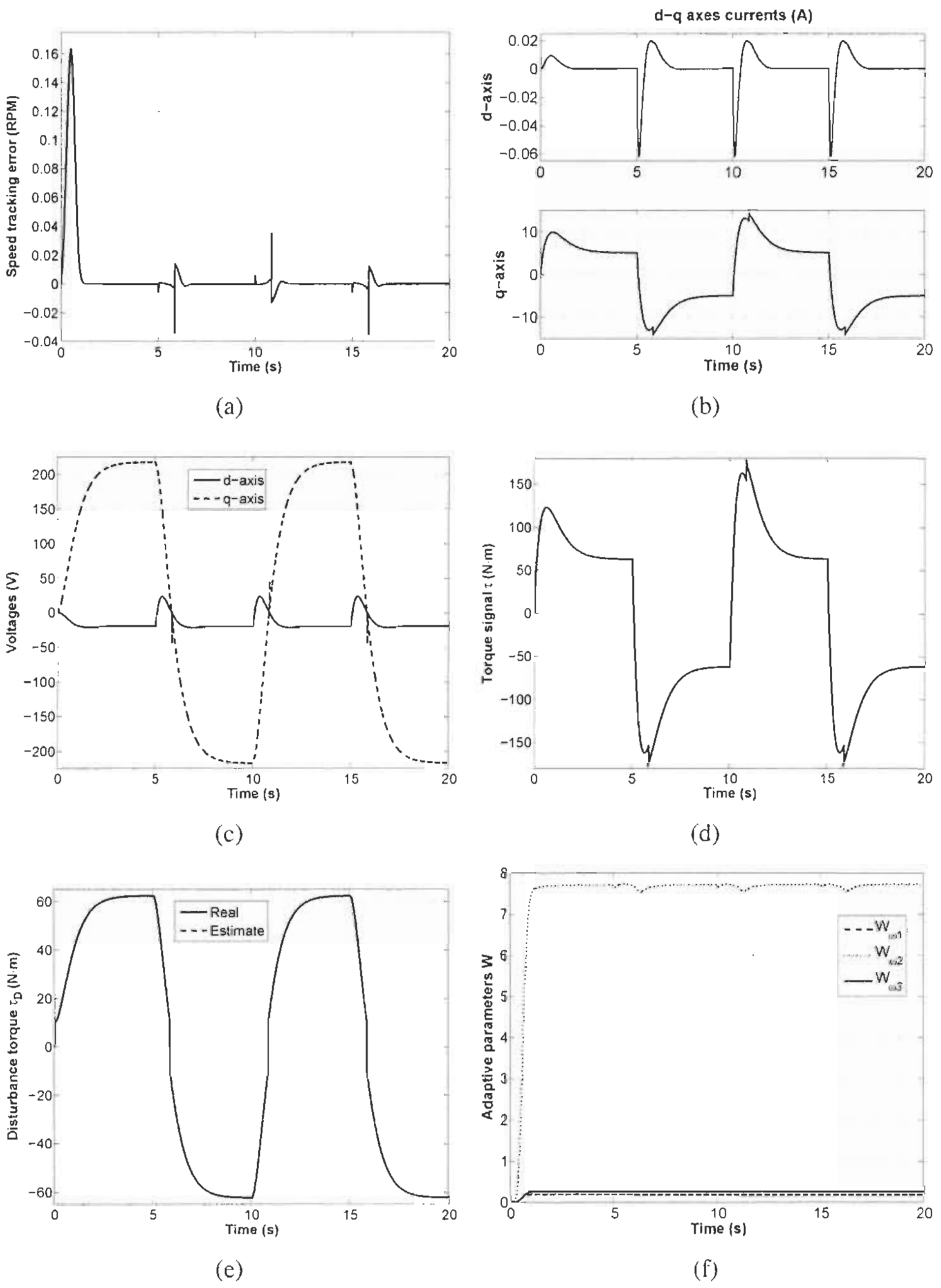


FIGURE 7.15 – Disturbance estimation response with magnified nonlinear friction : (a) speed tracking error ; (b) d-q axes currents ; (c) d-q axes voltages ; (d) output torque ; (e) disturbance ; and (f) Adaptive parameters $\hat{W}$.

to unstructured uncertainties. The adaptive vector control strategy is used and since electrical parameters are known to vary, two neural networks ANN$_d$ and ANN$_q$ are used to cope with parameter variations for d-q axes currents control loop. The resultant control scheme is illustrated in Fig. 7.16. Since conventional speed or position sensors decrease the system's reliability, an ANN-based observer (ANN$_v$) is used to estimate the rotor speed (Fig. 7.16) by only two-phase current measurements i_a and i_b.

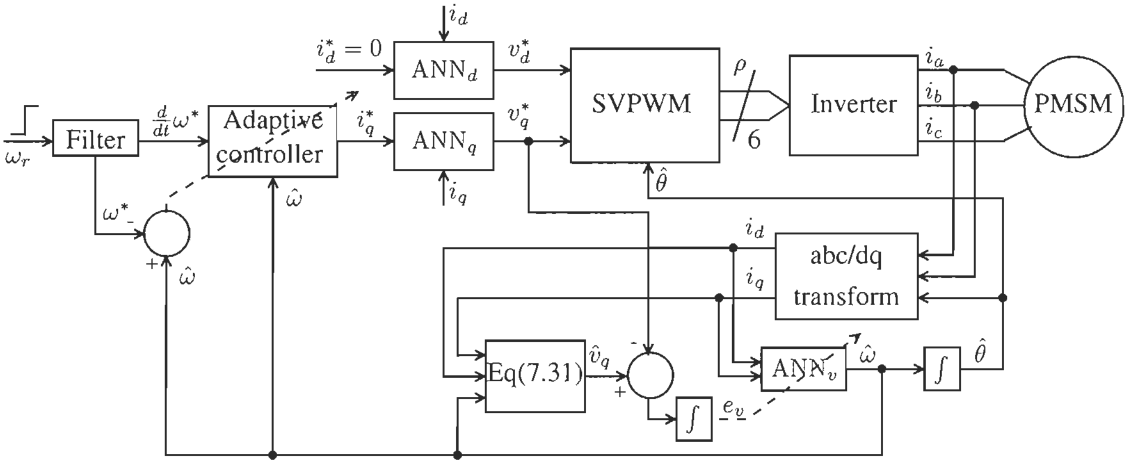


FIGURE 7.16 – Sensorless ANN-based adaptive control scheme

The neural networks ANN$_d$, ANN$_q$, and ANN$_v$ are composed of three layers each : one input layer of two neurons, one hidden layer with six neurons and one neuron for the output layer. The sigmoid function is used as activation function for all neurons except for the output neuron which uses a linear function.

Let $e_\omega = \hat{\omega} - \omega^*$ denote the machine speed error, and $e_d = i_d - i_d^*$ and $e_q = i_q - i_q^*$ the d-q axes current errors, with $\hat{\omega}$ being the velocity estimate and ω^*, i_d^*, and i_q^* being the desired time-dependent speed and currents signals, respectively. Also, let $e_v = \int(\hat{v}_q - v_q^*)dt$ denote the observer error, with v_q^* being the desired time-dependent quadrature voltage and $\hat{v}_q$ being the quadrature voltage estimate defined as follows,

$$\hat{v}_q = R\, i_q + L_q \frac{d}{dt} i_q + p\, \lambda\, \hat{\omega} \tag{7.31}$$

We define the open loop speed estimate as,

$$\hat{\omega} = \frac{1}{p\lambda}\,\hat{v}_q - \frac{R}{p\lambda}\,i_q - \frac{L_q}{p\lambda}\,\frac{d}{dt}i_q \tag{7.32}$$

The output of ANN$_\mathrm{d}$, ANN$_\mathrm{q}$, and ANN$_\mathrm{v}$ can be written as,

$$Y_d = \hat{\Phi}_d^T \hat{W}_d = \Phi_d^T W_d + \varepsilon_d \tag{7.33a}$$

$$Y_q = \hat{\Phi}_q^T \hat{W}_q = \Phi_q^T W_q + \varepsilon_q \tag{7.33b}$$

$$Y_v = \hat{\Phi}_v^T \hat{W}_v = \Phi_v^T W_v + \varepsilon_v \tag{7.33c}$$

where $\hat{\Phi}_d$, $\hat{\Phi}_q$, and $\hat{\Phi}_v$ are the hidden layer's vector of known functions (regressor), with $\hat{W}_d$, $\hat{W}_q$, and $\hat{W}_v$ being the weight matrices. The symbol ε is the output error of ANN$_\mathrm{d}$, ANN$_\mathrm{q}$, and ANN$_\mathrm{v}$ described by, $\varepsilon = \hat{\Phi}^T \hat{W} - \Phi^T W$.

Recall the PMSM formulation (7.1) and (7.2),

$$\frac{d}{dt}\omega = \frac{1}{J}\,(\tau - \tau_F - \tau_L)$$

$$\tau = \frac{3}{2}\,p\,[(L_d - L_q)\,i_d\,i_q + \lambda\,i_q]$$

Substitute τ into the mechanical equation and since the permanent magnet machine torque depends mainly on the quadrature current i_q, it is more convenient to set the direct reference current i_d^* to zero, which will minimize the torque vs. current ratio and increase the motor efficiency. In this design, friction nonlinearities compensation is beyond the scope. Hence, only viscous friction term is considered making $\tau_F = F\omega$. Therefore, we obtain for following model,

$$i_q = \frac{2J}{3p\lambda}\,\dot{\omega} + \frac{2F}{3p\lambda}\,\omega + \frac{2}{3p\lambda}\tau_L - \Delta e_d \tag{7.34}$$

where Δe_d is the uncertainty due to the direct current error e_d described by,

$$\Delta e_d = \frac{(L_d - L_q)i_q}{\lambda}\,e_d$$

It is noteworthy that for surface-mount PMSMs, $\Delta e_d = 0$ since $L_d = L_q$. In this work, the control design is done regardless of the PMSM type and hence, Δe_d is not taken to be equal to zero. The desired dynamics for (7.34) can be represented by the following linear regression,

$$i_q^* = \frac{2J}{3p\lambda}\,\dot{\omega}^* + \frac{2F}{3p\lambda}\,\hat{\omega} + \frac{2}{3p\lambda}\tau_L = \Phi_\omega^T W_\omega \tag{7.35}$$

Take the time-derivative of the error signal e_ω,

$$\dot{e}_\omega = \dot{\hat{\omega}} - \dot{\omega}^*$$

Multiply both sides by $\frac{2J}{3p\lambda}$:

$$\frac{2J}{3p\lambda}\dot{e}_\omega = \frac{2J}{3p\lambda}\dot{\hat{\omega}} - \frac{2J}{3p\lambda}\dot{\omega}^*$$

Add and substract $\frac{2J}{3p\lambda}\dot{\omega}$, substitute $\frac{2J}{3p\lambda}\dot{\omega}$ from (7.34) and use the linear regression (7.35) :

$$\frac{2J}{3p\lambda}\dot{e}_\omega = i_q - \Phi_\omega^T W_\omega + \Delta e_d + \rho_1 \Delta\dot{\omega} + \rho_2 \Delta\omega \tag{7.36}$$

where, $\rho_1 = \frac{2J}{3p\lambda}$, $\rho_2 = \frac{2F}{3p\lambda}$, $\Delta\omega = \hat{\omega} - \omega$, and $\Delta\dot{\omega} = \dot{\hat{\omega}} - \dot{\omega}$.
Recall the PMSM formulation (7.1),

$$v_d = R\, i_d + L_d\, \frac{d}{dt} i_d - L_q\, p\, \omega\, i_q \tag{7.37a}$$

$$v_q = R\, i_q + L_q\, \frac{d}{dt} i_q + L_d\, p\, \omega\, i_d + p\, \lambda\, \omega \tag{7.37b}$$

Therefore, we can also represent the desired dynamics for (7.32) and (7.37) by the following linear regression,

$$v_d^* = R\, i_d + L_d\, \frac{d}{dt} i_d^* - L_q\, p\, \omega\, i_q = \Phi_d^T W_d \tag{7.38a}$$

$$v_q^* = R\, i_q + L_q\, \frac{d}{dt} i_q^* + L_d\, p\, \omega\, i_d + p\, \lambda\, \omega = \Phi_q^T W_q \tag{7.38b}$$

$$\omega = \frac{1}{p\lambda}\, v_q^* - \frac{R}{p\lambda}\, i_q - \frac{L_q}{p\lambda}\, \frac{d}{dt} i_q = \Phi_v^T W_v \tag{7.38c}$$

Recall,

$$\Delta\omega = \hat{\omega} - \omega$$

Substitute $\hat{\omega}$ from (7.37) and ω from (7.38c),

$$\Delta\omega = \frac{1}{p\lambda}\, (\hat{v}_q - v_q^*)$$

Substitute $\dot{e}_v = \hat{v}_q - v_q^*$,

$$\Delta\omega = \frac{1}{p\lambda}\, \dot{e}_v \tag{7.39}$$

Define the observer reference model signal as,

$$s_v = K_{v1}\, \ddot{e}_v + K_{v2}\, \dot{e}_v \tag{7.40}$$

where K_{v1}, and K_{v2} are positive gains.

Take the time-derivative of the error signals e_d, e_q, and e_v,

$$\dot{e}_d = \frac{d}{dt} i_d - \frac{d}{dt} i_d^*$$

$$\dot{e}_q = \frac{d}{dt} i_q - \frac{d}{dt} i_q^*$$

$$\dot{e}_v = \hat{v}_q - v_q^*$$

Multiply the equations by L_d, L_q, and $\frac{1}{p\lambda}$, respectively,

$$\begin{aligned}
L_d\dot{e}_d &= L_d\frac{d}{dt}i_d - L_d\frac{d}{dt}i_d^*\\
L_q\dot{e}_q &= L_q\frac{d}{dt}i_q - L_q\frac{d}{dt}i_q^*\\
\frac{1}{p\lambda}\dot{e}_v &= \frac{1}{p\lambda}\hat{v}_q - \frac{1}{p\lambda}v_q^*
\end{aligned}$$

Substitute $L_d\frac{d}{dt}i_d$, and $L_q\frac{d}{dt}i_q$ from (7.37), and $\frac{1}{p\lambda}\hat{v}_q$ from (7.32) and use the linear regression (7.38) :

$$L_d\dot{e}_d = v_d - \Phi_d^T W_d \tag{7.41a}$$

$$L_q\dot{e}_q = v_q - \Phi_q^T W_q \tag{7.41b}$$

$$\frac{1}{p\lambda}\dot{e}_v = \hat{w} - \Phi_v^T W_v \tag{7.41c}$$

The control and estimation law can be defined as :

$$i_q^* = \Phi_\omega^T \hat{W}_\omega - K_\omega e_\omega - \Delta e_d - s_v \tag{7.42a}$$

$$v_d^* = \hat{\Phi}_d^T \hat{W}_d - K_d e_d \tag{7.42b}$$

$$v_q^* = \hat{\Phi}_q^T \hat{W}_q - K_q e_q - e_\omega \tag{7.42c}$$

$$\hat{w} = \hat{\Phi}_v^T \hat{W}_v - K_v e_v \tag{7.42d}$$

where K_ω, K_d, K_q, and K_v are positive gains.

Theorem 15 *Consider a nonlinear system in the form (7.1)-(7.2) with the control and estimation law (7.42). The closed-loop system's stability is achieved with the following adaptation* *law :*

$$\dot{\hat{W}}_\omega = -\Gamma_\omega \, \Phi_\omega \, e_\omega \tag{7.43a}$$

$$\dot{\hat{W}}_d = -\Gamma_d \, \hat{\Phi}_d \, e_d \tag{7.43b}$$

$$\dot{\hat{W}}_q = -\Gamma_q \, \hat{\Phi}_q \, e_q \tag{7.43c}$$

$$\dot{\hat{W}}_v = -\Gamma_v \, \hat{\Phi}_v \, e_v \tag{7.43d}$$

where $\Gamma_x = diag(\gamma_1, \gamma_2, \ldots, \gamma_n)$ and γ_i is a positive constant.

Proof 9 *Choose the following Lyapunov candidate function :*

$$V = \frac{1}{2}\{\frac{2J}{3p\lambda}e_\omega^2 + L_d e_d^2 + L_q e_q^2 + \frac{1}{p\lambda}e_v^2 + \tilde{W}_\omega^T \Gamma_\omega^{-1} \tilde{W}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \tilde{W}_d + \tilde{W}_q^T \Gamma_q^{-1} \tilde{W}_q + \tilde{W}_v^T \Gamma_v^{-1} \tilde{W}_v\}$$

Take the time-derivative of V :

$$\dot{V} = \frac{2J}{3p\lambda}\dot{e}_\omega e_\omega + L_d \dot{e}_d e_d + L_q \dot{e}_q e_q + \frac{1}{p\lambda}\dot{e}_v e_v + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d + \tilde{W}_q^T \Gamma_q^{-1} \dot{\hat{W}}_q + \tilde{W}_v^T \Gamma_v^{-1} \dot{\hat{W}}_v$$

The parameters vectors W_ω, W_d, W_q, and W_v are assumed to be constant, i.e., $\dot{\tilde{W}} = \dot{\hat{W}}$. Substitute $\frac{2J}{3p\lambda}\dot{e}_\omega$ from (7.36), $L_d \dot{e}_d$, $L_q \dot{e}_q$, and $\frac{1}{p\lambda}\dot{e}_v$ from (7.41),

$$\dot{V} = \{i_q - \Phi_\omega^T W_\omega + \Delta e_d + \rho_1 \Delta\dot{\omega} + \rho_2 \Delta\omega\} e_\omega + \{v_d - \Phi_d^T W_d\} e_d + \{v_q - \Phi_q^T W_q\} e_q$$
$$+ \{\hat{w} - \Phi_v^T W_v\} e_v + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d + \tilde{W}_q^T \Gamma_q^{-1} \dot{\hat{W}}_q + \tilde{W}_v^T \Gamma_v^{-1} \dot{\hat{W}}_v$$

Setting,

$$
\begin{aligned}
i_q &= i_q^* + e_q \\
v_d &= v_d^* + \Delta v_d \\
v_q &= v_q^* + \Delta v_q
\end{aligned}
$$

with $\Delta v_d = v_d - v_d^*$ *and* $\Delta v_q = v_q - v_q^*$*, yields,*

$$
\begin{aligned}
\dot{V} = {} & \{i_q^* + e_q - \Phi_\omega^T W_\omega + \Delta e_d + \rho_1 \Delta\dot{\omega} + \rho_2 \Delta\omega\} e_\omega + \{v_d^* + \Delta v_d - \Phi_d^T W_d\} e_d \\
& + \{v_q^* + \Delta v_q - \Phi_q^T W_q\} e_q + \{\hat{w} - \Phi_v^T W_v\} e_v + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d \\
& + \tilde{W}_q^T \Gamma_q^{-1} \dot{\hat{W}}_q + \tilde{W}_v^T \Gamma_v^{-1} \dot{\hat{W}}_v
\end{aligned}
$$

Since the inverter operates at a smaller period than the motor electrical time constant, the inverter is assumed to reproduce accurately the reference voltages v_d^* *and* v_q^**, which reduces the number of sensors and makes* $\Delta v_d = \Delta v_q = 0$*. Setting the control and estimation law as in (7.42) yields,*

$$
\begin{aligned}
\dot{V} = {} & \Phi_\omega^T \tilde{W}_\omega e_\omega - s_v e_\omega + \rho_1 \Delta\dot{\omega} e_\omega + \rho_2 \Delta\omega e_\omega + \varepsilon_d e_d + \varepsilon_q e_q + \varepsilon_v e_v + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega \\
& + \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d + \tilde{W}_q^T \Gamma_q^{-1} \dot{\hat{W}}_q + \tilde{W}_v^T \Gamma_v^{-1} \dot{\hat{W}}_v - K_\omega e_\omega^2 - K_d e_d^2 - K_q e_q^2 - K_v e_v^2
\end{aligned}
$$

Substitute s_v *from (7.40). Set* $K_{v1} = \frac{\rho 1}{p\lambda}$*, and* $K_{v2} = \frac{\rho 2}{p\lambda}$ *and substitute* $\Delta\omega$*, and* $\Delta\dot{\omega}$ *from (7.39),*

$$
\begin{aligned}
\dot{V} = {} & \Phi_\omega^T \tilde{W}_\omega e_\omega + \varepsilon_d e_d + \varepsilon_q e_q + \varepsilon_v e_v + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d + \tilde{W}_q^T \Gamma_q^{-1} \dot{\hat{W}}_q \\
& + \tilde{W}_v^T \Gamma_v^{-1} \dot{\hat{W}}_v - K_\omega e_\omega^2 - K_d e_d^2 - K_q e_q^2 - K_v e_v^2
\end{aligned}
$$

Substitute ε_d, ε_q, and ε_v from (6.26),

$$
\begin{aligned}
\dot{V} = {} & \Phi_\omega^T \tilde{W}_\omega e_\omega + \hat{\Phi}_d^T \tilde{W}_d e_d + \tilde{\Phi}_d^T W_d e_d + \hat{\Phi}_q^T \tilde{W}_q e_q + \tilde{\Phi}_q^T W_q e_q + \hat{\Phi}_v^T \tilde{W}_v e_v + \tilde{\Phi}_v^T W_v e_v \\
& + \tilde{W}_\omega^T \Gamma_\omega^{-1} \dot{\hat{W}}_\omega + \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d + \tilde{W}_q^T \Gamma_q^{-1} \dot{\hat{W}}_q + \tilde{W}_v^T \Gamma_v^{-1} \dot{\hat{W}}_v \\
& - K_\omega e_\omega^2 - K_d e_d^2 - K_q e_q^2 - K_v e_v^2
\end{aligned}
$$

Setting the adaptation law as in (7.43) implies that,

$$
\dot{V} = \tilde{\Phi}_d^T W_d e_d + \tilde{\Phi}_q^T W_q e_q + \tilde{\Phi}_v^T W_v e_v - K_\omega e_\omega^2 - K_d e_d^2 - K_q e_q^2 - K_v e_v^2
$$

Recall Young's inequality [80],

$$
2ab \leq \frac{1}{\alpha} a^2 + \alpha b^2 \qquad \forall a, b \in \mathbb{R} \text{ and } \forall \alpha > 0
$$

Therefore,

$$
\begin{aligned}
\dot{V} \leq {} & \frac{1}{2\alpha} (\tilde{\Phi}_d^T W_d)^2 + \frac{\alpha}{2} e_d^2 + \frac{1}{2\alpha} (\tilde{\Phi}_q^T W_q)^2 + \frac{\alpha}{2} e_q^2 + \frac{1}{2\alpha} (\tilde{\Phi}_v^T W_v)^2 + \frac{\alpha}{2} e_v^2 \\
& - K_\omega e_\omega^2 - K_d e_d^2 - K_q e_q^2 - K_v e_v^2
\end{aligned}
$$

Hence,

$$
\dot{V} \leq \frac{1}{2\alpha} \Psi - K_\omega e_\omega^2 - \left(K_d - \frac{\alpha}{2} \right) e_d^2 - \left(K_q - \frac{\alpha}{2} \right) e_q^2 - \left(K_v - \frac{\alpha}{2} \right) e_v^2
$$

with,

$$
\Psi = (\tilde{\Phi}_d^T W_d)^2 + (\tilde{\Phi}_q^T W_q)^2 + (\tilde{\Phi}_v^T W_v)^2
$$

Setting,

$$K_d = K_q = K_v = \frac{\alpha}{2} + \beta$$
$$K_\omega = \beta$$

where β is positive gain yields,

$$\dot{V} \leq \frac{1}{2\alpha}\Psi - \beta(e_\omega^2 + e_d^2 + e_q^2 + e_v^2)$$

So, it is possible to choose α and β so that $\dot{V} \leq 0$. Therefore, the system is stable in the sense of Lyapunov and converges to a small neighborhood of the origin, which is a region defined by the neural network approximation errors $\tilde{\Phi}$. This region gets smaller as $\tilde{\Phi} \to 0$. Theoretically, it can be made arbitrarily small by increasing the gains.

7.6.1 Results

Three simulation runs are carried out to study the proposed controller's performance. For each simulation, the system's response is studied taking into account the machine's speed estimation and tracking errors, the d-q axes currents error and voltages v_d and v_q, the output torque τ, and the parameters estimate $\hat{W}_\omega$. The desired rotor speed is shown in Fig. 7.17.

In the first simulation, the aforementioned nominal values are used to simulate the machine's dynamics. As shown in Fig. 7.18(b), the speed tracking error decays gradually before stabilizing within a negligible amplitude. On the other hand, satisfactory speed estimation is obtained with the proposed observer (less than 0.1%) while the controller provides smooth d-q axes currents and torque signal. Moreover, the estimated rotor speed follows the actual speed closely during fast changes in the electromagnetic torque. It is noteworthy that at low speed, a very low speed estimation error is obtained with the proposed speed observer. Furthermore, the adaptive controller copes with parameter uncertainties and achieves fast and precise parameters estimate convergence and tracking.

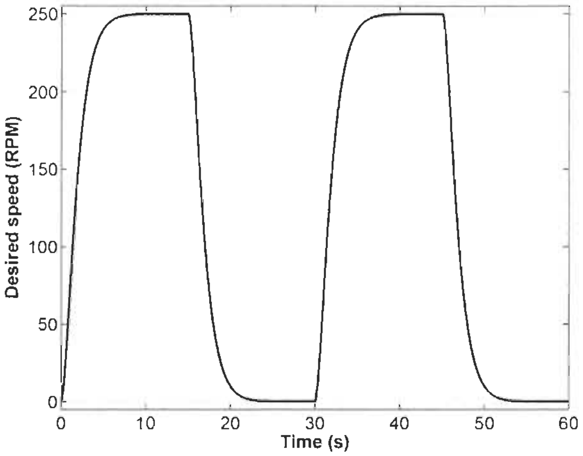


FIGURE 7.17 – ANN-based adaptive control speed reference signal

In the next simulation, a 100 (N·m) load torque is applied at time t=0s to evaluate the adaptive controller's performance to a torque disturbance. As shown in Fig. 7.19, when the drive is subjected to a torque disturbance, it compensates for this unexpected change by adjusting its parameters. Consequently, the controller was successful in coping with the load torque variation. Moreover, the speed tracking error remains small, which yielded smooth currents and control signal. However, a speed estimation error increase is noticed and higher torque is required for the heavier load. The low-speed performance of the rotor speed observer is clearly shown in this simulation.

The following simulation is meant to show the modularity of the proposed controller in motoring and regenerating modes. For that, the speed reference signal is subjected to a direction change, the results are shown in Fig. 7.20. As it can be seen, the change in the speed reference signal has little effect on the overall control performance. The speed tracking and estimation errors start increasing during fast accelerations but stay in an acceptable range. It maintains a similar behavior as in in Fig. 7.18, the estimated rotor speed follows the actual speed smoothly in the whole speed range in both motoring and regenerating modes. Furthermore, the ability of the proposed adaptive controller to cope with speed direction change is shown in this case.

Next, a simulation is carried out to study the proposed observer's ability to sustain various parametric uncertainties. The controller's performance is similar to the nominal case, which is

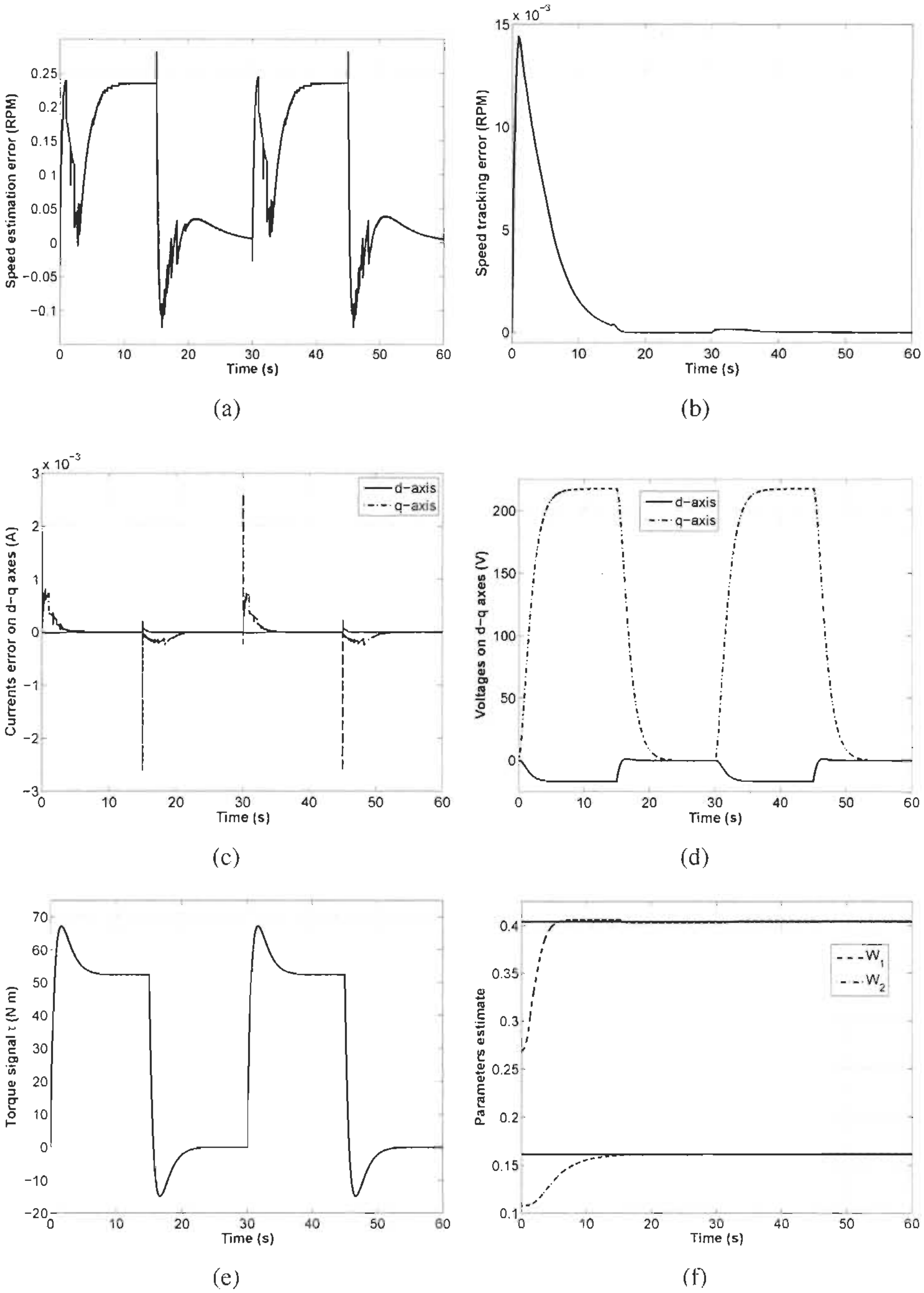
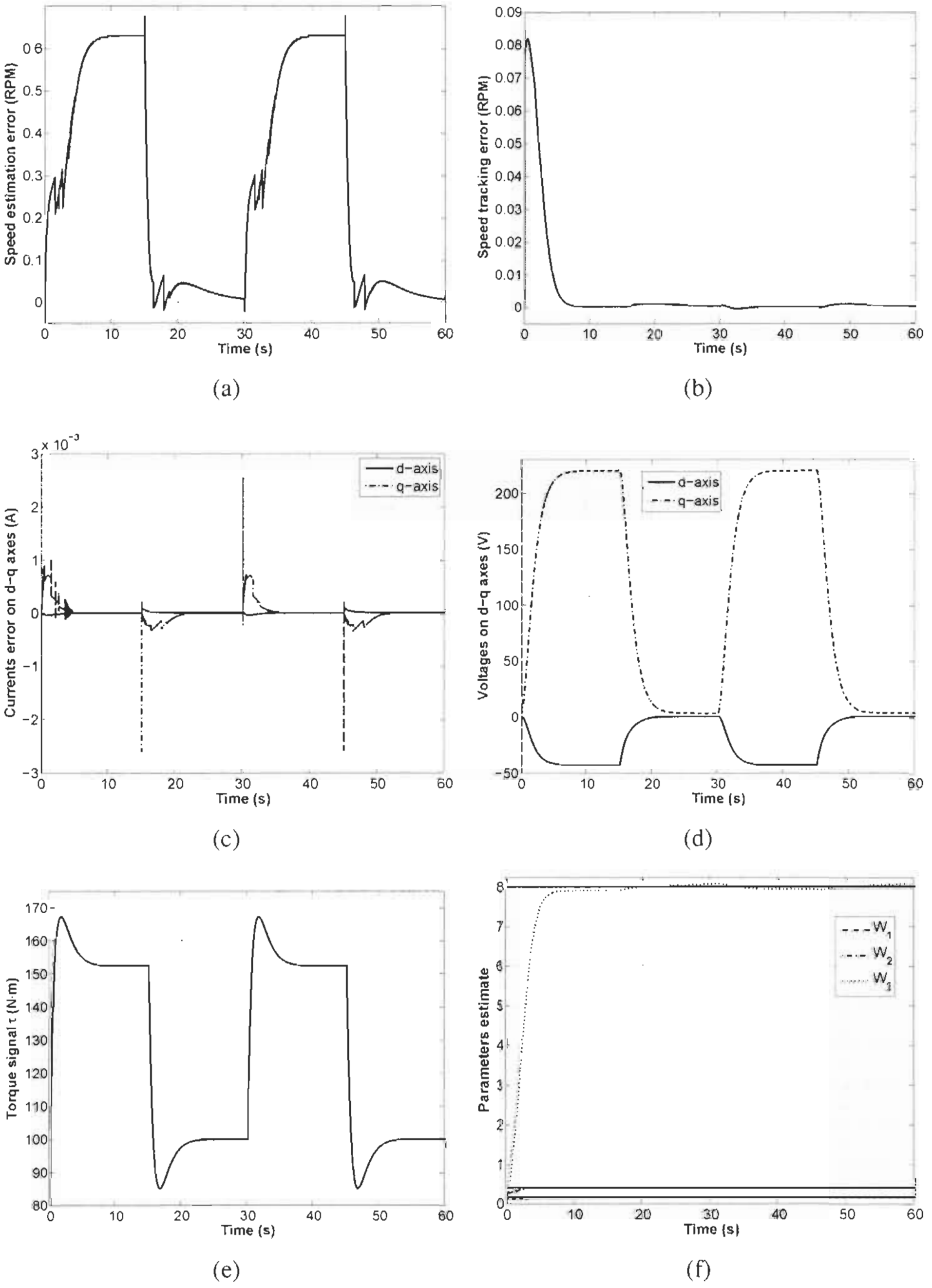


FIGURE 7.18 – ANN-based adaptive control response with nominal values : (a) speed estimation error ; (b) speed tracking error ; (c) d-q axes currents error ; (d) d-q axes voltages ; (e) controller's torque τ ; and (f) parameters estimate $\hat{W}_{\omega}$.



(a)

(b)

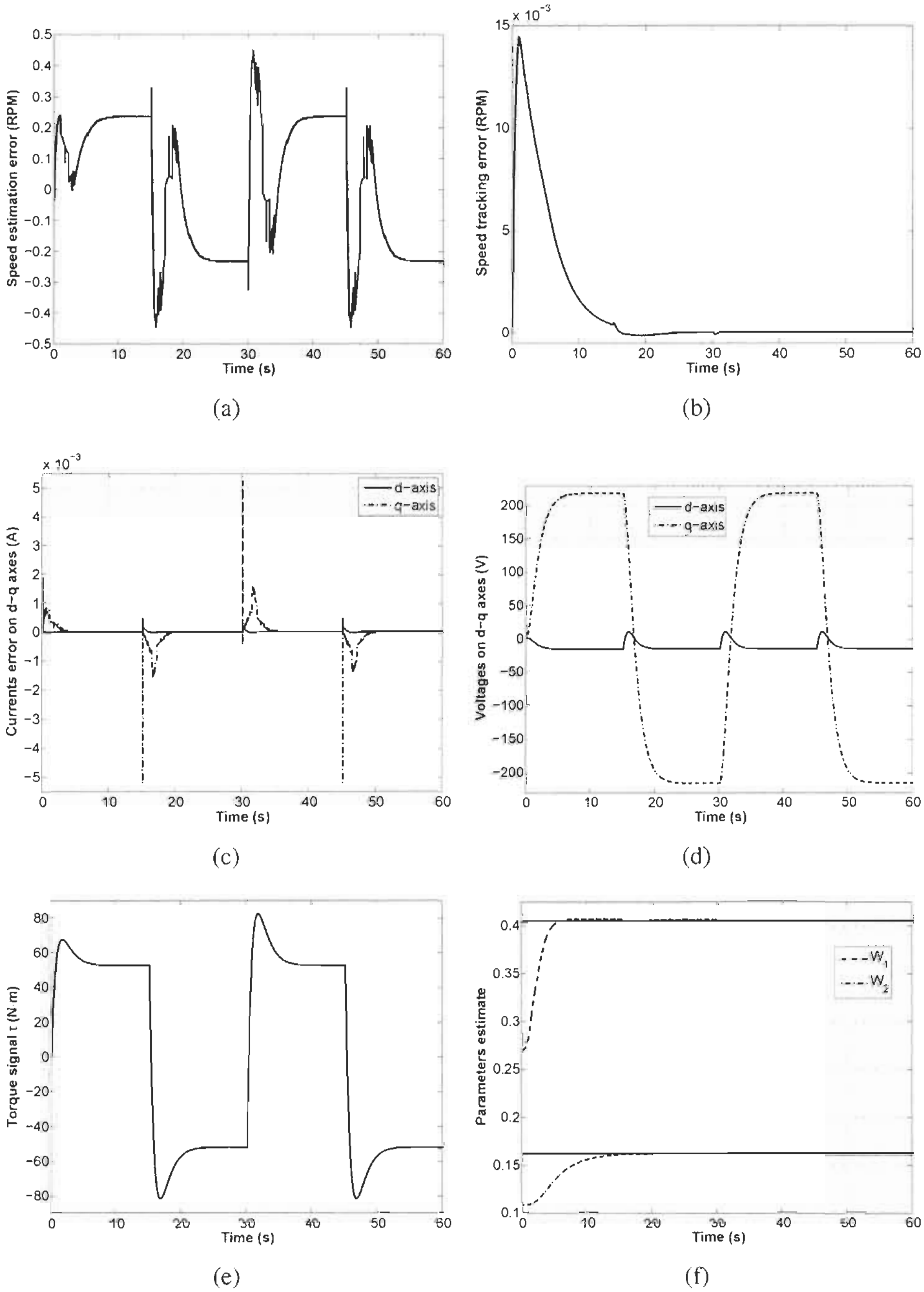
(c)

(d)

(e)

(f)

FIGURE 7.19 – ANN-based adaptive control response with load torque disturbance : (a) speed estimation error ; (b) speed tracking error ; (c) d-q axes currents error ; (d) d-q axes voltages ; (e) controller's torque τ ; and (f) parameters estimate $\hat{W}_{\omega}$.



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 7.20 – ANN-based adaptive control response in motor and generator mode : (a) speed estimation error ; (b) speed tracking error ; (c) d-q axes currents error ; (d) d-q axes voltages ; (e) controller's torque τ ; and (f) parameters estimate $\hat{W}_{\omega}$.

expected since the control design is independent of the machine's mechanical and electrical parameters. However, the observer is based on equation (7.1b), which includes parameters R, L_d, L_q, λ, and p. Therefore, the machine's electrical parameters, R, L_d, L_q, and λ are increased one at a time by 25% their nominal values. The observer's performance is depicted in Fig. 7.21. It is noteworthy that from equation (7.1b), $(L_d\ p\ i_d + p\ \lambda)\ \omega$ is the most significant term at $\omega_{nominal}$. Thus, the variation of R, and L_q does not have much effect on the observer's performance as it is shown in Fig. 7.21(a), and Fig. 7.21(c). Also, Fig. 7.21(b) shows good performance since setting i_d^* to zero makes the observer less sensitive to L_d. However, the flux λ remains the key parameter in achieving high estimation accuracy as it is observed in Fig. 7.21(d). This parameter can be estimated online using one of the so many techniques available in the litterature.

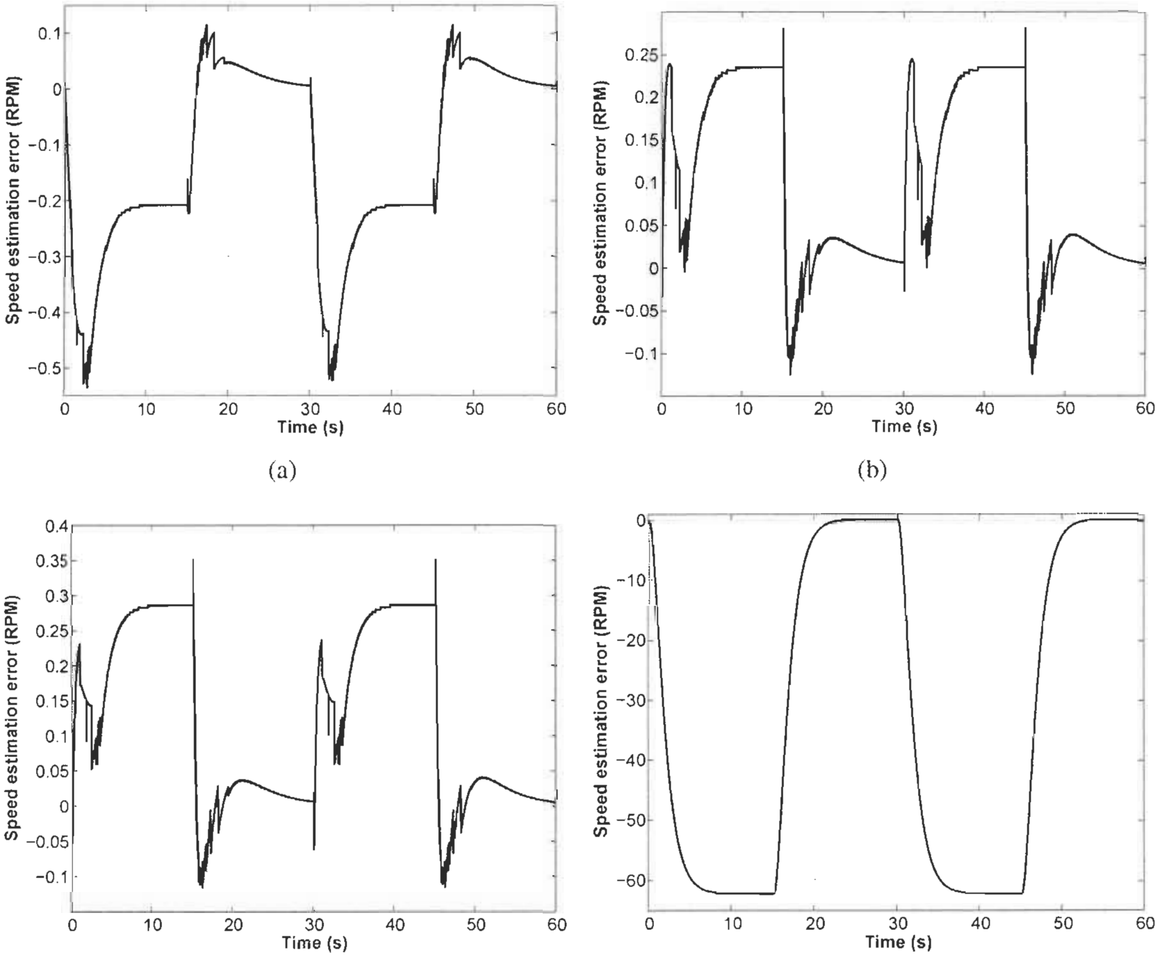
7.7 Sensorless ANN-Based Vector Control

The adaptive controller is removed from the previous adaptive ANN-based control technique and the ANN's approximation property is exploited to design a reduced complexity control scheme able to cope with parametric uncertainties. The resultant control scheme is illustrated in Fig. 7.22. As such, a neural network controller ANN$_d$ keeps i_d at zero constant while a second neural network controller ANN$_q$ achieves precise speed tracking by machine's inverse dynamics approximation. On the other hand, the neural network observer ANN$_\nu$ is used for speed estimation.

Define the quadrature voltage estimate as,

$$\hat{v}_q = R\ i_q + L_q\ \frac{d}{dt} i_q + \sigma\ \hat{\omega} \tag{7.44}$$

where, $\sigma = p(L_d i_d + \lambda)$. We define the open loop speed estimation model, which is to be



(a)

(b)

(c)

(d)

FIGURE 7.21 – ANN-based adaptive control response with variation of : (a) resistance R ; (b) inductance L_d ; (c) inductance L_q ; and (d) flux λ.

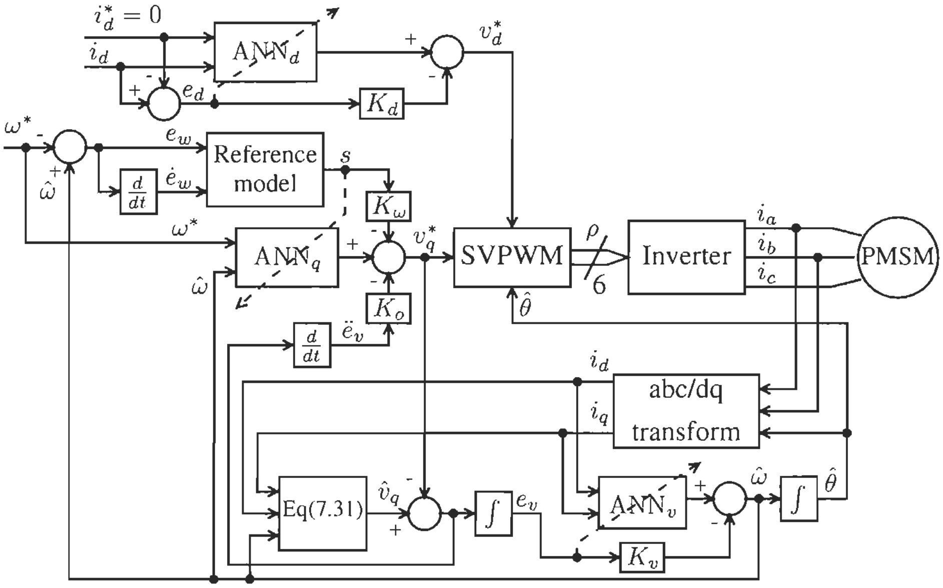


FIGURE 7.22 – Sensorless ANN-based vector control scheme

reproduced by the neural network ANN$_v$ through its adaptation law,

$$\hat{\omega} = \frac{1}{\sigma}\,\hat{v}_q - \frac{R}{\sigma}\,i_q - \frac{L_q}{\sigma}\,\frac{d}{dt}i_q \tag{7.45}$$

Define the reference model signal s,

$$s = e_\omega + \psi\,\dot{e}_\omega = \omega - \omega_t \tag{7.46}$$

$$\omega_t = \omega^* - \psi\,\dot{e}_\omega$$

with ψ being a positive constant.

The output of ANN$_d$, ANN$_q$, and ANN$_v$ can be written as,

$$Y_d = \hat{\Phi}_d^T \hat{W}_d = \Phi_d^T W_d + \varepsilon_d \tag{7.47a}$$

$$Y_q = \hat{\Phi}_q^T \hat{W}_q = \Phi_q^T W_q + \varepsilon_q \tag{7.47b}$$

$$Y_v = \hat{\Phi}_v^T \hat{W}_v = \Phi_v^T W_v + \varepsilon_v \tag{7.47c}$$

where $\hat{\Phi}_d$, $\hat{\Phi}_q$, and $\hat{\Phi}_v$ are the hidden layer's vector of known functions (regressor), with $\hat{W}_d$, $\hat{W}_q$, and $\hat{W}_v$ being the weight matrices. The symbol ε is the output error of ANN$_d$, ANN$_q$, and ANN$_v$ described in (6.26).

Recall the formulation (7.14),

$$v_q = \mu\ \dot{\omega} - \frac{R}{\lambda}\ (L_d - L_q)\ i_d\ i_q + L_q\ \frac{d}{dt} i_q + L_d\ p\ \omega\ i_d + p\ \lambda\ \omega + \frac{\mu}{J}\ (\tau_F + \tau_L) \tag{7.48}$$

Recall the PMSM formulation (7.1),

$$v_d = R\ i_d + L_d\ \frac{d}{dt} i_d - L_q\ p\ \omega\ i_q \tag{7.49}$$

The desired dynamics for (7.45), (7.48) and (7.49) can be represented by the following linear regression,

$$\mu\ \dot{\omega}_t - \frac{R}{\lambda}\ (L_d - L_q)\ i_d\ i_q + L_q\ \frac{d}{dt} i_q + L_d\ p\ \omega\ i_d + p\ \lambda\ \omega + \frac{\mu}{J}\ (\tau_F + \tau_L) = \Phi_q^T W_q \tag{7.50a}$$

$$R\ i_d + L_d\ \frac{d}{dt} i_d^* - L_q\ p\ \omega\ i_q = \Phi_d^T W_d \tag{7.50b}$$

$$\frac{1}{\sigma}\ v_q^* - \frac{R}{\sigma}\ i_q - \frac{L_q}{\sigma}\ \frac{d}{dt} i_q = \Phi_v^T W_v \tag{7.50c}$$

The control law can be defined as :

$$v_q^* = \hat{\Phi}_q^T \hat{W}_q - K_\omega s - K_o \ddot{e}_v \tag{7.51a}$$

$$v_d^* = \hat{\Phi}_d^T \hat{W}_d - K_d e_d \tag{7.51b}$$

$$\hat{\omega} = \hat{\Phi}_v^T \hat{W}_v - K_v e_v \tag{7.51c}$$

where K_ω, K_o, K_d, and K_v are positive gains.

Theorem 16 *Consider a nonlinear system in the form (7.1)-(7.2) with the control law (7.51).*

The closed-loop system's stability is achieved with the following adaptation law :

$$\dot{\hat{W}}_q = -\Gamma_q \, \hat{\Phi}_q \, s \tag{7.52a}$$

$$\dot{\hat{W}}_d = -\Gamma_d \, \hat{\Phi}_d \, e_d \tag{7.52b}$$

$$\dot{\hat{W}}_v = -\Gamma_v \, \hat{\Phi}_v \, e_v \tag{7.52c}$$

where $\Gamma_x = diag(\gamma_1, \gamma_2, \ldots, \gamma_n)$ and γ_i is a positive constant.

Proof 10 *Take the time-derivative of s,*

$$\dot{s} = \dot{\hat{\omega}} - \dot{\omega}_t$$

Add and substract $\dot{\omega}$.

$$\dot{s} = \dot{\omega} - \dot{\omega}_t + \Delta\dot{\omega}$$

where $\Delta\dot{\omega} = \dot{\hat{\omega}} - \dot{\omega}$.

Multiply both sides by μ :

$$\mu\dot{s} = \mu\dot{\omega} - \mu\dot{\omega}_t + \mu\Delta\dot{\omega}$$

Substitute $\mu\dot{\omega}$ from (7.48) and use the linear regression (7.50) :

$$\mu\dot{s} = v_q - \Phi_q^T W_q + \mu\Delta\dot{\omega} \tag{7.53}$$

Take the time-derivative of the error signal e_d,

$$\dot{e}_d = \frac{d}{dt} i_d - \frac{d}{dt} i_d^*$$

Multiply both sides by L_d,

$$L_d\dot{e}_d = L_d\frac{d}{dt} i_d - L_d\frac{d}{dt} i_d^*$$

Substitute $L_d \frac{d}{dt} i_d$ *from (7.49) and use the linear regression (7.50b),*

$$L_d \dot{e}_d = v_d - \Phi_d^T W_d \tag{7.54}$$

Recall,

$$\Delta\omega = \hat{\omega} - \omega$$

Substitute $\hat{\omega}$ *from (7.45) and* ω *from (7.50c),*

$$\Delta\omega = \frac{1}{\sigma}\left(\hat{v}_q - v_q^*\right)$$

Substitute $\dot{e}_v = \hat{v}_q - v_q^*$*,*

$$\Delta\omega = \frac{1}{\sigma}\,\dot{e}_v \tag{7.55}$$

Substract (7.50c) from (7.45),

$$\frac{1}{\sigma}\left(\hat{v}_q - v_q^*\right) = \hat{\omega} - \Phi_v^T W_v$$

Since the inverter operates at a smaller period than the motor electrical time constant, a sensible practical assumption is that the inverter reproduces accurately the reference voltages v_d^**, and* v_q^**, which reduces the number of sensors and makes* $v_d^* = v_d$*, and* $v_q^* = v_q$*. Considering the definition of* $\dot{e}_v$*, we obtain,*

$$\frac{1}{\sigma}\dot{e}_v = \hat{\omega} - \Phi_v^T W_v \tag{7.56}$$

Choose the following Lyapunov candidate function :

$$V = \frac{1}{2}\{\mu s^2 + L_d e_d^2 + \frac{1}{\sigma}e_v^2 + \tilde{W}_q^T \Gamma_q^{-1} \tilde{W}_q + \tilde{W}_d^T \Gamma_d^{-1} \tilde{W}_d + \tilde{W}_v^T \Gamma_v^{-1} \tilde{W}_v\}$$

Take the time-derivative of V :

$$\dot{V} = \mu \dot{s} s + L_d \dot{e}_d e_d + \frac{1}{\sigma}\dot{e}_v e_v + \tilde{W}_q^T \Gamma_q^{-1} \dot{\hat{W}}_q + \tilde{W}_d^T \Gamma_d^{-1} \dot{\hat{W}}_d + \tilde{W}_v^T \Gamma_v^{-1} \dot{\hat{W}}_v$$

The parameters vectors W_q, W_d, and W_v are assumed to be constant, i.e., $\dot{\tilde{W}} = \dot{\hat{W}}$. Substitute $\mu\dot{s}$ from (7.53), $L_d\dot{e}_d$ from (7.54) and $\frac{1}{\sigma}\dot{e}_v$ from (7.56),

$$\dot{V} = \{v_q - \Phi_q^T W_q + \mu\Delta\dot{\omega}\}s + \{v_d - \Phi_d^T W_d\}e_d + \{\hat{\omega} - \Phi_v^T W_v\}e_v \\ + \tilde{W}_q^T \Gamma_q^{-1}\dot{\hat{W}}_q + \tilde{W}_d^T \Gamma_d^{-1}\dot{\hat{W}}_d + \tilde{W}_v^T \Gamma_v^{-1}\dot{\hat{W}}_v$$

Setting the control law as in (7.51) yields,

$$\dot{V} = \varepsilon_q s + (\mu\Delta\dot{\omega} - K_o\ddot{e}_v)s + \varepsilon_d e_d + \varepsilon_v e_v + \tilde{W}_q^T \Gamma_q^{-1}\dot{\hat{W}}_q + \tilde{W}_d^T \Gamma_d^{-1}\dot{\hat{W}}_d + \tilde{W}_v^T \Gamma_v^{-1}\dot{\hat{W}}_v \\ - K_\omega s^2 - K_d e_d^2 - K_v e_v^2$$

Set $K_o = \frac{\mu}{\sigma}$ and substitute $\Delta\dot{\omega}$ from (7.55),

$$\dot{V} = \varepsilon_q s + \varepsilon_d e_d + \varepsilon_v e_v + \tilde{W}_q^T \Gamma_q^{-1}\dot{\hat{W}}_q + \tilde{W}_d^T \Gamma_d^{-1}\dot{\hat{W}}_d + \tilde{W}_v^T \Gamma_v^{-1}\dot{\hat{W}}_v \\ - K_\omega s^2 - K_d e_d^2 - K_v e_v^2$$

Substitute ε_q, ε_d, and ε_v from (6.26),

$$\dot{V} = \hat{\Phi}_q^T \tilde{W}_q s + \tilde{\Phi}_q^T W_q s + \hat{\Phi}_d^T \tilde{W}_d e_d + \tilde{\Phi}_d^T W_d e_d + \hat{\Phi}_v^T \tilde{W}_v e_v + \tilde{\Phi}_v^T W_v e_v \\ + \tilde{W}_q^T \Gamma_q^{-1}\dot{\hat{W}}_q + \tilde{W}_d^T \Gamma_d^{-1}\dot{\hat{W}}_d + \tilde{W}_v^T \Gamma_v^{-1}\dot{\hat{W}}_v - K_\omega s^2 - K_d e_d^2 - K_v e_v^2$$

Setting the adaptation law as in (7.52) implies that,

$$\dot{V} = \tilde{\Phi}_q^T W_q s + \tilde{\Phi}_d^T W_d e_d + \tilde{\Phi}_v^T W_v e_v - K_\omega s^2 - K_d e_d^2 - K_v e_v^2$$

Recall Young's inequality [80],

$$2ab \leq \frac{1}{\alpha}a^2 + \alpha b^2 \qquad \forall a, b \in \mathbb{R} \text{ and } \forall \alpha > 0$$

Therefore,

$$\dot{V} \leq \frac{1}{2\alpha}(\tilde{\Phi}_q^T W_q)^2 + \frac{\alpha}{2}s^2 + \frac{1}{2\alpha}(\tilde{\Phi}_d^T W_d)^2 + \frac{\alpha}{2}e_d^2 + \frac{1}{2\alpha}(\tilde{\Phi}_v^T W_v)^2 + \frac{\alpha}{2}e_v^2 - K_\omega s^2 - K_d e_d^2 - K_v e_v^2$$

Setting,

$$K_\omega = K_d = K_v = \frac{\alpha}{2} + \beta$$

where β is positive gain yields,

$$\dot{V} \leq \frac{1}{2\alpha}\Psi - \beta\left(s^2 + e_d^2 + e_v^2\right)$$

with,

$$\Psi = (\tilde{\Phi}_q^T W_q)^2 + (\tilde{\Phi}_d^T W_d)^2 + (\tilde{\Phi}_v^T W_v)^2$$

So, it is possible to choose α and β so that $\dot{V} \leq 0$. Therefore, the system is stable in the sense of Lyapunov and converges to a small neighborhood of the origin, which is a region defined by the neural network approximation errors $\tilde{\Phi}$. This region gets smaller as $\tilde{\Phi} \to 0$. Theoretically, it can be made arbitrarily small by increasing the gains.

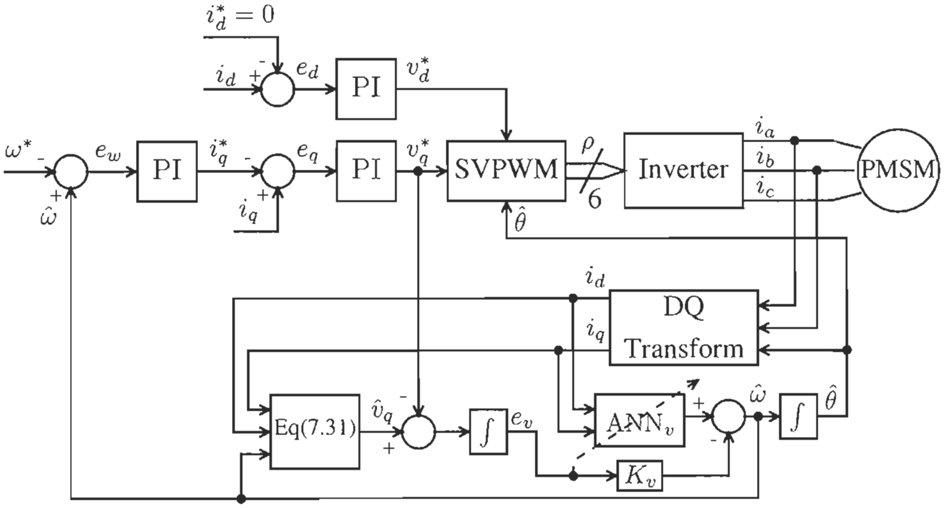


FIGURE 7.23 – Sensorless vector control scheme

7.7.1 Results

The controller is compared in similar operating conditions to the well-known sensorless vector control technique shown in Fig. 7.23, which is widely used in the industry. It is noteworthy that for the sake of a fair comparison, there is no decoupling terms between d-q axes as in few vector control strategies since the proposed ANN control structure is not based on any decoupling term. On the other hand, the observer from Fig. 7.22 is being used. The desired rotor speed is taken as the step response of a critically damped second order system with a natural frequency of 2 rad/s, as shown in Fig. 7.24.

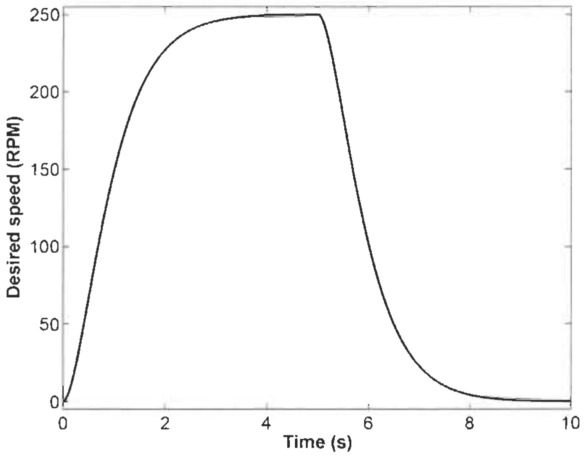


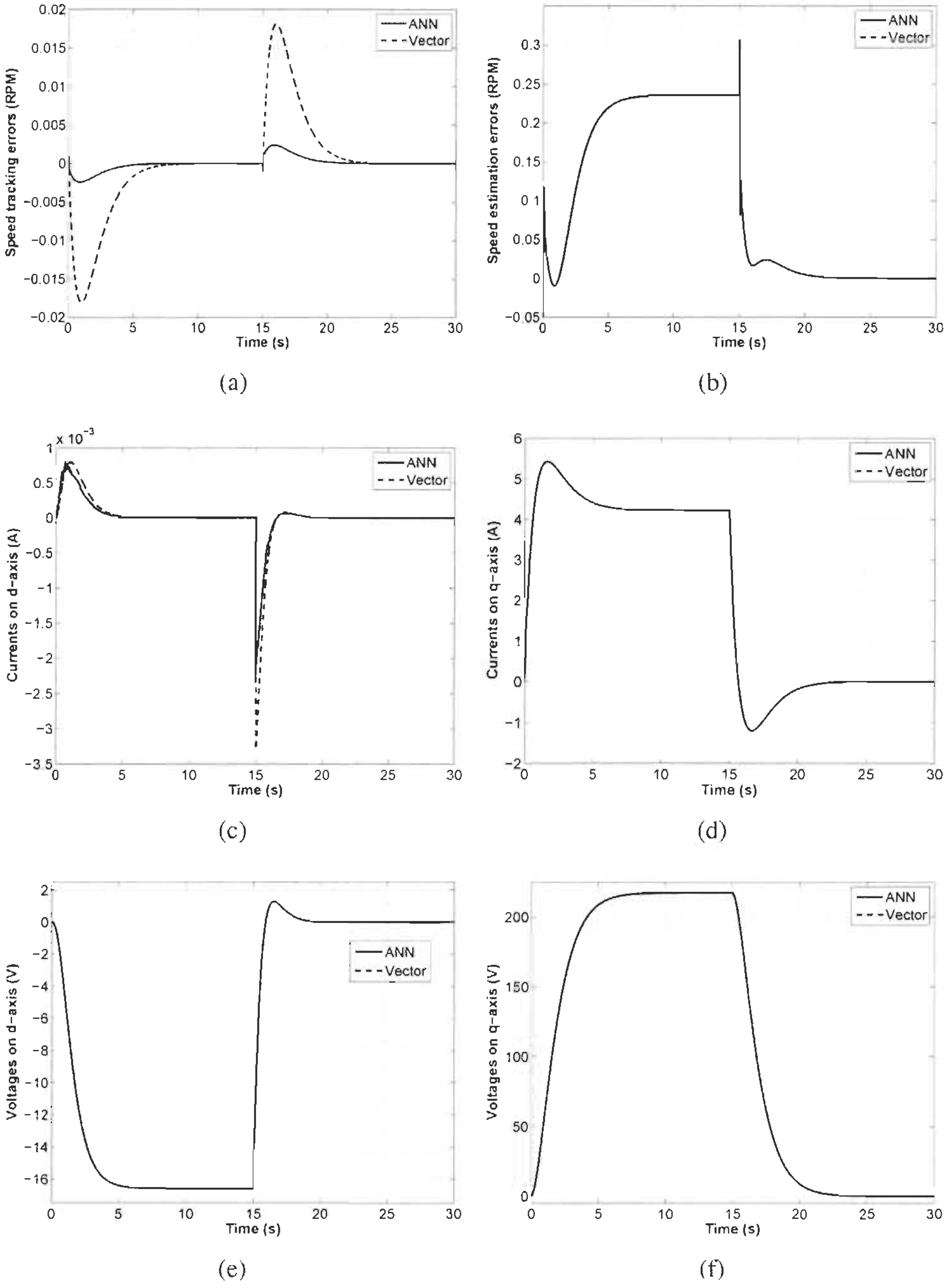
FIGURE 7.24 – Nonlinear observer speed reference signal

Five simulation runs are carried out to study the controller's performance. For each simulation, the ANN's initial weights are set to zero and the system's response is studied taking into account the machine's tracking and estimation speed errors, and the d-q axes currents and voltages.

In the first simulation, the nominal values (Table 7.1) are used to simulate the machine's dynamics. As shown in Fig. 7.25, the ANN control scheme outperforms the vector controller by its high tracking precision, i.e., the speed error amplitude is kept in within sensors resolution. It is noteworthy that this performance is achieved by a control effort similar to the one of the vector control strategy (Fig. 7.25(e) and Fig. 7.25(f)). On the other hand, a satisfactory trajectory tracking is obtained with the vector controller while both controllers provide smooth d-q axes currents and voltages. Moreover, the observer delivers good performance during fast changes in the electromagnetic torque in both cases. Although the ANN-based vector control did not use decoupling terms, it can be observed on Fig. 7.25 that it resulted into reduced coupling among the speed control loop and the flux control loop (I_d response) as compared to the vector control.

Since the machine's parameters are time-varying, three simulation sets are carried out to study the proposed adaptive controller's ability to sustain various intensities of parametric uncertainties. First, the machine's mechanical inertia parameter J is decreased by 50%, the results are depicted in Fig. 7.26. Next, the machine's electrical parameter L_d is decreased by 50%. The performance of both controllers is shown in Fig. 7.27. Finally, the machine's electrical parameter L_q is decreased by 75% with respect to its nominal value (table 7.1). The performance under low inductance is shown in Fig. 7.28. The ANN controller deals successfully with the changes by keeping similar performance as with nominal parameters. However, oscillations are observed with the vector controller. On the other hand, although q axes current is not explicitly controlled by a current loop regulator as in the vector control technique, the proposed controller was able to provide smooth control signal, which yields smooth q-axis current. Moreover, the rotor speed follows the desired speed closely. The superiority of the ANN controller over conventional control techniques is clearly shown in these simulation runs.

The next simulation is meant to show the modularity of the proposed control approach in motoring and regenerating modes. For that, the first simulation is repeated with a direction change in the speed reference signal, the results are shown in Fig. 7.29. As it can be seen, the change in the speed reference signal has little effect on the control performance. It maintains a similar behavior as in the first simulation (Fig. 7.25), the estimated rotor speed follows the actual speed smoothly in the whole speed range in both motoring and regenerating modes.



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 7.25 – ANN-based vector control response with nominal values : (a) speed errors ; (b) speed estimation errors ; (c) d-axis currents ; (d) q-axis currents ; (e) d-axis voltages ; and (f) q-axis voltages.

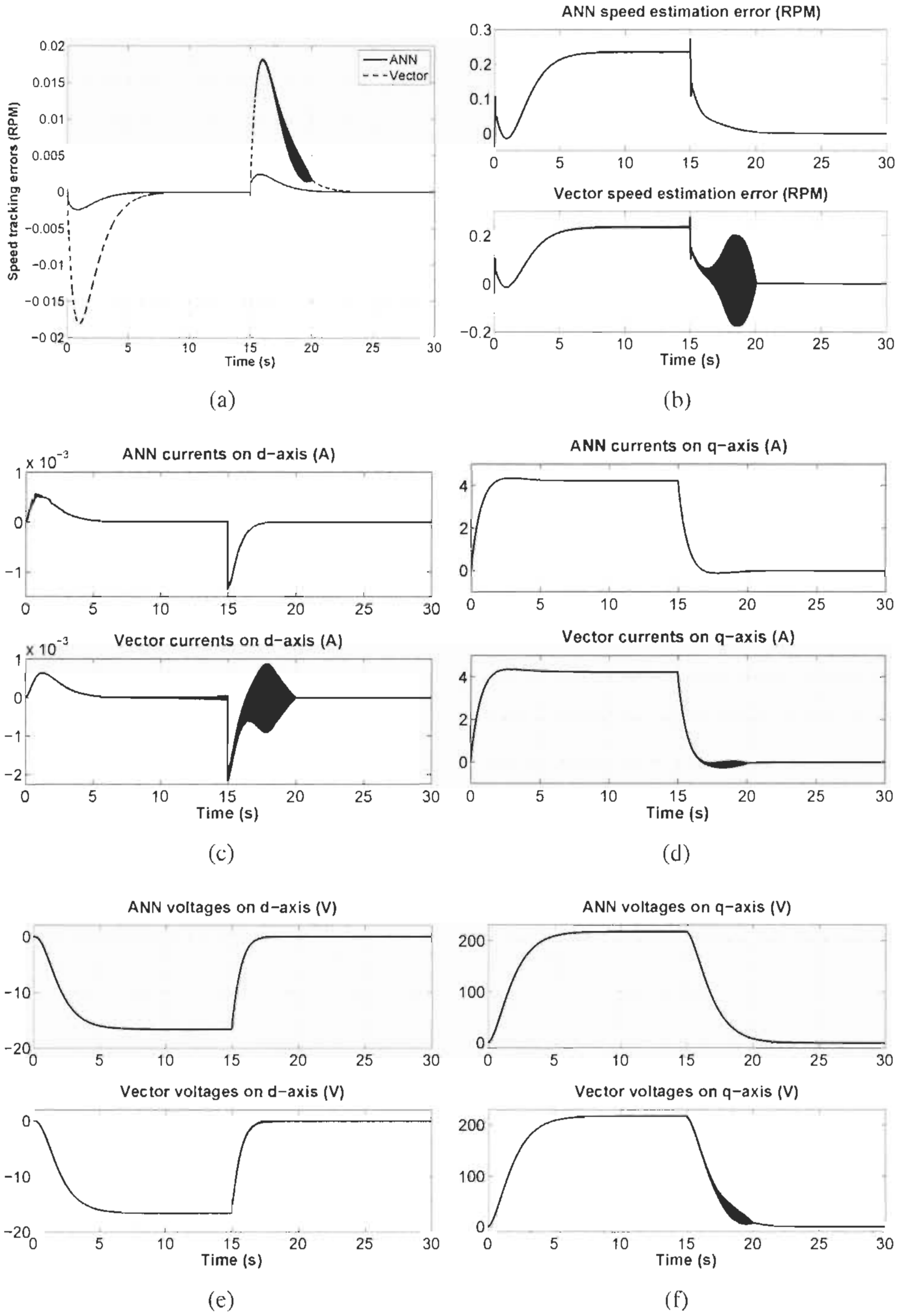
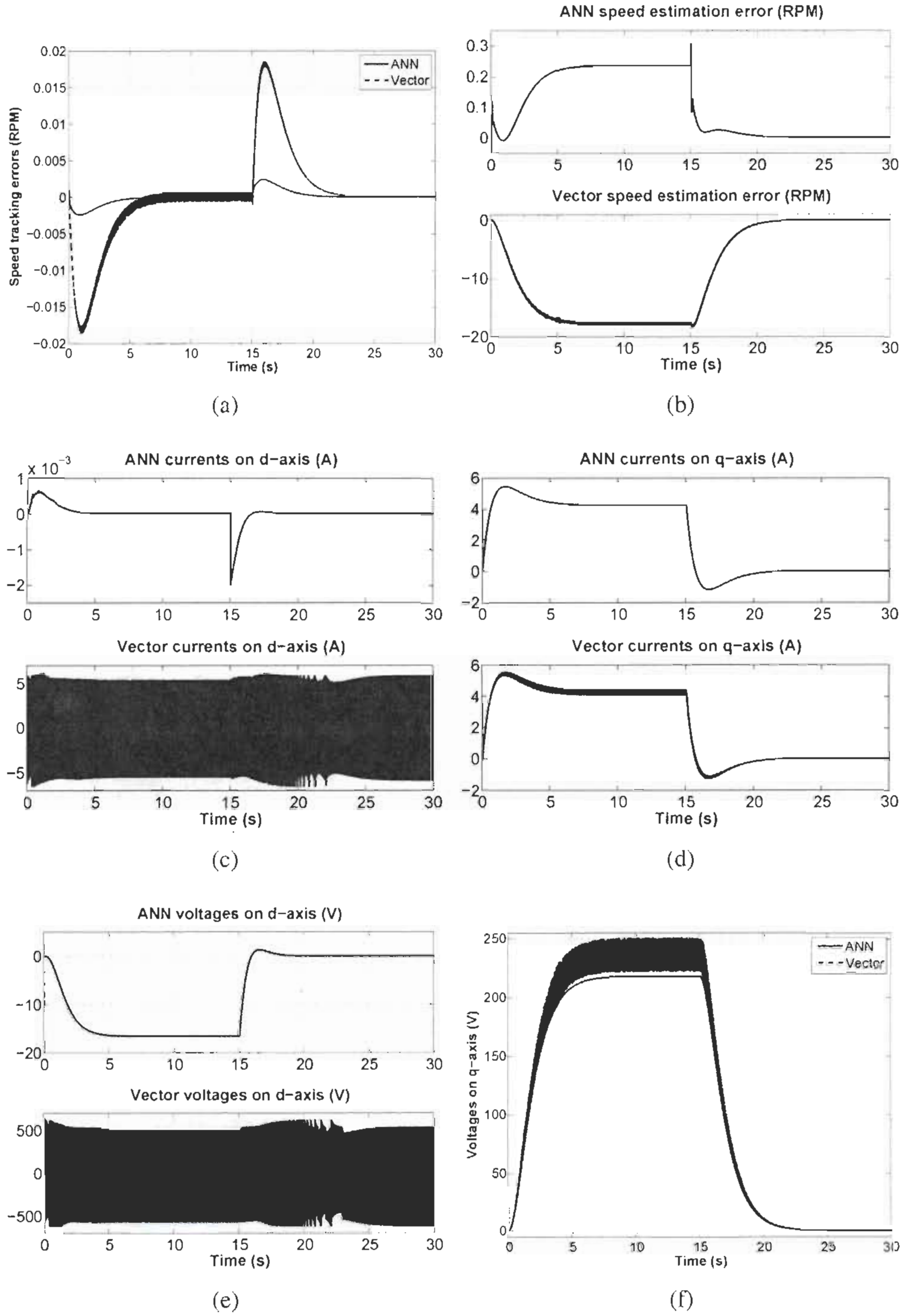


FIGURE 7.26 – ANN-based vector control response with low inertia J : (a) speed errors ; (b) speed estimation errors ; (c) d-axis currents ; (d) q-axis currents ; (e) d-axis voltages ; and (f) q-axis voltages.



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 7.27 – ANN-based vector control response with low inductance L_d : (a) speed errors ; (b) speed estimation errors ; (c) d-axis currents ; (d) q-axis currents ; (e) d-axis voltages ; and (f) q-axis voltages.

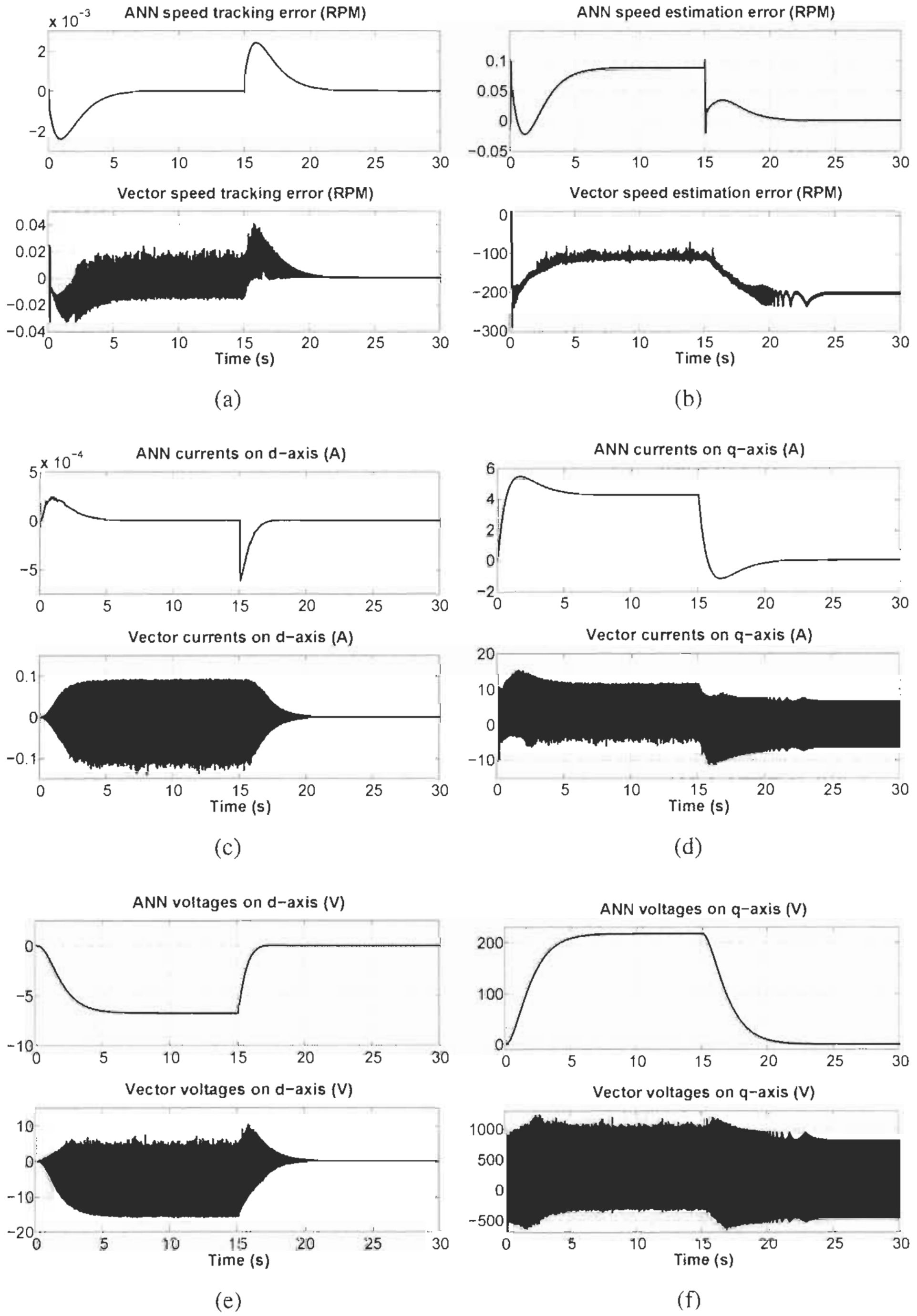
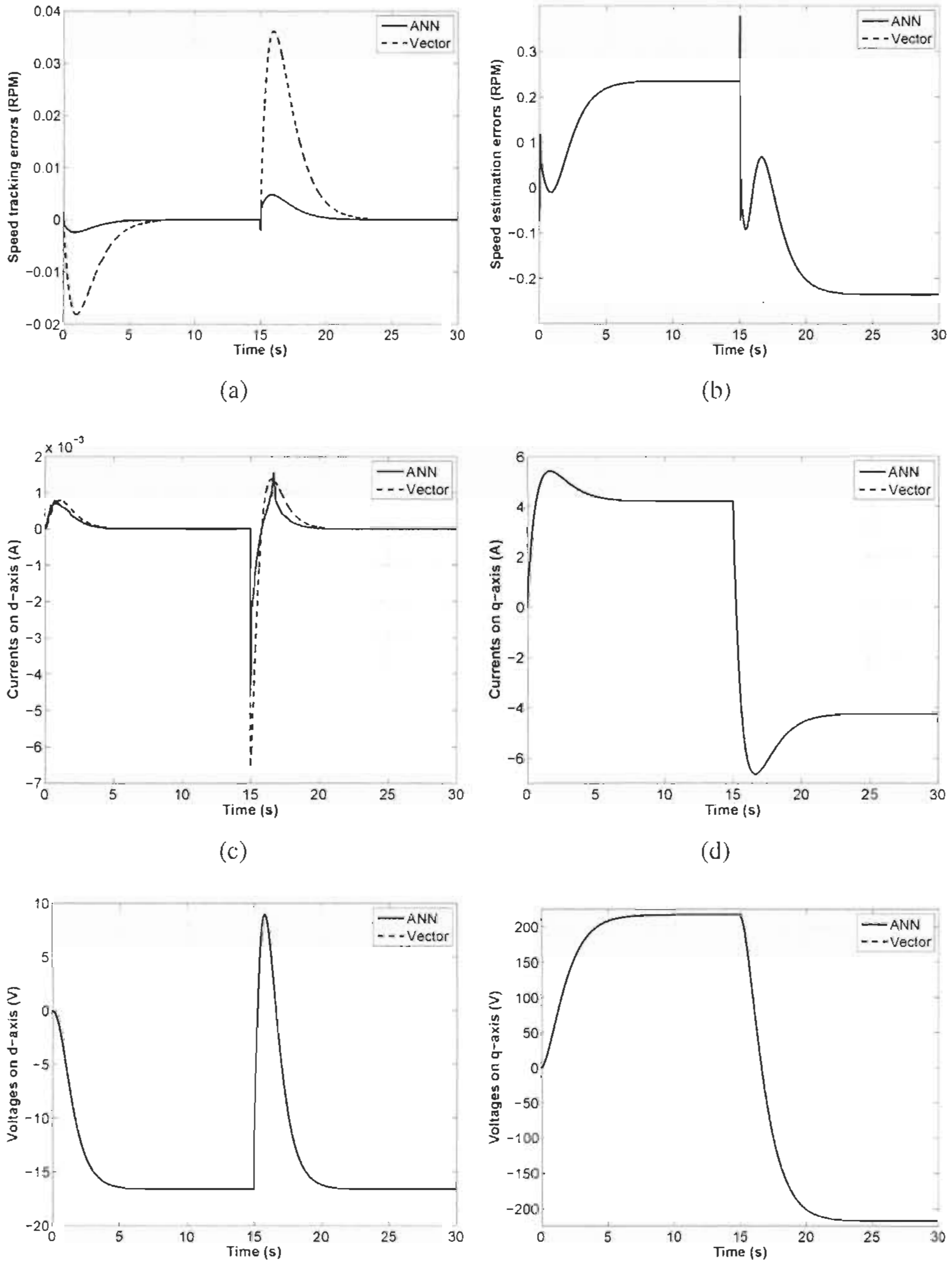


FIGURE 7.28 – ANN-based vector control response with low inductance L_q : (a) speed errors ; (b) speed estimation errors ; (c) d-axis currents ; (d) q-axis currents ; (e) d-axis voltages ; and (f) q-axis voltages.



(a) (b)

(c) (d)

(e) (f)

FIGURE 7.29 – ANN-based vector control response in motor and generator mode : (a) speed errors ; (b) speed estimation errors ; (c) d-axis currents ; (d) q-axis currents ; (e) d-axis voltages ; and (f) q-axis voltages.

7.8 Robust ANN-Based Nonlinear Speed Observer

A robust ANN-based nonlinear speed observer is proposed without any *a priori* dynamics knowledge. Thus, the proposed observer is able to cope with higher degrees of nonlinearity since it is not based on a linear-in-parameters model, unlike many neural network observers. Moreover, parameters knowledge is not required, which solves the parameter sensitivity problem. Furthermore, the measurement of the noisy motor voltage is also not required, which favors better convergence and tracking and reduces the number of sensors with respect to other methods.

The field-oriented vector control approach [71] is used in this work to highlight the performance of the proposed observer as shown in Fig. 7.30. The speed controller is based on a proportional integral (PI) controller that achieves tracking by taking the error between the reference and estimated velocities to deliver a desired quadrature current signal i_q^*. Two other PI controllers are used for d-q axes currents regulation. The proposed method uses i_a, i_b, and θ as system's measurable states and the system parameters are assumed to be *a priori* unknown.

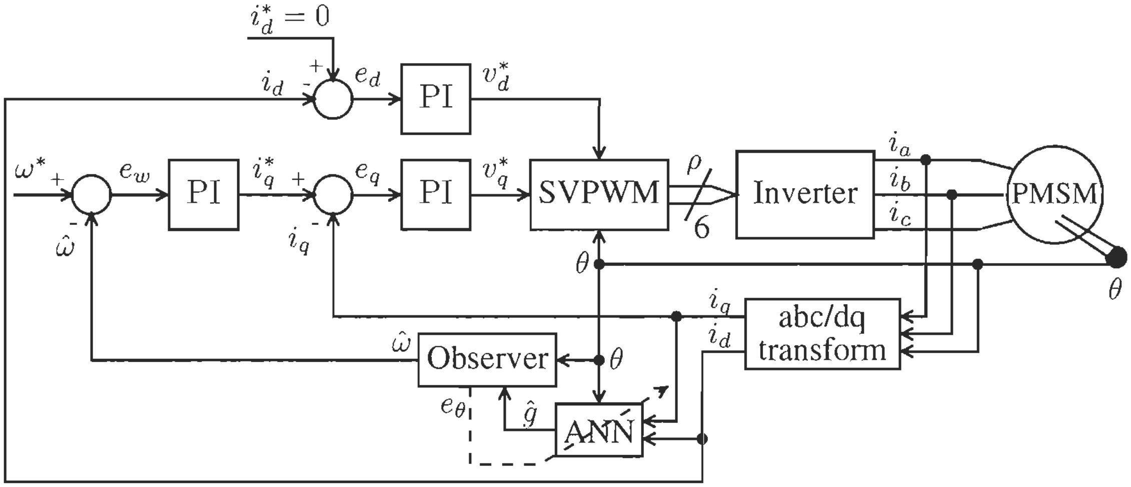


FIGURE 7.30 – Robust ANN-based nonlinear speed observer scheme

The system dynamics (7.1) and (7.2) can be expressed as a general nonlinear multiple input multiple output (MIMO) system model,

$$\dot{x}(t) = f(x,u)$$
$$y(t) = Cx(t)$$

where, $u = [i_d, \; i_q]^T \in \mathbb{R}^{m_u}$ is the input, $y \in \mathbb{R}^{m_y}$ is the output, $x = [\theta, \; \omega]^T \in \mathbb{R}^n$ is the state vector, and f is an unknown nonlinear function.
Add and subtract $Ax(t)$,

$$\dot{x}(t) = Ax(t) + g(x,u)$$
$$y(t) = Cx(t)$$

where, $g(x,u) = f(x,u) - Ax(t)$ is the unknown nonlinear function to be approximated by the neural network. The matrix A is selected such as the pair (C, A) is observable. Therefore, the observer can be designed as,

$$\dot{\hat{x}}(t) = A\hat{x}(t) + \hat{g}(\hat{x},u) + G(y - C\hat{x}(t))$$
$$\hat{y}(t) = C\hat{x}(t)$$

where, $e_\theta = y - C\hat{x}(t)$ is the observer error, $\hat{x}$ is the observer state vector and $G \in \mathbb{R}^{n \times m_y}$ is the observer gain matrix. Therefore, G can be chosen to make the matrix $(A - GC)$ Hurwitz. Since the observer model is based on a linear technique, a neural network is designed to approximate the nonlinearities of the function $g(x,u)$ by minimizing the error e_θ. Therefore, the neural network output is expressed as,

$$\hat{g}(\hat{x},u) = g(x,u) + \delta$$

with, δ being the neural network approximation error. The neural network is composed of three layers each : one input layer of three neurons, one hidden layer with eight neurons and one neuron for the output layer. The sigmoid function is used as activation function for all neurons except for the output neurons, which use a linear function.

7.8.1 Results

Three numerical simulations are carried out to study the proposed observer's performance. For each simulation, the system's response is studied taking into account the machine's speed tracking and estimation errors, the currents error on d-q axes, and the control output voltages v_d and v_q. The desired rotor speed is shown in Fig. 7.24.

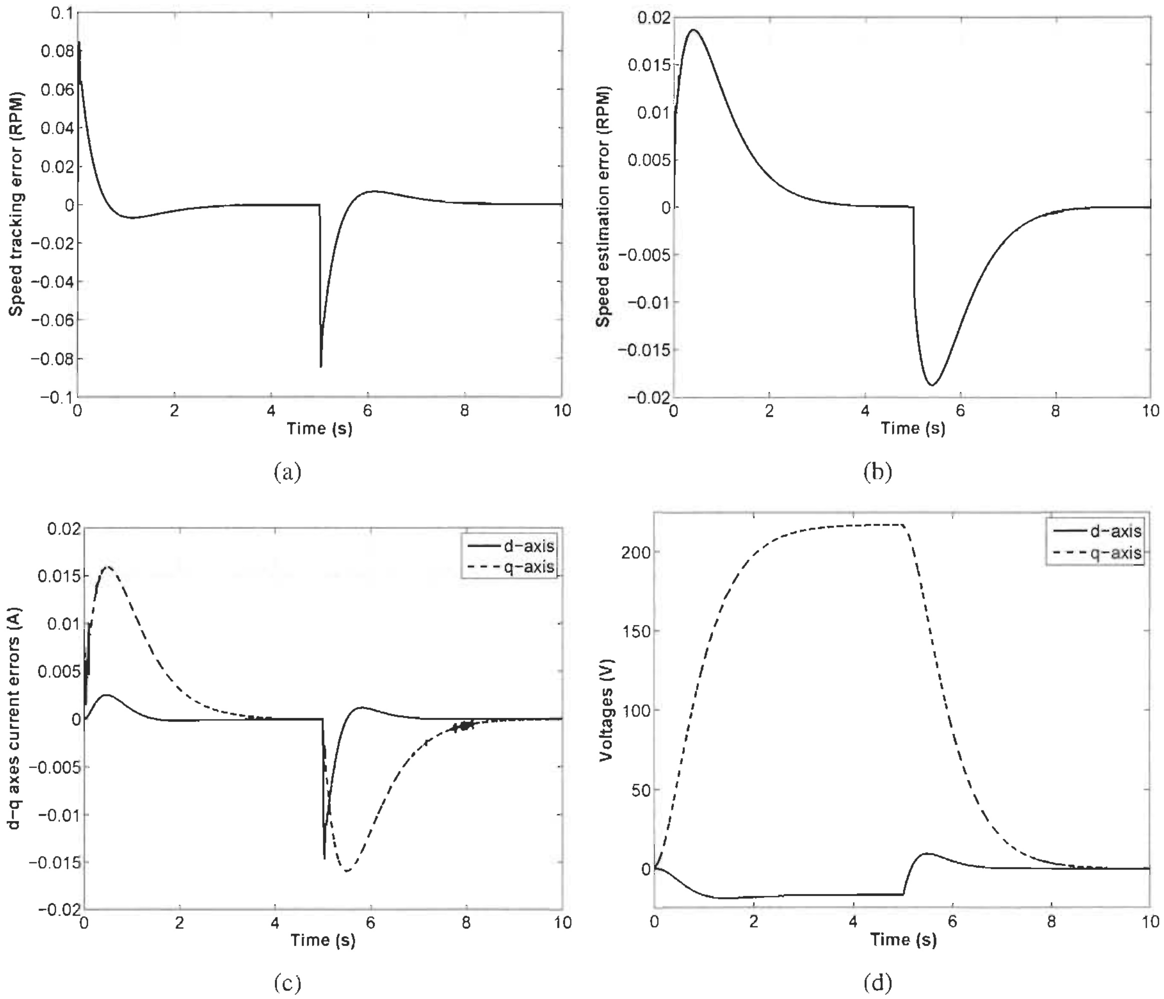
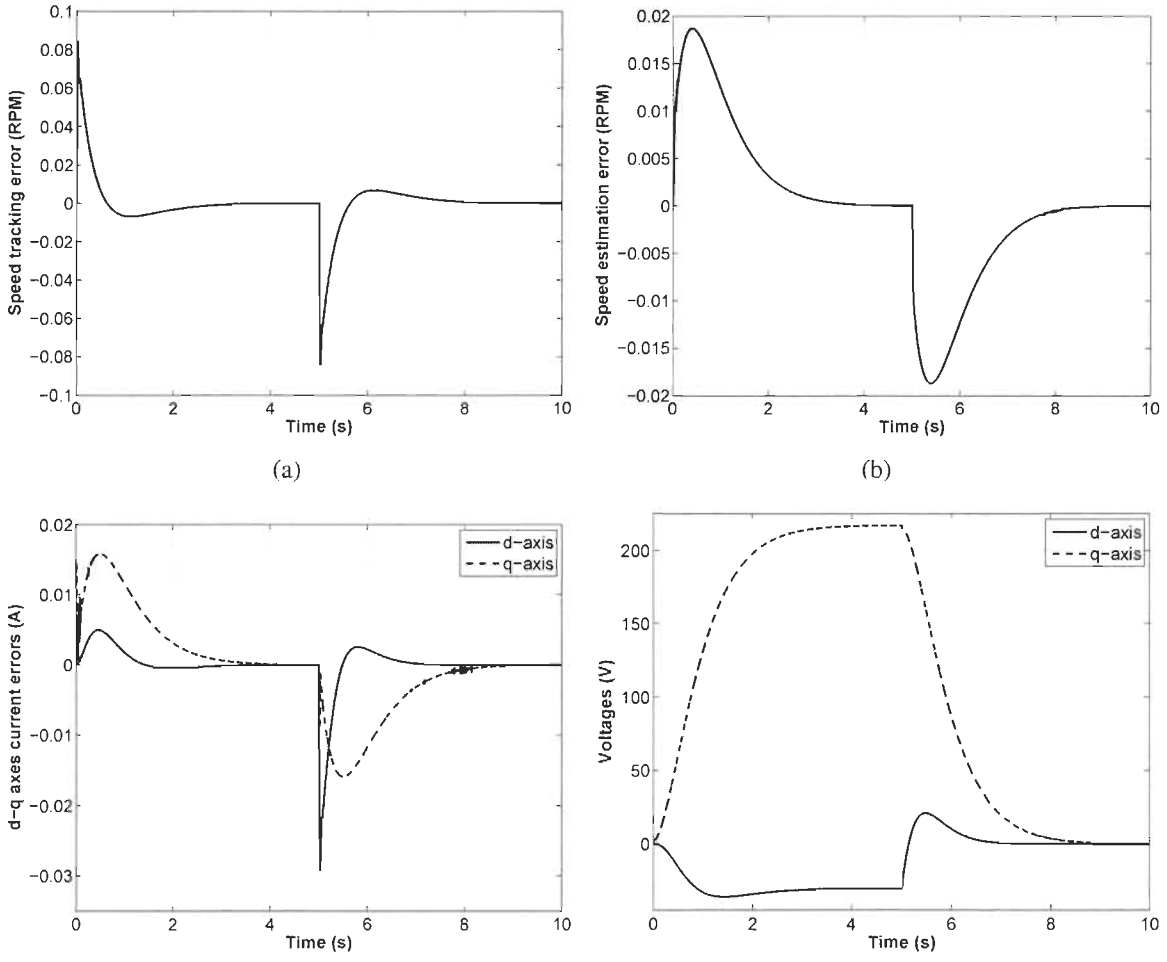


FIGURE 7.31 – System response with nominal values : (a) speed tracking error ; (b) speed estimation error ; (c) d-q axes current errors ; and (d) controller's output voltages v_d and v_q.

In the first simulation, the aforementioned nominal values are used to simulate the machine's dynamics. As shown in Fig. 7.31, the estimation position and speed errors decay gradually before stabilizing within a negligible amplitude. On the other hand, a satisfactory trajectory tracking is obtained with the proposed observer while the controller provides smooth d-q axes currents and torque signal. Moreover, the estimated rotor speed follows the actual speed closely during fast changes in the electromagnetic torque.



(a)

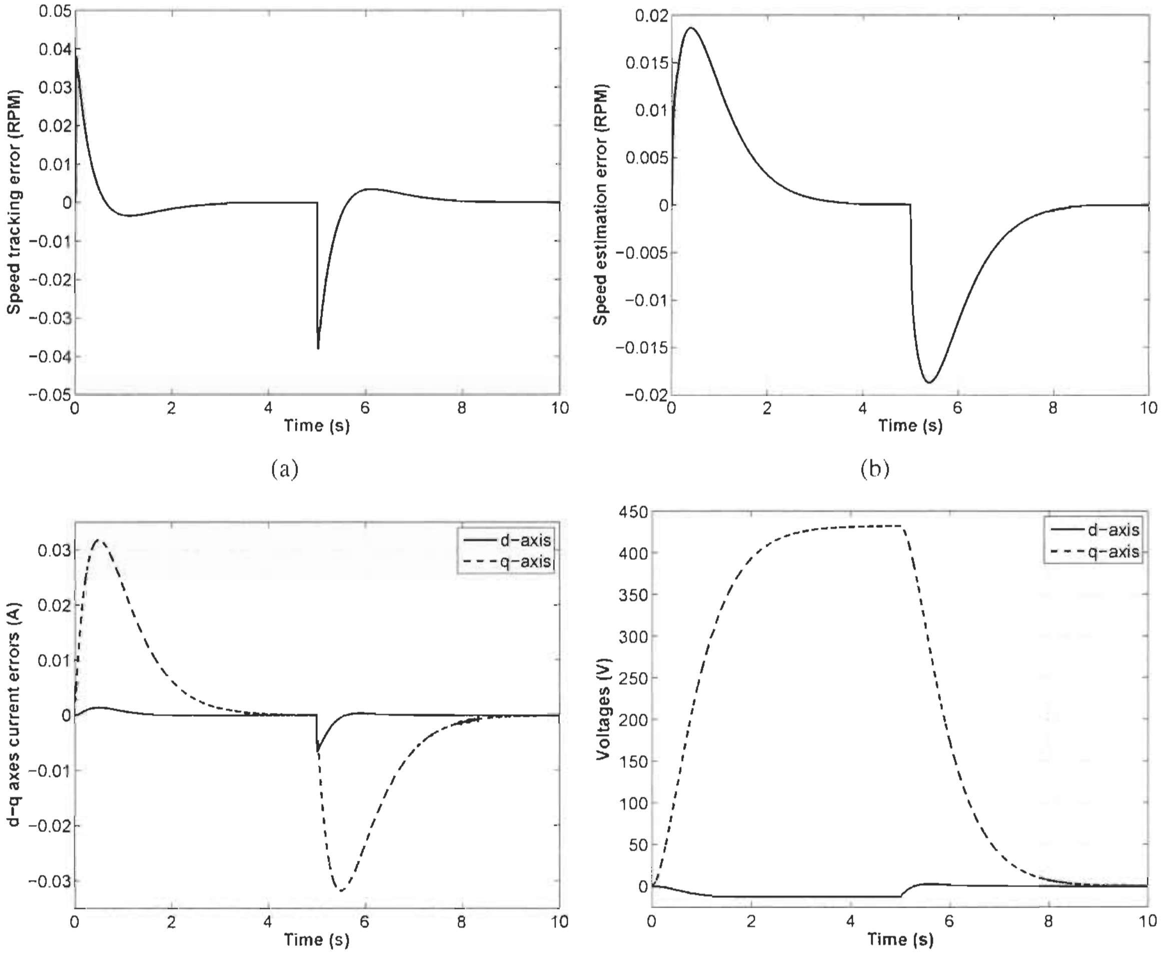
(b)

(c)

(d)

FIGURE 7.32 – System response with inductance variations ($L_q = L_q^{\text{nom}} \times 2$) : (a) speed tracking error ; (b) speed estimation error ; (c) d-q axes current errors ; and (d) controller's output voltages v_d and v_q.

Since the machine's electrical parameters are known to be time-varying, a simulation is car-



(a)

(b)

(c)

(d)

FIGURE 7.33 – System response with flux variations ($\lambda = \lambda^{\text{nom}} \times 2$) : (a) speed tracking error ; (b) speed estimation error ; (c) d-q axes current errors ; and (d) controller's output voltages v_d and v_q.

ried out to study the proposed observer's ability to sustain various parametric uncertainties. The formulation (7.1b) shows that the performance of observers based on a linear-in-parameters model is dependent on parameters R, L_d, L_q, λ, and p. The parameter p is the number of pairs of poles and is constant. Moreover, setting i_d^* to zero makes the observer less sensitive to L_d. It is noteworthy that from equation (7.1b), $p\ \lambda\ \omega$ is the most significant term at ω_{nominal}. Thus, linear-in-parameters model based observers offer poor estimation performance when parameter λ varies. This parameter used to be estimated online, which adds more complexity to existing estimation schemes. Therefore, the machine's electrical parameters, L_q, and λ are increased one at a time by 200% their nominal values. As it is shown in Fig. 7.32 and Fig. 7.33, parameters variation have no significant effect on the observer's performance. Unlike conventional estimation techniques, the proposed observer assumes *a priori* unknown dynamics and is able to cope with higher degrees of uncertainties. The proposed observer achieves high estimation accuracy in the presence of high parametric uncertainties, which yielded smooth currents and control signals. This shows the modularity of the proposed observer as it maintains a similar behavior as in nominal parameters case.

7.9 Adaptive Fuzzy Logic Control

The aim of the proposed control strategy is to achieve robustness to both structured and unstructured uncertainties with a single controller, which reduces the system's complexity compared to classical cascaded-based control structures. A Lyapunov stability-based adaptation technique is used as an alternative to the conventional heuristic tuning methods. Thus, the stability of the proposed approach is guaranteed unlike many computational intelligence-based controllers. Moreover, the proposed adaptive fuzzy logic controller (FLC) uses a computational efficient membership functions and operators to alleviate the computational burden associated with soft-computing tools, which makes it practically realizable. Furthermore, the measurement of the noisy machine voltages and currents is not required as in most of the PMSM-based controllers, which favors better convergence and tracking and reduces the number of sensors, with respect to other methods. To the authors' best knowledge, this work is one of the first attempts, if any, to achieve high tracking control performance in the presence of both structured and unstructured uncertainties by adaptive fuzzy logic control architecture without the use of currents sensing or loop regulation for PMSMs.

The control strategy uses quadrature voltage v_q to achieve speed tracking and keeps the direct voltage v_d constant to zero. The resultant control scheme is illustrated in Fig. 7.34. As such, the adaptive fuzzy controller achieves a precise speed tracking by machine's inverse dynamics approximation.

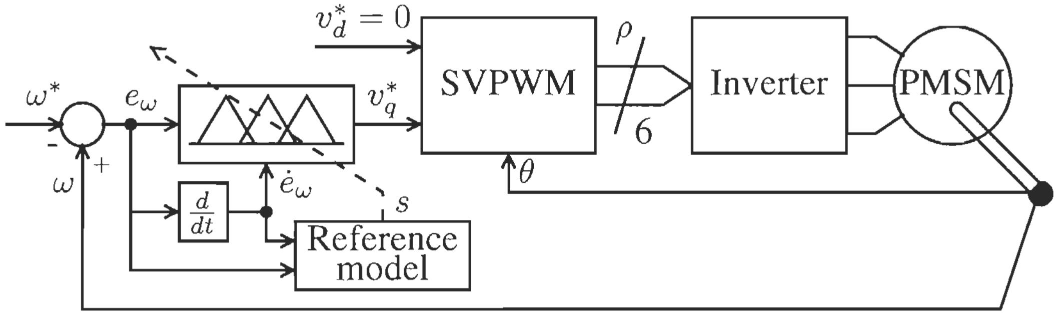


FIGURE 7.34 – Adaptive FLC control scheme

The fuzzy control strategy is based on a human operator experience to interpret a situation and initiate its control action. Given the speed signals ω^* and ω, the speed error e_w is then computed. The FLC takes e_w and its derivative $\dot{e}_w$ and provides a control action v_q. These signals are quantized into 7 levels represented by a set of linguistic variables : Negative Large (NL), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), and Positive Large (PL). In this study, triangular membership functions are used, mainly due to their high computational and performance efficiencies. The input membership functions and the fuzzy rules adopted by the fuzzy control system are shown in Fig. 7.35 and Table 7.2, respectively. They have been accredited for their computational efficiency and satisfactory performance and can be refined by an expert to get better control performance. However, these procedures are conducted ahead of time and are not part of the control cycle. The adopted rules are inspired from the rules described in Table 5.2. This way, the FLC forces the machine's speed error e_ω and its derivative $\dot{e}_\omega$ to approach zero. It is worth mentioning that an empirical study was conducted beforehand to tune the input membership functions. The center of area method is used for defuzzification.

TABLE 7.2 – Fuzzy logic rules for PMSMs

	e_w						
$\dot{e}_w$	NL	NM	NS	Z	PS	PM	PL
PL	Z	PL	PL	PL	PL	PL	PL
PM	NS	Z	PS	PS	PL	PL	PL
PS	NL	NS	Z	PS	PS	PL	PL
Z	NL	NL	NS	Z	PS	PL	PL
NS	NL	NL	NS	NS	Z	PS	PL
NM	NL	NL	NL	NS	NS	Z	PS
NL	NL	NL	NL	NL	NL	PL	Z

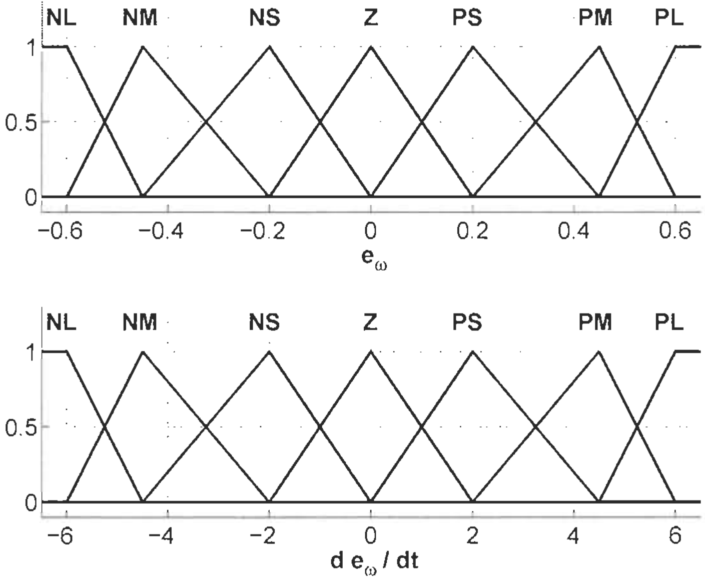


FIGURE 7.35 – Fuzzy logic membership functions

The adaptive FLC structure is depicted in Fig. 7.36. It consists of four layers. Input nodes and fuzzification nodes are shown in layer 1 and layer 2, respectively, forming the antecedent part of the fuzzy rules. Consequent parts are represented by layer 3 and 4 which are constructed with fuzzy rule nodes and output nodes. They are linked by the weight matrix $\hat{W}$, which is tuned online using a Lyapunov-based adaptation technique.

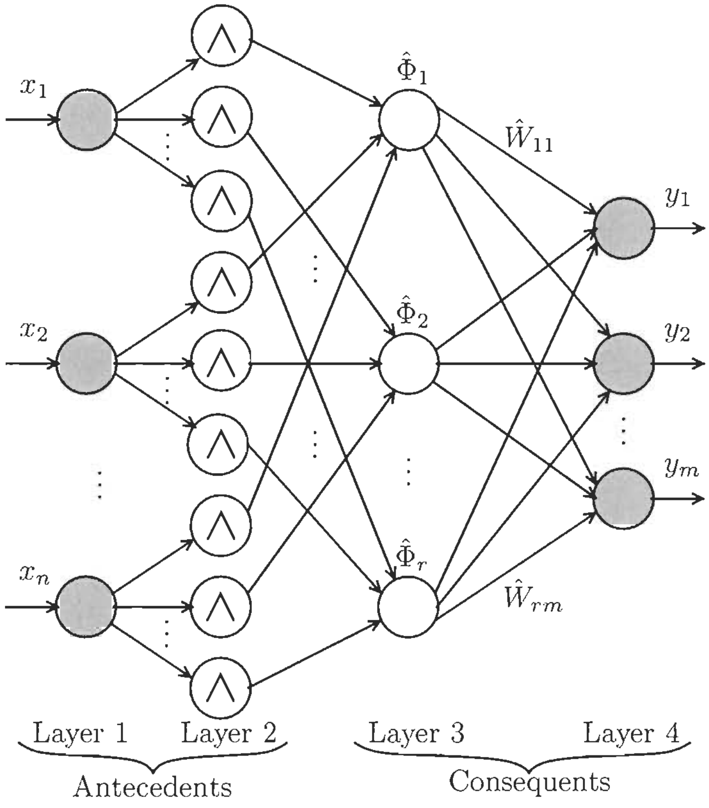


FIGURE 7.36 – Adaptive fuzzy logic control structure

The FLC's output can be written as :

$$Y = \Phi^T W + \varepsilon = \hat{\Phi}^T \hat{W} \tag{7.57}$$

where, $\varepsilon = \hat{\Phi}^T \hat{W} - \Phi^T W$ is the fuzzy logic output error, $\hat{W} \in \mathbb{R}^{r \times m}$ is the fuzzy logic consequent part weight matrix and $\hat{\Phi} \in \mathbb{R}^r$ is the r-dimensional fuzzy logic antecedent part vector of known functions (regressor) defined as :

$$\hat{\Phi} = \frac{\mu_{B^p}(y)}{\sum_{p=1}^{r} \mu_{B^p}(y)}$$

Since in this work the FLC is a single output fuzzy logic system, i.e., ($m = 1$), the matrix

$\hat{W} \in \mathbb{R}^{r \times m}$ is reduced to a vector $\hat{W} \in \mathbb{R}^{r \times 1} = [y_1, y_2, \ldots, y_r]$, where y_p is the fuzzy logic consequent part of the p^{th} rule, $p = 1, \ldots, r$. The sign $\hat{\bullet}$ denotes the parameter estimate. Define the reference model signal s as in (7.46).

Recall the formulation (7.14),

$$v_q = \mu\, \dot{\omega} - \frac{R}{\lambda}\, (L_d - L_q)\, i_d\, i_q + L_q\, \frac{d}{dt} i_q + L_d\, p\, \omega\, i_d + p\, \lambda\, \omega + \frac{\mu}{J}\, (\tau_F + \tau_L)$$

The desired dynamics of this model can be represented by the following linear regression, which is a well established representation in adaptive control theory and commonly used in adaptive control strategies [43, 52] :

$$\mu \dot{\omega}_t - \frac{R}{\lambda}(L_d - L_q) i_d i_q + L_q \frac{d}{dt} i_q + L_d p \omega i_d + p\lambda\omega + \frac{\mu}{J}\tau_F + \frac{\mu}{J}\tau_L = \Phi^T W \tag{7.58}$$

In this work, we set $r = 7$, making the dimension of vectors $\hat{\Phi}$ and $\hat{W}$ equal to dimension of vectors Φ and W in (7.58). Therefore, the control law can be defined as :

$$v_q = \hat{\Phi}^T \hat{W} - K_D s \tag{7.59}$$

with $K_D = \frac{\alpha}{2} + \beta$, where α and β are positive gains.

Theorem 17 *Consider a nonlinear system in the form (7.1)-(7.2) with the control law (7.59). The closed-loop system's stability is achieved with the following adaptation law :*

$$\dot{\hat{W}} = -\Gamma\, \hat{\Phi}\, s$$

where $\Gamma = diag(\gamma_1, \gamma_2, \ldots, \gamma_r)$ and γ_l is a positive constant, $l = 1, \ldots, r$.

Proof 11 *Take the derivative of the signal s in (7.46) :*

$$\dot{s} = \dot{\omega} - \dot{\omega}_t$$

Multiply both sides by μ *:*

$$\mu \dot{s} = \mu \dot{\omega} - \mu \dot{\omega}_t$$

Substitute $\mu \dot{\omega}$ *from (7.14) and use the linear regression (7.58) :*

$$\mu \dot{s} = v_q - \Phi^T W \tag{7.60}$$

Choose the following Lyapunov candidate :

$$V = \mu s^2 + \frac{1}{2}\{\tilde{W}^T \Gamma^{-1} \tilde{W}\}$$

Taking the derivative of V *:*

$$\dot{V} = \mu \dot{s} s + \tilde{W}^T \Gamma^{-1} \dot{\hat{W}}$$

Substitute $\mu \dot{s}$ *from (7.60) :*

$$\dot{V} = \{v_q - \Phi^T W\} s + \tilde{W}^T \Gamma^{-1} \dot{\hat{W}}$$

Setting the control law as

$$v_q = \hat{\Phi}^T \hat{W} - K_D s$$

leads to

$$\dot{V} = \varepsilon s - K_D s^2 + \tilde{W}^T \Gamma^{-1} \dot{\hat{W}}$$

where, $\varepsilon = \Phi^T W - \hat{\Phi}^T \hat{W}$ *from (7.57). Substitute* ε *from (6.26) :*

$$\dot{V} = \hat{\Phi}^T \tilde{W} s + \tilde{\Phi}^T W s - K_D s^2 + \tilde{W}^T \Gamma^{-1} \dot{\hat{W}}$$

Setting the adaptation law as

$$\dot{\hat{W}} = -\Gamma \, \hat{\Phi} \, s$$

implies that

$$\dot{V} = \tilde{\Phi}^T W s - K_D s^2$$

Recall Young's inequality [80],

$$2ab \leq \frac{1}{\alpha} a^2 + \alpha b^2 \qquad \forall a, b \in \mathbb{R} \text{ and } \forall \alpha > 0$$

Therefore,

$$\dot{V} \leq \frac{1}{2\alpha}(\tilde{\Phi}^T W)^2 + \frac{\alpha}{2} s^2 - K_D s^2$$

Set $K_D = \frac{\alpha}{2} + \beta$ *yields,*

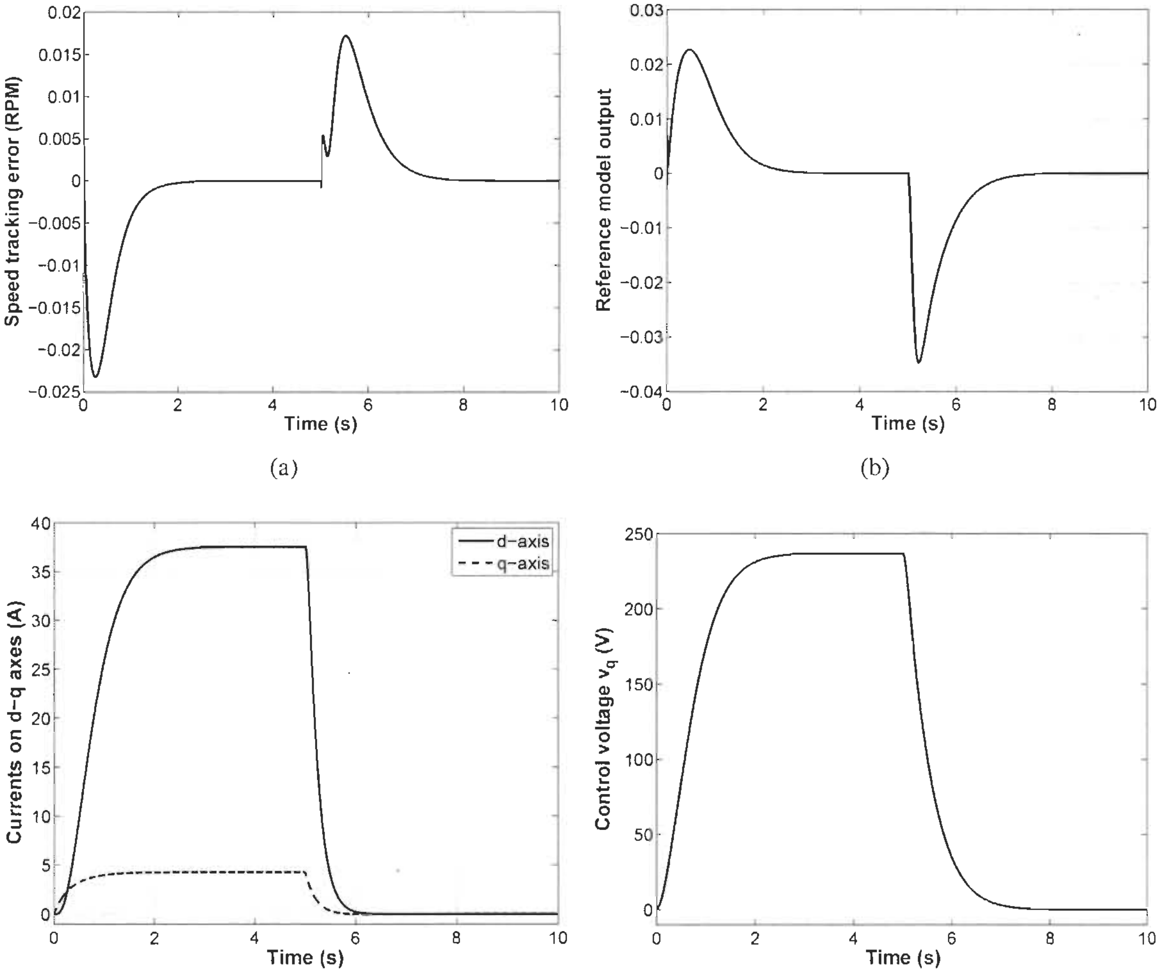
$$\dot{V} \leq \frac{1}{2\alpha}(\tilde{\Phi}^T W)^2 - \beta s^2$$

It is possible to choose α *and* β *so that* $\dot{V} \leq 0$*, except possibly in a neighborhood of* $s = 0$*. Then, the system is stable in the sense of Lyapunov. The neighborhood of* $s = 0$ *is a region defined by the fuzzy logic approximation error* $\tilde{\Phi}$ *and gets smaller as* $\tilde{\Phi} \to 0$*.*

7.9.1 Results

Five simulation runs are carried out to study the proposed controller's performance. For each simulation set, the system's response is studied taking into account the machine's speed tracking error, the currents on d-q axes, the reference model output s, and the control signal, i.e., applied voltage v_q^*. The desired rotor speed is shown in Fig. 7.24.

The aforementioned nominal values are used to simulate the machine's dynamics. The advantage behind the use of the adaptive fuzzy controller is clearly shown in (Fig. 7.37) by very good tracking performance and negligible amplitude where the speed error is kept in, i.e, 0.01%. At $t = 5$ s, the system is subjected to a trajectory change, which introduces a sudden brief fluctuation on the voltage v_q. This causes a dip on the speed tracking error, which is of a negligible magnitude. A compromise between trajectory tracking and disturbance rejection is achieved by adjusting ψ in (7.46). On the other hand, although d-q axes currents are not explicitly controlled by current loop regulators as in many conventional control techniques, the proposed controller was able to provide smooth control signal, which yields smooth d-q axes currents. Moreover, the rotor speed follows the desired speed closely during fast changes in the electromagnetic torque. It is noteworthy that the adaptive controller copes easily with nonlinear friction uncertainties and achieves fast and precise convergence and tracking.



(a)

(b)

(c)

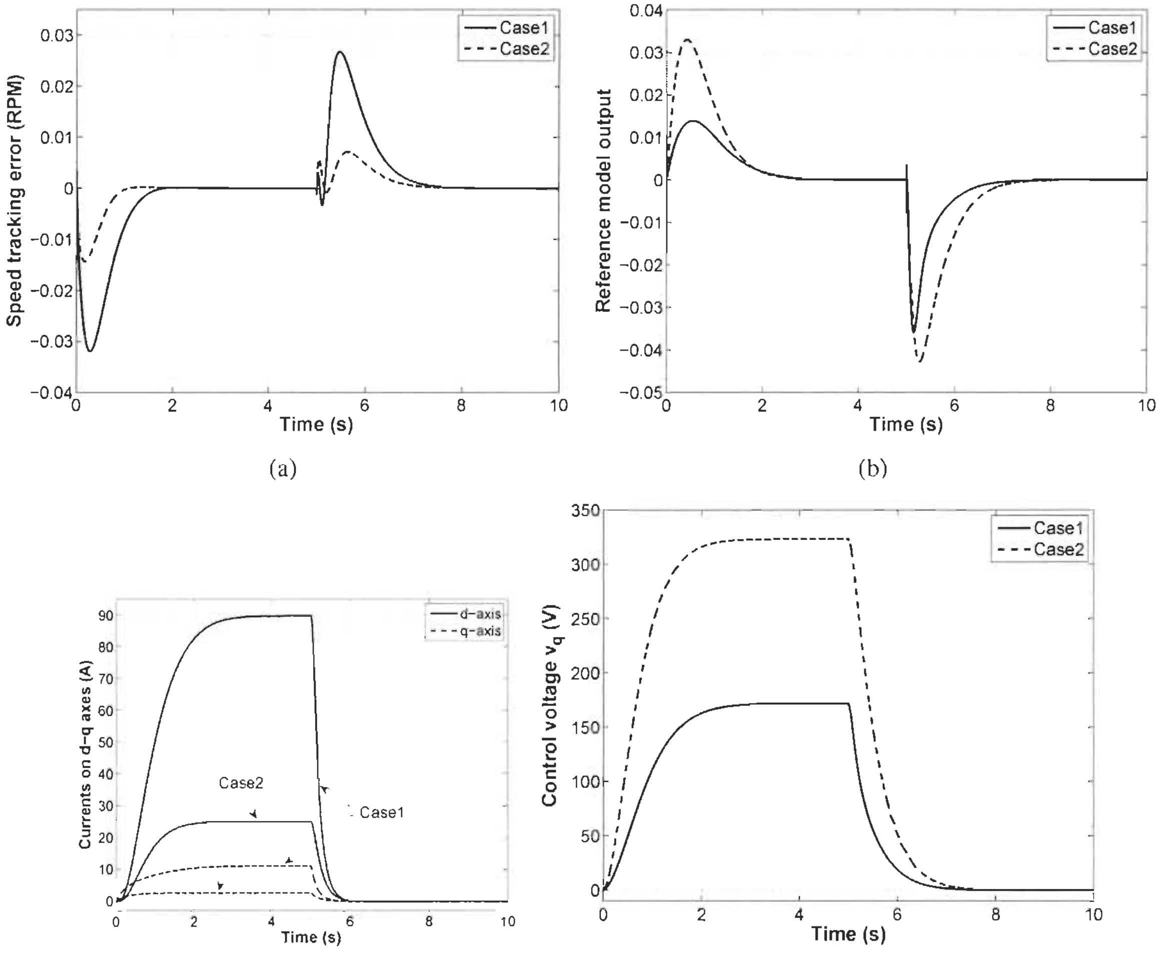
(d)

FIGURE 7.37 – Adaptive FLC response with nominal values : (a) velocity error ; (b) reference model output ; (c) d-q axes currents ; and (d) applied voltage v_q^*.

Since the machine's electrical parameters are time-varying, i.e., stator resistance is known to vary with temperature and inductance varies with current, two simulation sets are carried out to study the proposed adaptive controller's ability to sustain various intensities of parametric uncertainties. First, the machine's electrical parameters, R, L_d, L_q, and λ are decreased by 50%. The adaptive fuzzy controller's performance is shown in Fig. 7.38 under case1 label. The controller provides less control effort than nominal case due to parameters decrease. Moreover, it dealt successfully with the change with a slight increase in speed error, which yields smooth currents and control signal. In the next simulation, the machine's electrical parameters are amplified to 150% with respect to their nominal values (table 7.1). The adaptive fuzzy controller's performance to parameters increase is shown in Fig. 7.38 under case2 label. When the drive is subjected to parameters increase, it provides nearly twice control effort than nominal case. Consequently, the controller was successful in coping with parametric uncertainties by keeping the speed tracking error within negligible amplitude, which yielded smooth currents and control signal.

In the next simulation, a 10 (N·m) load torque step is introduced at time $t = 3$ s to evaluate the adaptive controller's performance to external disturbance. As shown in Fig. 7.39, the controller copes easily with the sudden change on the load torque and provides a fast velocity tracking response. Moreover, the velocity error remains small and without overshoot, which yielded smooth currents and control signals. The ability of the adaptive fuzzy controller in compensating for disturbance is clearly shown in this simulation.

The next simulation is meant to show the modularity of the proposed controller in compensating for friction nonlinearities. For that, the nonlinear Coulomb friction term is magnified 5 times and the machine operates in motoring and regenerating modes to enable zero velocity crossing (i.e., Coulomb friction effect) as shown in Fig. 5.3. The results are shown in Fig. 7.40. As it can be seen, the change in the speed reference signal has no effect on the overall control performance. The speed tracking error starts increasing during fast accelerations but stays in an acceptable range. Conventional control techniques tend to overcompensate for these effects and lead to severe tracking errors, limit cycles, chattering, excessive noise, and even instability [17]. As shown in Fig. 7.40, the nonlinearities around zero speed cause a tracking error. However, the controller is able to compensate for the nonlinear friction and no oscillation is observed, which yielded accurate speed tracking in both motoring and regenerating modes.



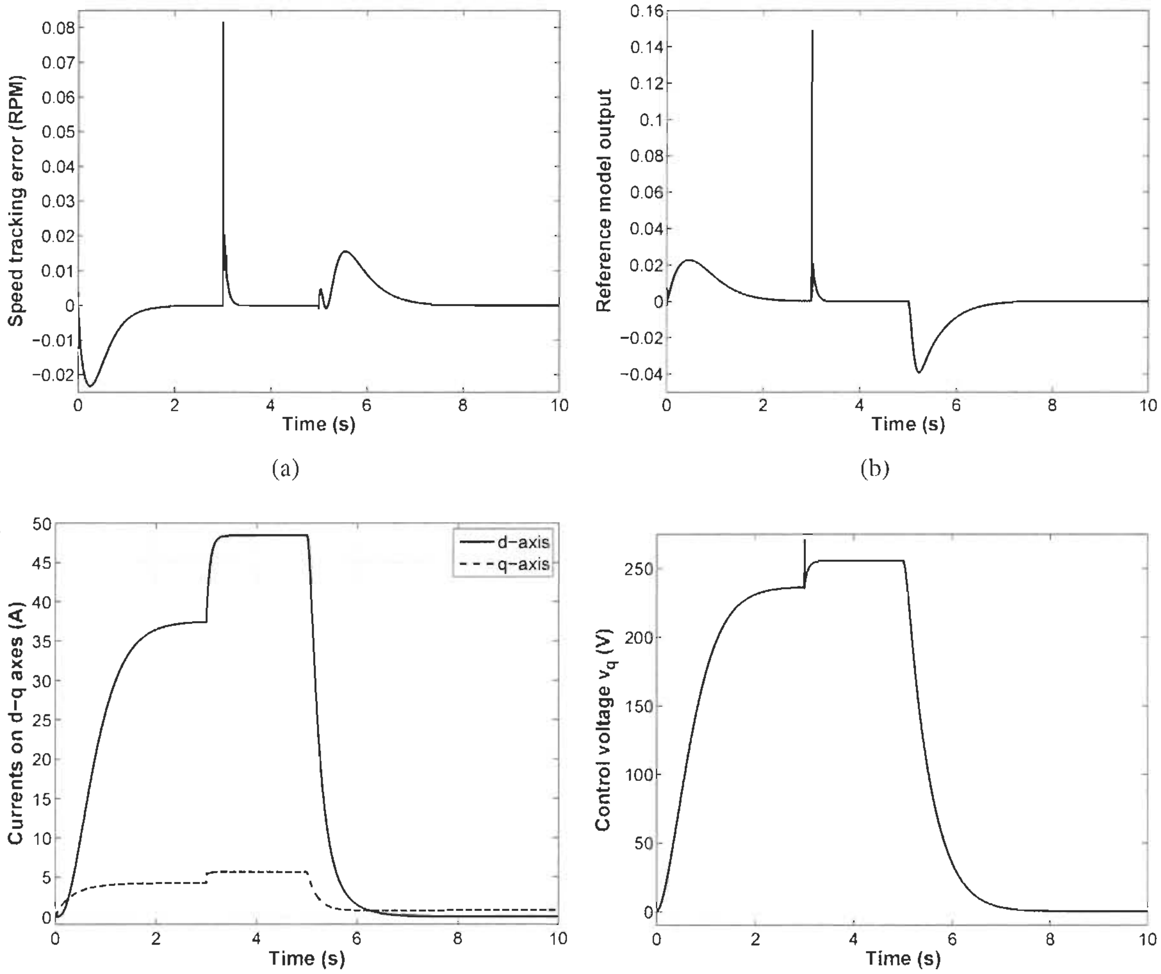
(a)

(b)

(c)

(d)

FIGURE 7.38 – Adaptive FLC response with parameters variation : (a) velocity error ; (b) reference model output ; (c) d-q axes currents ; and (d) applied voltage v_q^*.



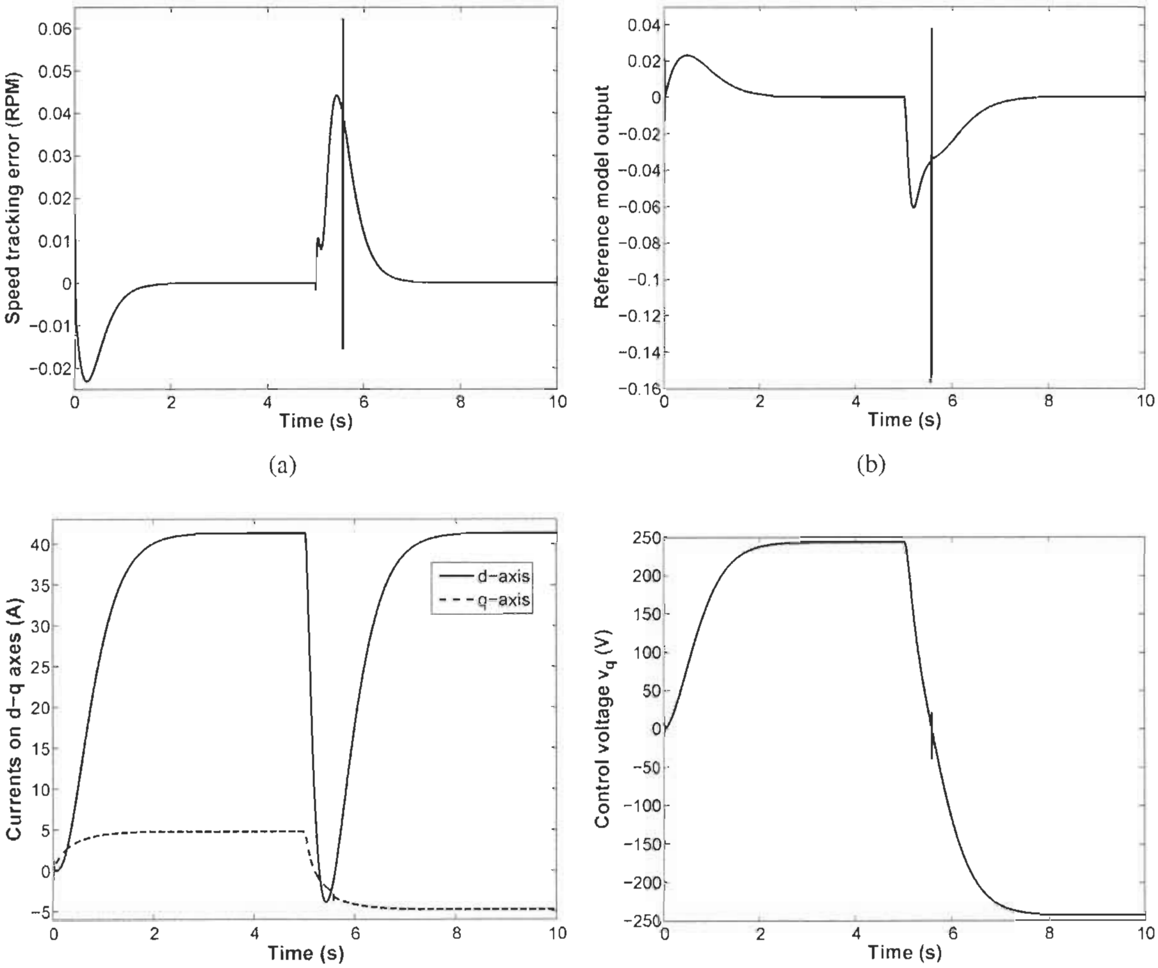
(a)

(b)

(c)

(d)

FIGURE 7.39 – Adaptive FLC response with load torque disturbance : (a) velocity error ; (b) reference model output ; (c) d-q axes currents ; and (d) applied voltage v_q^*.



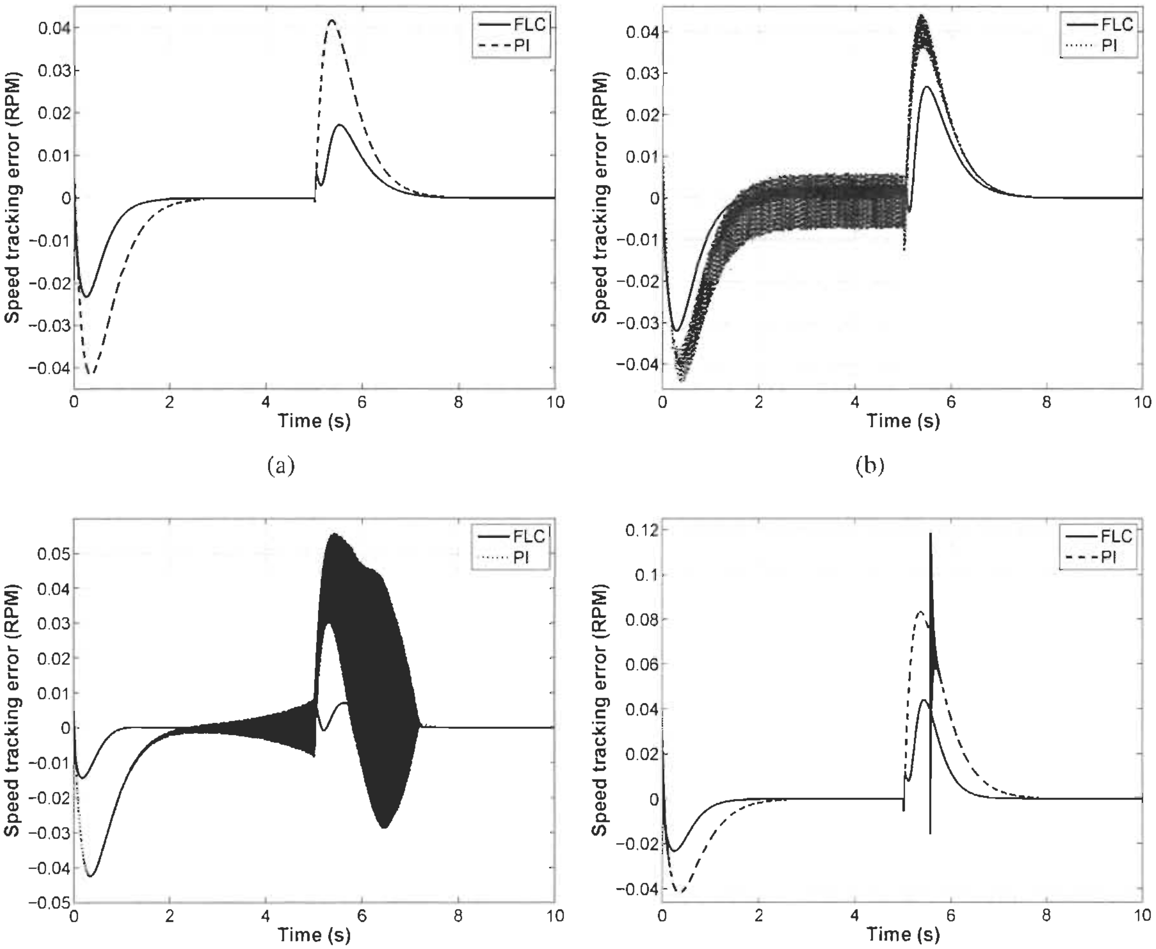
(a)

(b)

(c)

(d)

FIGURE 7.40 – Adaptive FLC response with magnified nonlinear friction : (a) velocity error ; (b) reference model output ; (c) d-q axes currents ; and (d) applied voltage v_q^*.



(a)

(b)

(c)

(d)

FIGURE 7.41 – Speed tracking error for : (a) nominal parameters ; (b) 50% nominal parameters ; (c) 150% nominal parameters ; and (d) magnified friction.

The FLC controller is compared in similar operating conditions to the well-known vector control technique, which is widely used in the industry. Both controllers are tuned to reach similar transient response and overall performance in nominal case for a fair comparison. Next, different situations corresponding to varying operating conditions are presented to both controllers to highlight their respective performance. The superiority of the FLC over conventional control techniques is clearly shown in these results. Fig. 7.41(a) shows the speed tracking error of both controllers in the nominal case. The FLC outperforms the vector controller by its high tracking precision, i.e., the speed error amplitude is kept a very small magnitude. Next, electrical parameters are decreased by 50% and then increased to 150% their nominal values. The performance of both controllers is depicted in Fig. 7.41(b) and Fig. 7.41(c). As it can be seen, the FLC copes successfully with the changes by keeping similar performance as in nominal parameters case. However, oscillations are observed with the vector controller in the presence of parameters variation. Finally, when the nonlinear Coulomb friction term is magnified 5 times, the speed error fluctuates with the vector controller at zero velocity crossing (Fig. 7.41(d)). On the other hand, the FLC controller attenuates the friction effects with no oscillations as it is shown in the previous simulation.

7.10 Conclusion

This chapter presents several soft-computing based adaptive techniques for high performance PMSM drives. Simplified adaptive control structures have been proposed with a speed and disturbance observer. To achieve better robustness to uncertainties, a sensorless ANN-based adaptive control strategy is proposed. However, tuning is not trivial in this kind of cascaded control structures. This structure is further simplified and compared to the well-known PI-based vector controller. Comparison shows better performance in transient, steady-state, and standstill conditions in the presence of parametric uncertainties. This comparison can be extended further by adding d-q axes decoupling terms to the vector control. On the other hand, an ANN-based speed observer is proposed. However, the speed estimation shows sensitivity to flux variations. This problem is solved with an ANN-based nonlinear observer. This strategy achieves good results with a higher tolerance to parameters variation. Finally, an adaptive fuzzy control strategy shows its ability to achieve similar performance with a single controller. In addition, this structure uses no voltage or current transducers reducing the number of sensors with respect to other techniques. However, this controller achieves high performance at the expense of reduced efficiency since no current control loop is used. Moreover, a Lyapunov stability-based adaptation technique is used as an alternative to the conventional heuristic tuning methods.

PMSMs are widely used in many energy conversion systems such as wind turbines and electric vehicles. These systems make also use of batteries and converters for efficient energy transfer. Therefore, next chapter presents some intelligent adaptive approaches for control of converters and state of charge estimation for batteries.

Chapitre 8

Management and Control of Intelligent Energy Production Systems

8.1 Introduction

Renewable energy systems have received a thorough attention and have been considered lately as a way of fighting climate change. They also have been used as energy complement to standalone production systems. These systems are composed of a main energy source such as diesel generators, storage devices such as batteries, and a grid. The aim of using diesel generators as a main power source is to provide the grid with uninterrupted power. Usually, a main AC bus is used to connect the renewable energy based inverter and the diesel generator to the grid as shown in Fig. 8.1. Reliability and cost reduction are among the main advantages of such systems. However, diesel generators require frequent costly maintenance and they remain a source of pollution. Recently, the idea of reducing their operation time has received increasing interest from the scientific and industrial community. On the other hand, storage devices have a limited amount of energy. Therefore, optimal energy utilization is among the various challenges to be faced. But, optimal operation depends on the control accuracy of converters. A poor control performance ultimately results in reduced converters efficiency. This raises the urgency to consider alternative control approaches for efficient energy transfer to keep up with the increasingly stringent energy demand requirements.

DC-DC converters have been extensively used in many electric power supply systems,

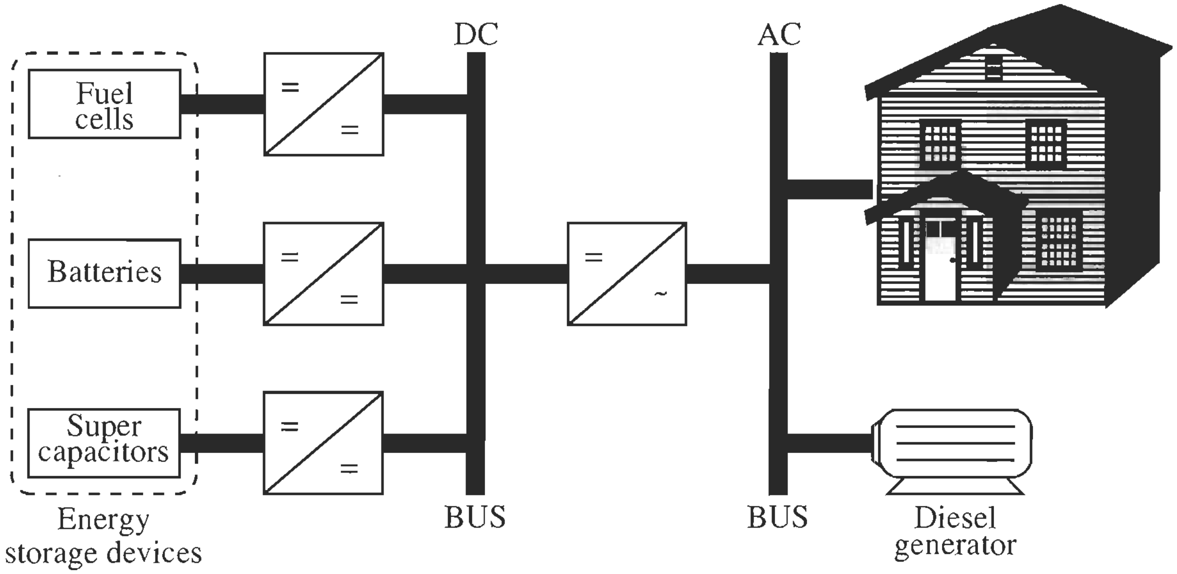


FIGURE 8.1 – Energy production system

thanks to their high efficiency, compact size, cheap price, and low weight [81, 82]. These advantages make them good candidates for many portable electronic devices such as, cellular phones, and mp3 players. However, in order to operate efficiently, high inductance is required to generate smooth current. These inductors raise the converter's cost and size, depleting the advantages listed above. In many industrial applications, high inductance is often used to reduce the current ripple and achieve acceptable tracking performance at the expense of the converter efficiency. The design of robust controllers reduces the size and the cost of inductors, which yields smaller, lighter, and cheaper power supply. However, controlling such systems faces numerous challenges that need to be addressed, such as parametric and load uncertainties. This raises the urgency of considering other control alternatives capable of dealing with uncertainties of higher magnitudes.

On the other hand, battery management systems have received a thorough attention and have been extensively used as power management tools in many applications such as, laptops, mobile phones and electric vehicles. Optimal battery energy utilization, battery life extension, and minimization of degradation effects are among the various challenges to be faced. However, optimal battery operation depends on the accuracy of the state of charge (SOC) algorithm.

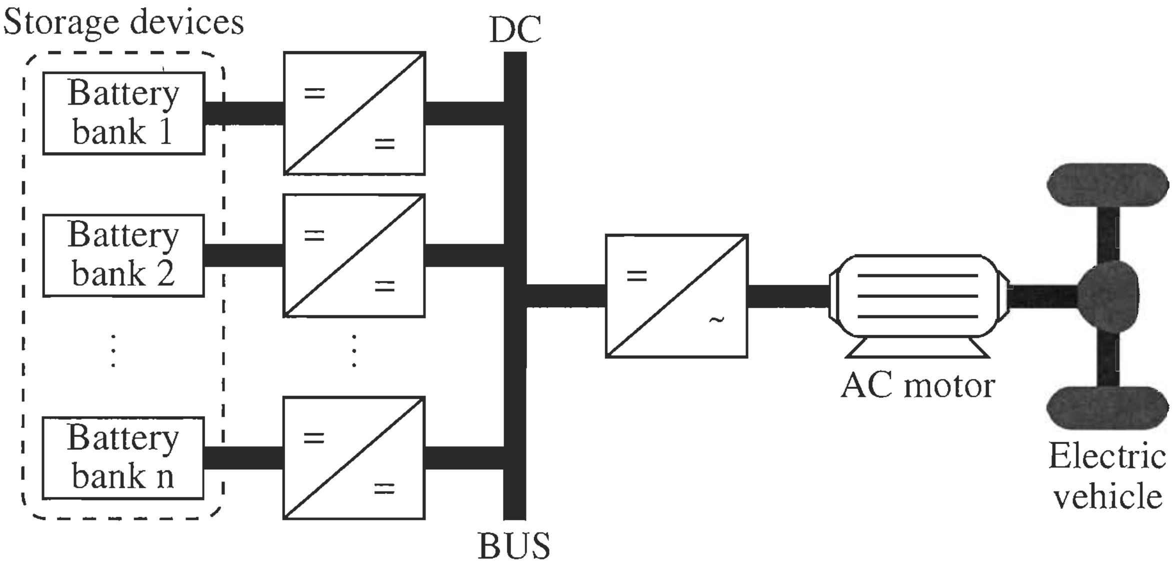


FIGURE 8.2 – Electric vehicle energy system

A bad SOC estimation might significantly damage the battery and ultimately result in reduced battery life.

In vehicular technology, electrical and power electronic systems are taking over mechanical, hydraulic, or pneumatic systems. These systems use multiple energy sources as shown in Fig. 8.2 such as batteries, fuel cells, and supercapacitors, which offer great flexibility to achieve higher performance. However, their maintenance and management are becoming more and more difficult and costly. Moreover, energy storage devices such as batteries have a limited amount of energy, which should be used efficiently. Optimal energy utilization of storage devices is among the various challenges to be faced.

This chapter presents advanced adaptive strategies for energy production systems. These control techniques are applied to the control of classical DC-DC boost converters, state of charge (SOC) estimation as well as to DC bus voltage control.

8.2 Modeling

8.2.1 Bidirectional DC-DC Converters

In this section, we represent DC-AC power system using a DC-DC bidirectional converter (Fig. 8.3). The IGBTs are controlled by the pulse width modulator (PWM). When the IGBT Q_1 is turned off and Q_2 is turned on, the circuit is divided into two independent parts allowing the current of the inductor and its energy to increase as shown in Fig. 8.4. When Q_1 is turned on and Q_2 is turned off, the energy stored in the inductor decreases and its voltage is added to the input voltage charging the output capacitor. Therefore, this repetitive operation makes the output voltage higher than the input voltage source. The same principle applies for the backward stage. For simplicity, the inverter is replaced by a resistance (load) in Fig. 8.5.

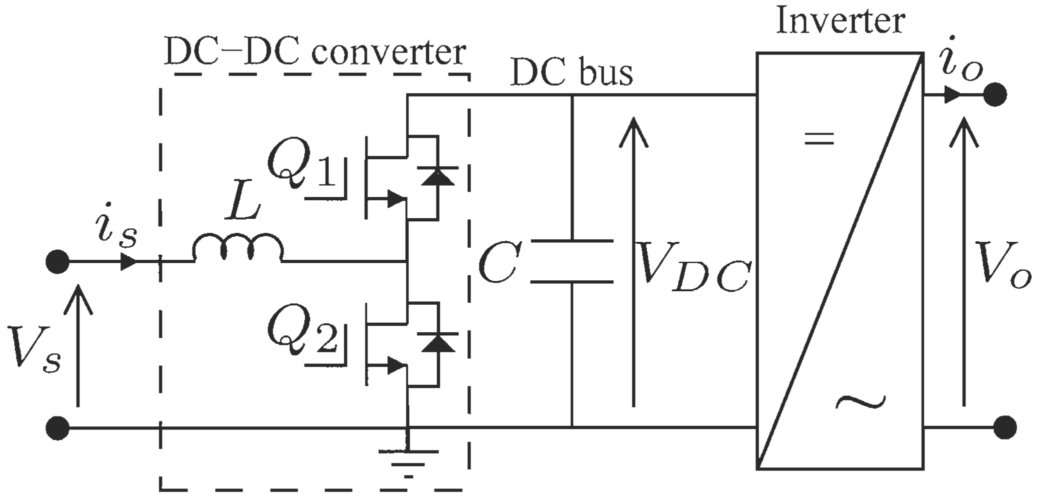


FIGURE 8.3 – DC-AC power system with a DC-DC bidirectional converter

The equivalent circuit of a DC-DC converter is illustrated in Fig. 8.5. Without loss of generality, the equivalent series resistance of the inductor, the capacitor, and the IGBTs, as well as the voltage drop across the diode are neglected.

Let us define the variable q such that $q = 1$ when $Q_1 =$ OFF and $Q_2 =$ ON, and $q = 0$ when $Q_1 =$ ON and $Q_2 =$ OFF.

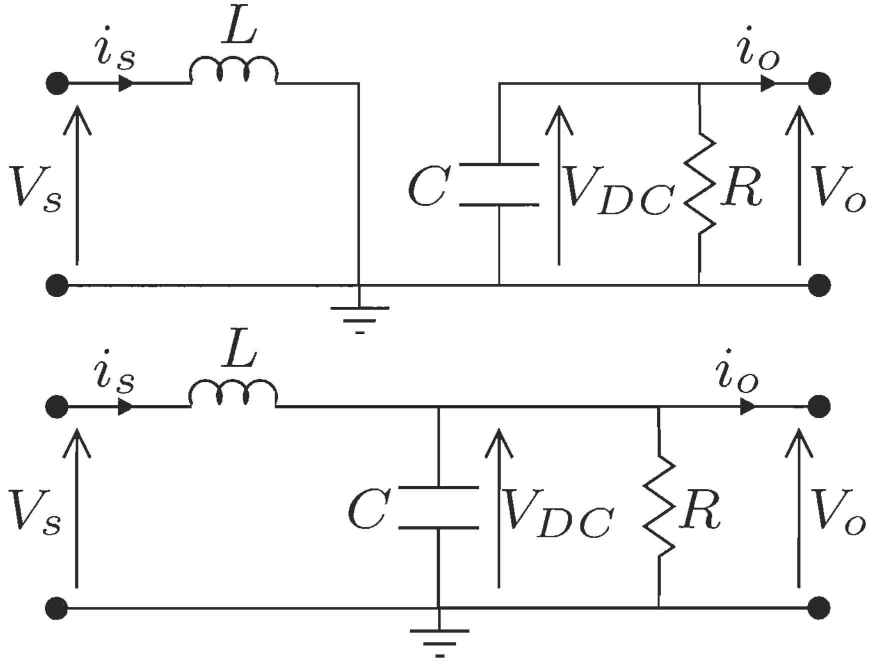


FIGURE 8.4 – DC-DC converter states of operation

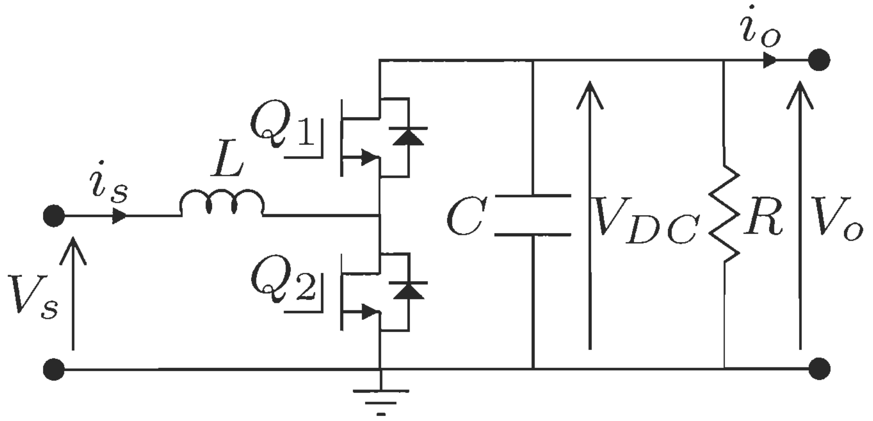


FIGURE 8.5 – DC-DC converter

The dynamic equation of the DC-DC converter can be described by the average mathematical model :

$$\begin{aligned}\frac{d}{dt}i_s(t) &= \frac{1}{L}\left(V_s(t) - (1-q)\ V_{DC}(t)\right)\\ \frac{d}{dt}V_{DC}(t) &= \frac{1}{C}\left((1-q)\ i_s(t) - i_o(t)\right)\end{aligned}$$

where L is the inductance, C the capacitance, $i_s(t)$ the inductor current, $i_o(t)$ the inverter current, $V_{DC}(t)$ the DC bus voltage, and $V_s(t)$ the supply voltage.

Now, define the control action as $\rho = 1-q$,

$$\begin{aligned}\frac{d}{dt}i_s(t) &= \frac{1}{L}\left(V_s(t) - \rho\ V_{DC}(t)\right)\\ \frac{d}{dt}V_{DC}(t) &= \frac{1}{C}\left(\rho\ i_s(t) - i_o(t)\right)\end{aligned} \tag{8.1}$$

It is noteworthy the nonlinearities within the system (8.1) in form of state dependencies. It is also clear that the converter's output voltage V_{DC} and its inductor current i_s are highly dependent on the parameters R, L, and C. Therefore, the control objective is to design a control law which ensures that the DC bus voltage V_{DC} tracks its desired reference voltage V_{DC}^*, in the presence of uncertainties. The proposed controller uses V_{DC}, i_s, and i_o as system's measurable states and the parameters, R, C, and L are assumed to be *a priori* unknown.

8.2.2 Batteries

The battery circuit model for an electrochemical battery is shown in Fig. 8.6. Its dynamic mathematical model can be described by the following equations [83] :

$$\dot{V}_p = -\frac{V_p}{R_dC} + \frac{V_{oc}}{R_dC} - \frac{I_b}{C}, \quad V_p \leq V_{oc} \tag{8.2a}$$

$$\dot{V}_p = -\frac{V_p}{R_cC} + \frac{V_{oc}}{R_cC} - \frac{I_b}{C}, \quad V_p > V_{oc} \tag{8.2b}$$

$$V_b = V_p - R_bI_b \tag{8.2c}$$

where,

V_{oc}	open circuit voltage
V_p	voltage across the capacitor
V_b	voltage at the battery terminals
I_b	current at the battery terminals
R_b	internal resistance
R_c	charging resistance
R_d	discharging resistance
C	capacitor

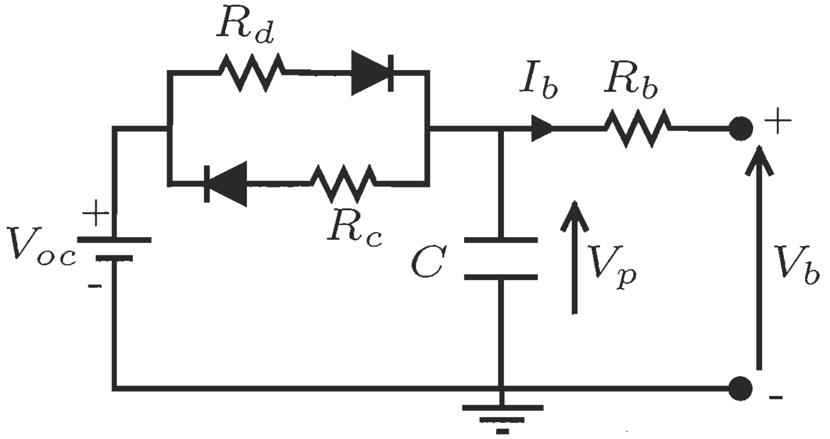


FIGURE 8.6 – Equivalent circuit of a battery

8.3 Adaptive FLC of a DC-DC Boost Converter

The DC-DC converters control problem has been a subject of great interest for many years and various control techniques have been proposed, such as, linear control [81], current mode control [82], predictive control [84], and sliding mode control [85]. Among various switching control methods, these control techniques use pulse width modulation (PWM). In the last decade, PWM switching control [86] was widely used in the proportional integral (PI) control structure. This technique is limited with the tradeoff between robustness and transient response. The current mode control technique is used for its fast transient response induced by the current loop, faster overload protection, and its ability to deal with parameter variations. This strategy consists of an external voltage loop with an inner current loop, which is known as multiloop control. On the other hand, boost converters have received a thorough attention lately due to their nonminimum phase nature, which can result in severe tracking errors, limit cycles, chattering, and excessive noise. In fact, boost converters are modeled as bilinear second order nonminimum phase systems with a highly uncertain load resistance. In [87], experimental results of a buck converter using a fuzzy controller and a proportional integral derivative (PID) showed a comparable performance. But, the fuzzy controller showed faster transient response and better tracking performance when applied to a boost converter.

The averaged model technique [88] is usually used in nonlinear DC-DC converters analysis. However, small signal analysis neglects the system's dynamic at high frequency, the effect of saturation, and initial conditions. On the other hand, the presence of high uncertainties and varying operating conditions changes significantly the system's dynamics. In this case, the controller design cannot be based efficiently on presumably accurate mathematical models. When bounded uncertainty exists, a robust control technique such as sliding-mode control can be applied. But, robustness to uncertainties is obtained only when sliding mode truly occurs.

For the boost converter to produce a desired output voltage, let us denote the output voltage error as $e = V_o - V_r$ and its discrete derivative as $\Delta e = e(t) - e(t-1)$. Before proceeding further, let us define the following reference model as $s = e + \psi\, \Delta e$, with ψ being a positive constant. In this work, we use a 2-input 1-output fuzzy logic controller in which e and Δe are fed as inputs. The control action provided by the FLC is the increment $\Delta\rho$ to compute the duty cycle that is then fed to the converter. The inputs, e and Δe, and the output $\Delta\rho$ can easily be modeled through human-like reasoning mechanism. Note that the output $\Delta\rho$ is proportional to the inputs. As such, the FLC controller forces the error e and its discrete derivative Δe to approach the hyperplanes $e = 0$ and $\Delta e = 0$ for a fast convergence requirement in $s = 0$. The resultant control scheme is illustrated in Fig. 8.7.

The membership functions for the input variables, e and Δe are shown in Fig. 8.8(a) and

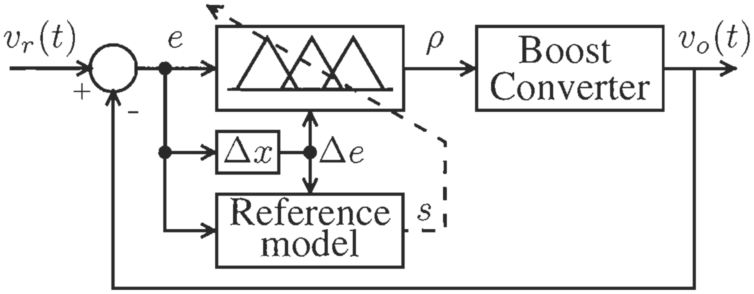


FIGURE 8.7 – Adaptive fuzzy control scheme for DC-DC boost converters

8.8(b), respectively. The linguistic terms used for the input-output membership functions are labeled as *"NL (Negative Large)", "NS (Negative Small)", "Z (Zero)", "PS (Positive Small)",* and *"PL (Positive Large)"*. An empirical analysis for the parameters of membership functions is performed to improve the FLC's performance. The *if-then* rules for fuzzy inference are reported in Table 5.2. The defuzzification method for the output, $\Delta\rho$, is chosen to be the centroid of area, which can also be written as

$$y(k) = G^T(k)W(k)$$

where $W(k)$ is the fuzzy logic consequent part vector at instant k and $G(k)$ is its antecedent part vector of known functions (regressor) at instant k defined as

$$G(k) = \frac{\mu_{B^p}(y)}{\sum_{p=1}^{R} \mu_{B^p}(y)}$$

Then, the fuzzy logic consequent parts can be tuned online using a gradient descent technique :

$$W(k) = W(k-1) + \Delta W(k) = W(k-1) + \Gamma G(k)s$$

where $\Gamma = [\gamma_1, \gamma_2, \ldots, \gamma_R]^T$ and γ_i is a positive constant.

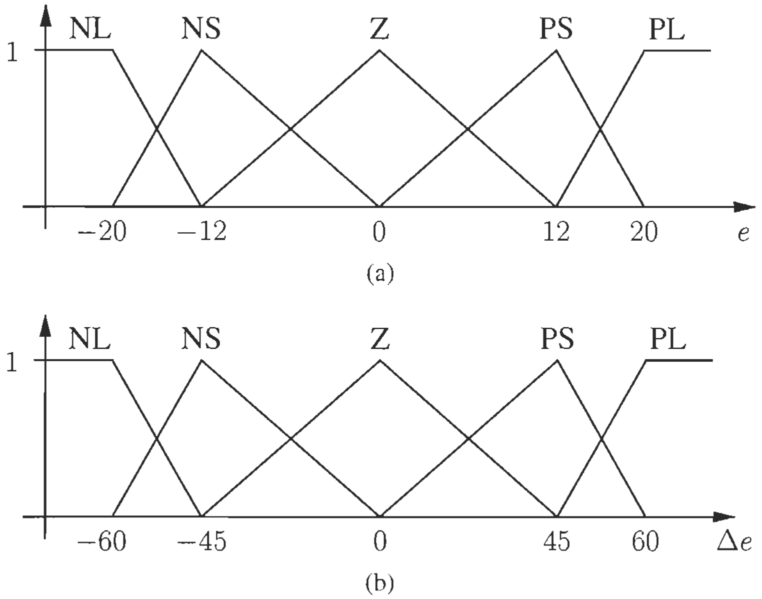


FIGURE 8.8 – Fuzzy membership functions for : (a) voltage error ; and (b) voltage error derivative.

8.3.1 Setup

The DC-DC converter (Fig. 8.5) presented by its average model dynamics (8.1) is used in boost mode. Therefore, the IGBT Q_1 is blocked and the duty cycle produced by the proposed controller is fed to Q_2. The converter's nominal parameters are defined by : $V_s = 12\ V$, $R = 2\ \Omega$, $C = 0.5\ mF$, and $L = 30\ \mu H$. The simulations are conducted for time period of $t = 0.25$ sec, where the desired output voltages $V_r = 20\ V$, for $0 \leq t \leq 0.125$ sec, and $V_r = 15\ V$, for $0.125 < t \leq 0.25$ sec. The system's model is implemented using SimPowerSystems Simulink toolbox in Matlab and the adaptive fuzzy logic controller algorithm is coded in C. The switching and sampling frequencies of both controllers are set to 1 KHz.

8.3.2 Results

Three different simulation sets are conducted to evaluate the performance of the proposed adaptive FLC under various parametric configurations of a DC-DC boost converter. The performance metrics of the boost converter are the output voltage V_o, the voltage tracking error e, the inductor current $i_L(t)$, and the duty cycle ρ. In order to justify the use of the proposed controller, the same simulation sets are carried out to test the converter's performance using a conventional PI controller with $K_p = 10^{-3}$, and $K_i = 1.5$.

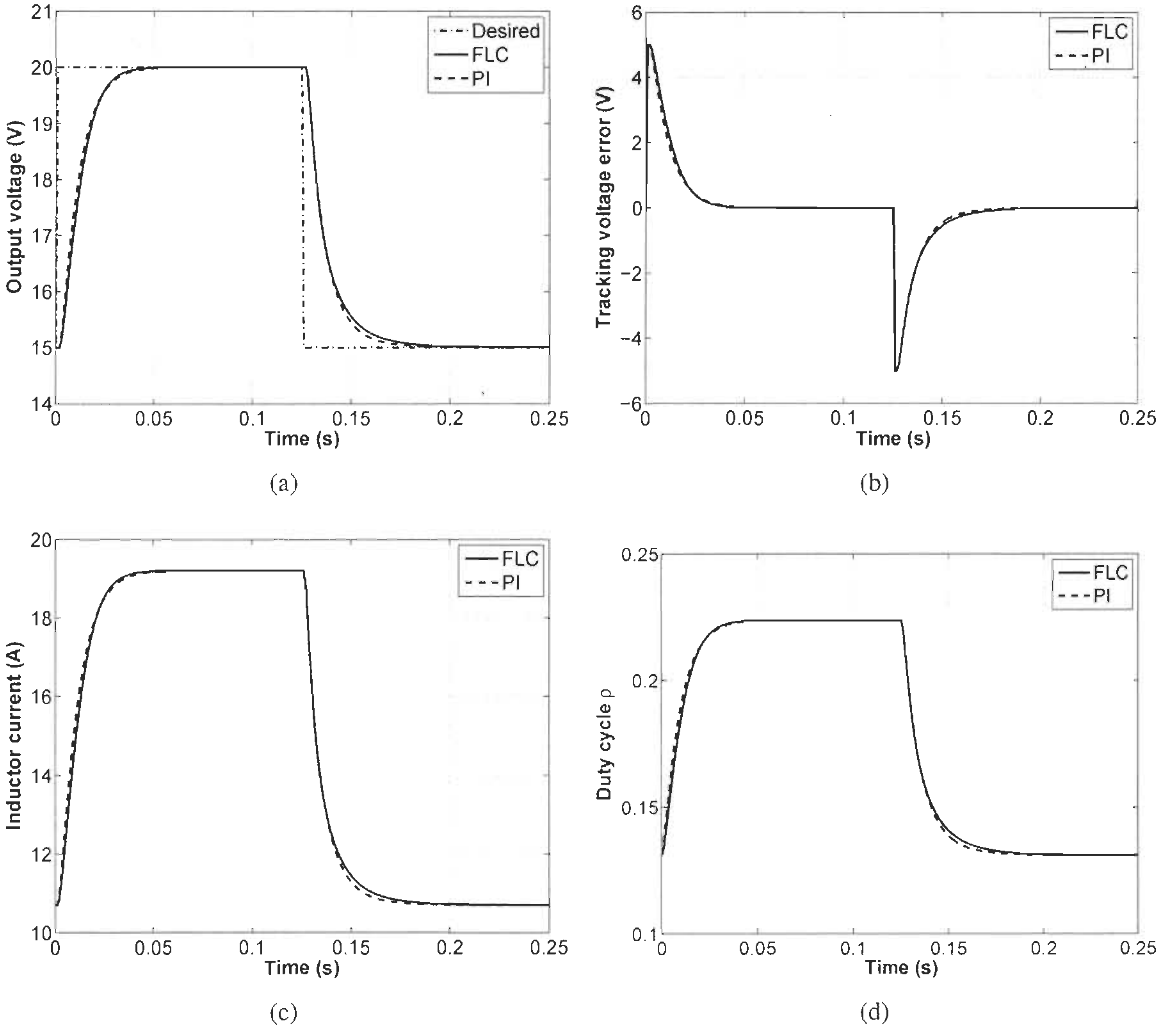


FIGURE 8.9 – Adaptive FLC and PI responses with nominal parameters : (a) output voltage ; (b) voltage error ; (c) inductor current ; and (d) controller's duty cycle ρ.

In this simulation, the aforementioned nominal values are used to simulate the converter's dynamics. As shown in Fig. 8.9, both controllers lead to a satisfactory performance in this case as they both attenuate the voltage tracking error to zero within comparable time delays. Both controllers were tuned to reach similar transient response and overall performance in nominal case for a fair comparison of both controllers in the next simulation sets where only one parameter is allowed to vary at a time.

The purpose of this simulation is to study the controller's ability to sustain various intensities of load uncertainties. For that, the converter's load is increased to $R = 0.5\ \Omega$. As shown in Fig. 8.10, the adaptive FLC performance is not affected by the load change in the sense that it could still decay the voltage error to zero in less than 0.05 s. On the other hand, a slower response is obtained with the PI controller.

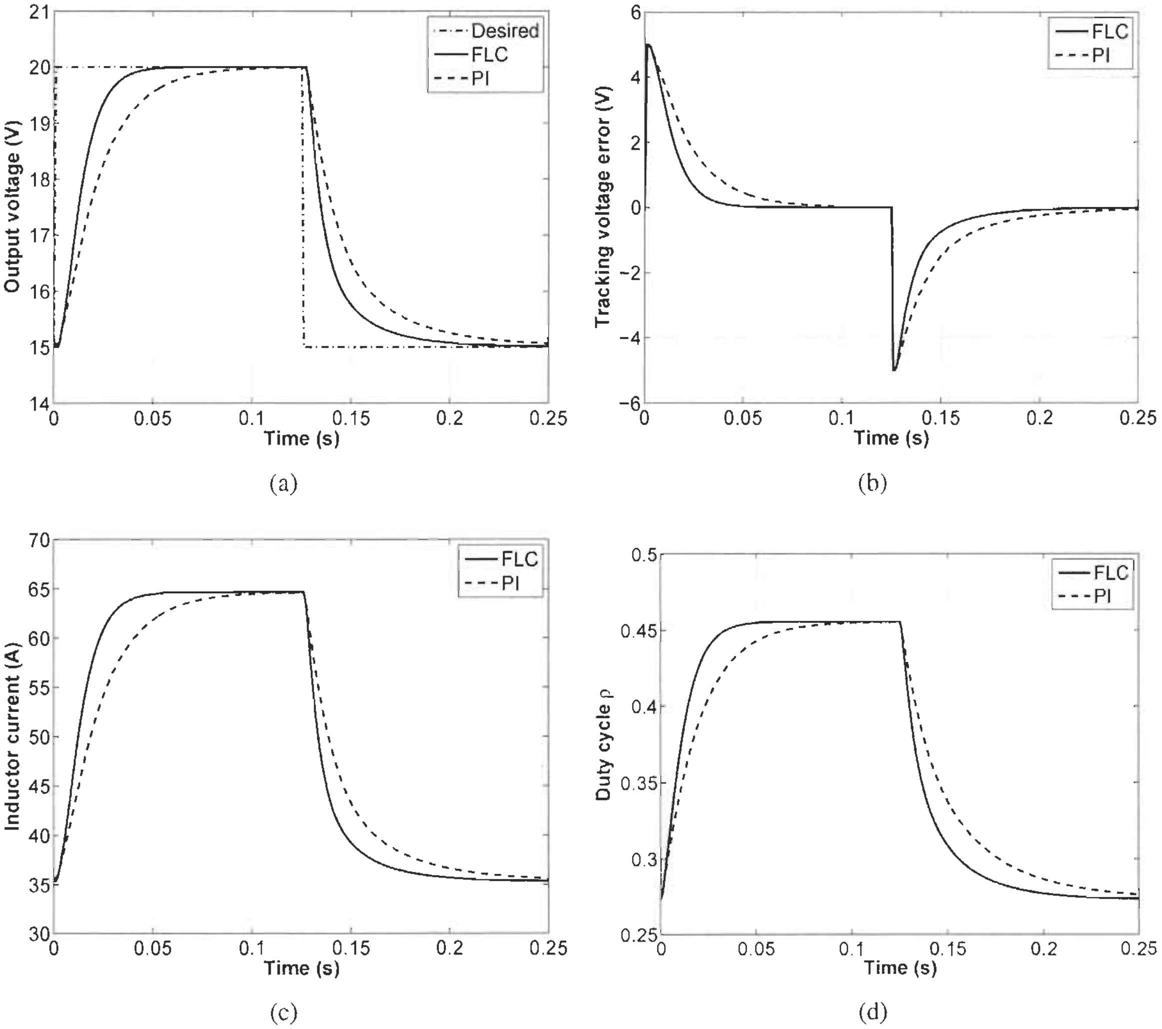
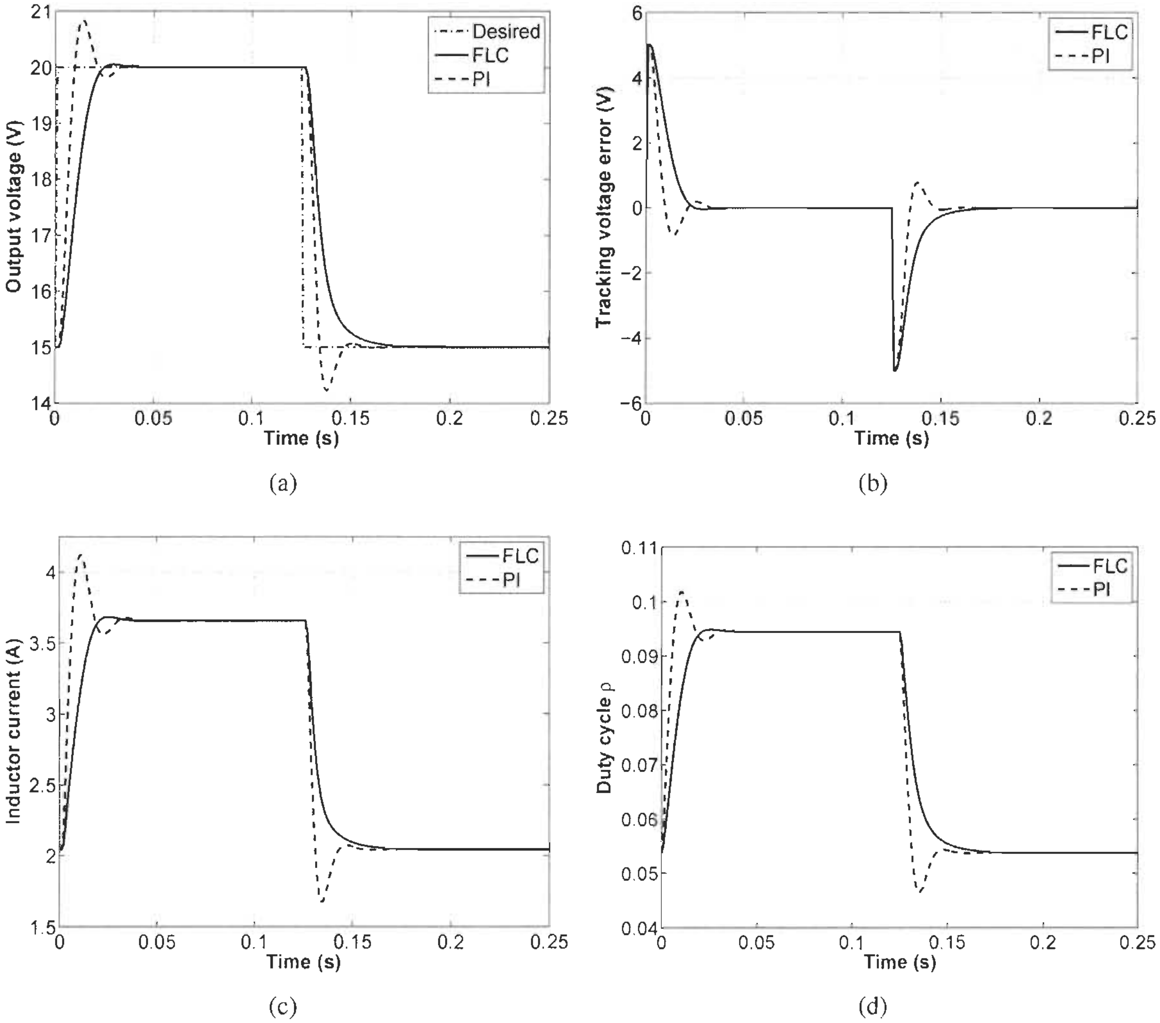


FIGURE 8.10 – Adaptive FLC and PI responses with a higher load : (a) output voltage ; (b) voltage error ; (c) inductor current ; and (d) controller's duty cycle ρ.

Then, the converter's load is decreased to $R = 10\ \Omega$. The results are shown in Fig. 8.11. The proposed controller keeps similar performance to that in the nominal case as opposed to the overshoot obtained with PI controller. The ability of the FLC controller to cope with large load uncertainties is clearly shown.



(a)

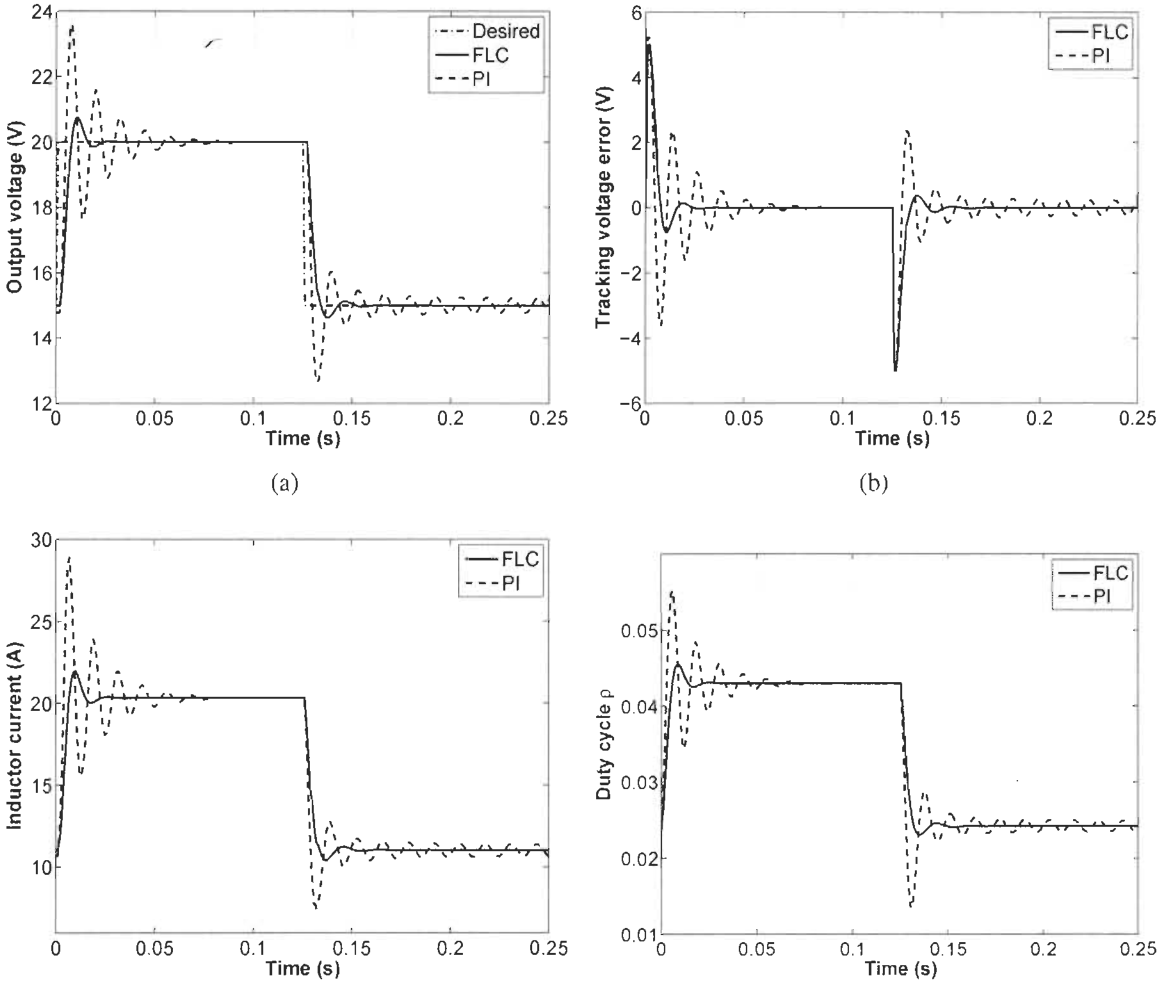
(b)

(c)

(d)

FIGURE 8.11 – Adaptive FLC response with a lower load : (a) output voltage ; (b) voltage error ; (c) inductor current ; and (d) controller's duty cycle ρ.

In the next simulation, we observe the converter's behavior by varying the values of the inductor and the capacitor. First, the converter's inductor is reduced by 30 times with respect to its nominal value ($1\ \mu H$). The results are depicted in Fig. 8.12, the performance of the FLC is not affected much by the change and acceptable performance is reached. However, more overshoot is observed with the PI controller and its response is fairly fluctuating as opposed to a smooth and steady convergence behavior with the FLC controller. The superiority of the adaptive FLC is revealed more clearly in this simulation. Henceforth, the use of the adaptive FLC reduces the size of the inductor, which yields smaller, lighter, and cheaper power supply.



(a)

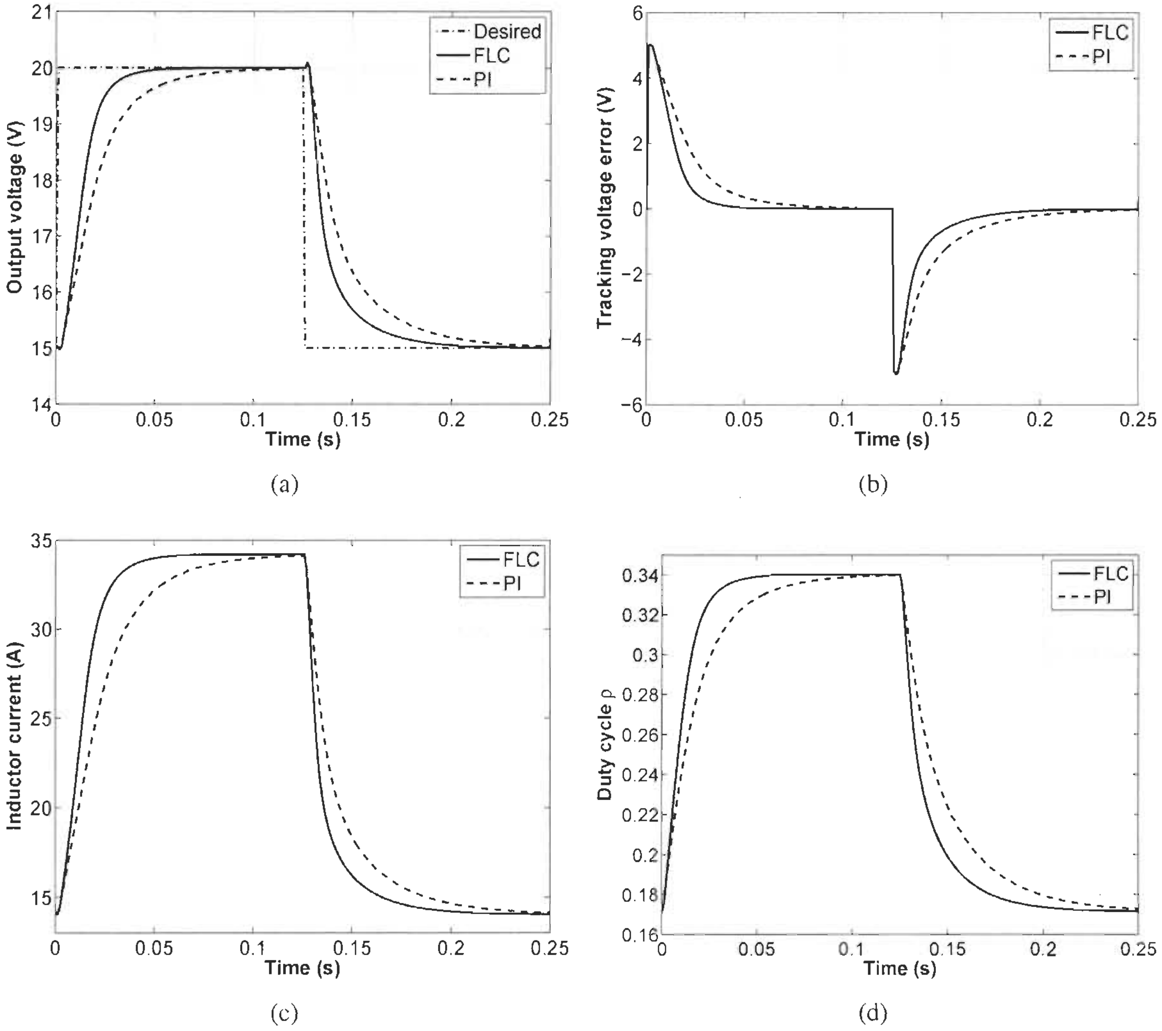
(b)

(c)

(d)

FIGURE 8.12 – Adaptive FLC and PI responses with inductor variation : (a) output voltage ; (b) voltage error ; (c) inductor current ; and (d) controller's duty cycle ρ.

Secondly, the converter's capacitor is given a value of 0.1 mF. The results are depicted in Fig. 8.13. As can be seen, the response of the PI controller slows down as opposed to the adaptive FLC that copes easily with such uncertainty.



(a)

(b)

(c)

(d)

FIGURE 8.13 – Adaptive FLC and PI responses with capacitor variation : (a) output voltage ; (b) voltage error ; (c) inductor current ; and (d) controller's duty cycle ρ.

8.4 Adaptive DC Bus Control

Over the years, researchers attempted various control techniques and several solutions have been proposed including conventional and advanced control laws. Classical linear control laws [89] are widely used in the industry because of their simplicity. However, their performance degrades significantly in the presence of varying operating conditions. This problem has been addressed in [90], where an adaptive control strategy uses a PI controller for DC bus regulation with an adaline-based currents control tuned online using least-mean squares (LMS) algorithm. A cascaded linearization based control structure is proposed in [91] for AC/DC boost converters. In [92], the DC bus voltage regulation problem is investigated using passivity-based control theory. Advanced controllers preserve systems nonlinearity characteristics in their design. Therefore, they offer good performance under uncertainties such as temperature, and parameter variations. Several advanced control techniques have been applied to control different types of converters such as sliding mode control. But, their good performance is often achieved at the expense of added complexity and difficulty on the design approach. In [93], a sliding mode controller is proposed for a multisource/multiload electrical hybrid system using fuel cells and supercapacitors. However, robustness to parameter variations is obtained only when sliding mode truly occurs. In [94], a fuzzy-neural sliding-mode (FNSM) control system has been proposed to cope with uncertainties, such as parameter variations.

Let $e_v = V_{DC} - V_{DC}^*$ denotes the DC bus voltage tracking error and $e_s = i_s - i_s^*$ the source current tracking error, where V_{DC}^* and i_s^* are the DC bus voltage and source current reference signals, respectively. Adaptive control theory is used in this work to highlight its advantage over classical control techniques. As depicted in Fig. 8.14, the adaptive controller achieves DC bus voltage tracking by minimizing the error e_v between the reference and actual voltages to deliver a desired source current signal i_s^*. A PI controller is used to minimize the error e_s, (i.e., current regulation). The duty cycle is then fed to a Pulse Width Modulation (PWM) algorithm.

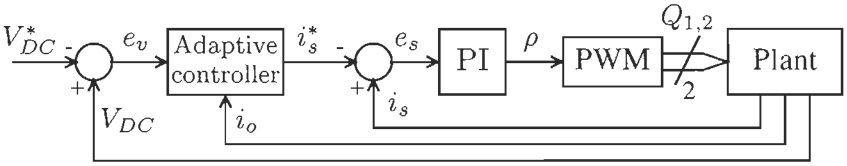


FIGURE 8.14 – Adaptive DC bus control scheme

The aim of the adaptive controller is an efficient power flow by maintaining e_v at zero (i.e.,

$P_s \approx P_o$). This is achieved with *a priori* unknown dynamics. P_s and P_o are the source and output power variables, respectively.

Therefore, we can express the relationship between the source current i_s and the output (inverter) current i_o by,

$$i_s = \frac{V_{DC}}{V_s}\, i_o = k\, i_o$$

where, k is assumed to be unknown and tracked by the adaptive controller. Therefore, the control law is defined as,

$$i_s^* = \hat{k}\, i_o - k_d\, e_v$$

where, k_d is a positive gain that ensures stability and robustness of the adaptive controller.

Therefore, we can define the adaptation law as,

$$\dot{\hat{k}} = -\gamma\, i_o\, e_v$$

where, γ is a positive gain that defines the convergence rate.

8.4.1 Setup

To demonstrate the performance of the proposed adaptive controller, a set of computer simulation runs is carried out on a DC-DC converter depicted in Fig. 8.5. The system's parameters are summarizes as follows, the DC bus reference voltage $V_{DC}^* = 400$V, the inductance $L = 250\mu$H, and the capacitance $C = 1$mF. The converter's model is implemented using PSIM Software and the adaptive control algorithm is coded in C. The control structure is implemented using Matlab/Simulink. The switching and sampling frequencies are both set to 5 kHz. The system's performance metrics are the inverter current i_o, DC bus voltage V_{DC}, source currents i_s^* and i_s, duty cycle ρ, and the adaptive parameter estimate $\hat{k}$. The proposed controller is also compared to its conventional PI counterpart under the same conditions as depicted in Fig. 8.15 with $k_p = 50$, and $k_i = 10^3$. The adaptive controller parameter k_d is set to 50.

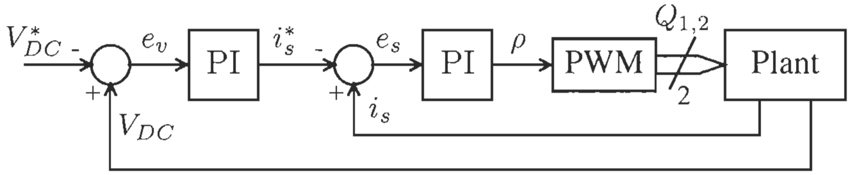
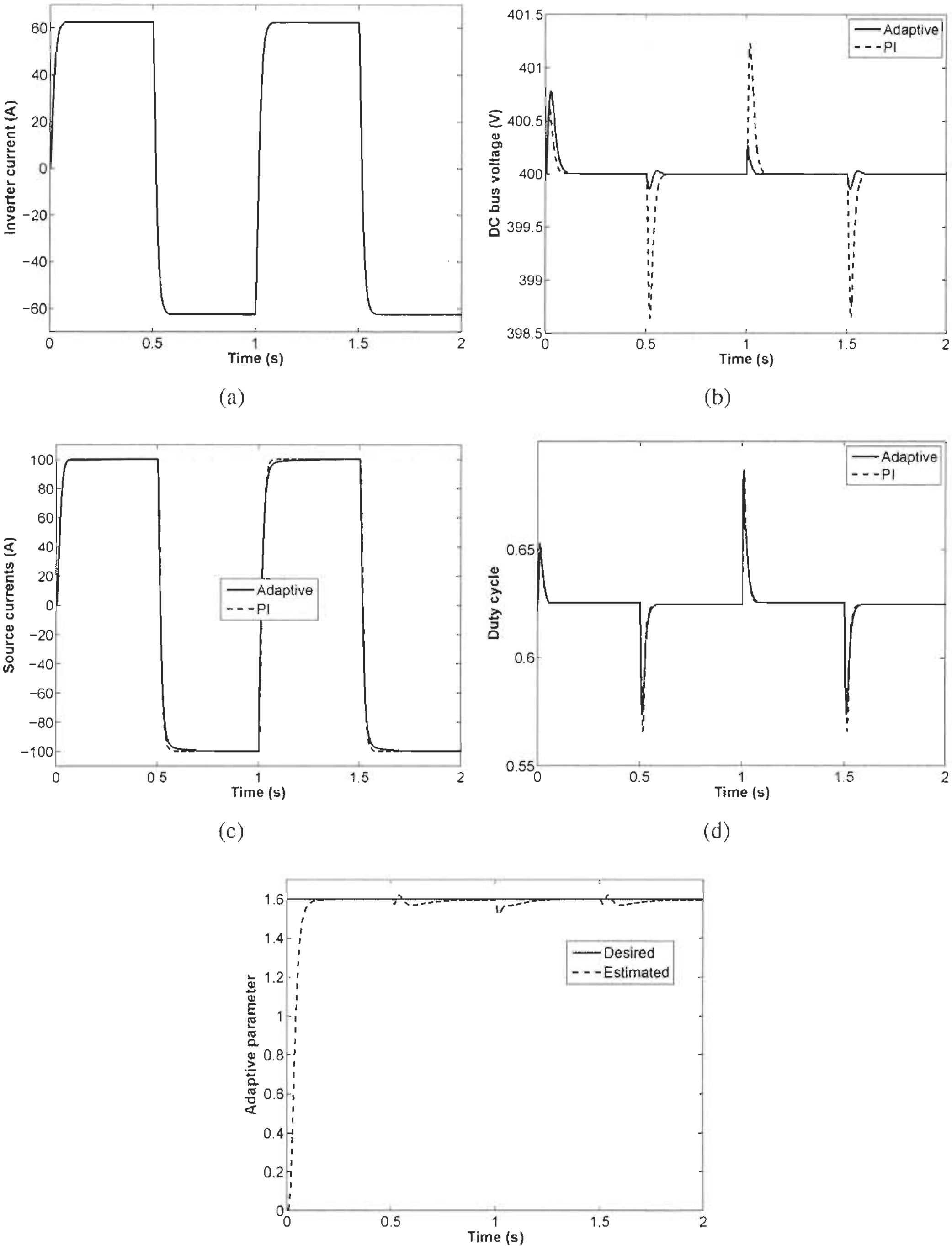


FIGURE 8.15 – Classical PI-based cascaded DC bus control scheme

8.4.2 Results

Two numerical simulations are carried out to study the proposed controller's performance. In the first simulation, the aforementioned nominal values are used to simulate the system's dynamics and the inverter current is taken as the step response of a critically damped second order system with a natural frequency of 100 rad/s, as shown in Fig. 8.16(a). In this case, the voltage variation of the power source is ignored and the source voltage V_s is set to 250V. As shown in Fig. 8.16, the DC bus voltage tracking error decay gradually with the adaptive controller before stabilizing within a negligible amplitude. On the other hand, a satisfactory trajectory tracking is obtained with the PI controller. Although both controllers were tuned to achieve similar performance in nominal case, the adaptive controller outperforms its PI counterpart with high tracking accuracy. Moreover, the proposed controller achieves better performance with less control effort as it is shown in Fig. 8.16(d), which yields a more efficient energy transfer.

Next, the power source voltage variation is taken into account to evaluate the controller's performance to uncertainties. The source voltage V_s is set as $250 + 10\sin(\pi t)$V, where t is the time index. As it can be seen in the results depicted in Fig. 8.17, the adaptive controller compensates for the variation of the input voltage while the DC bus voltage tracking error keeps oscillating around its reference signal with the PI controller. Thus, the power source uncertainties have little effect on the proposed controller's performance as opposed to the PI controller, which yields a more efficient system's operation.



(a)

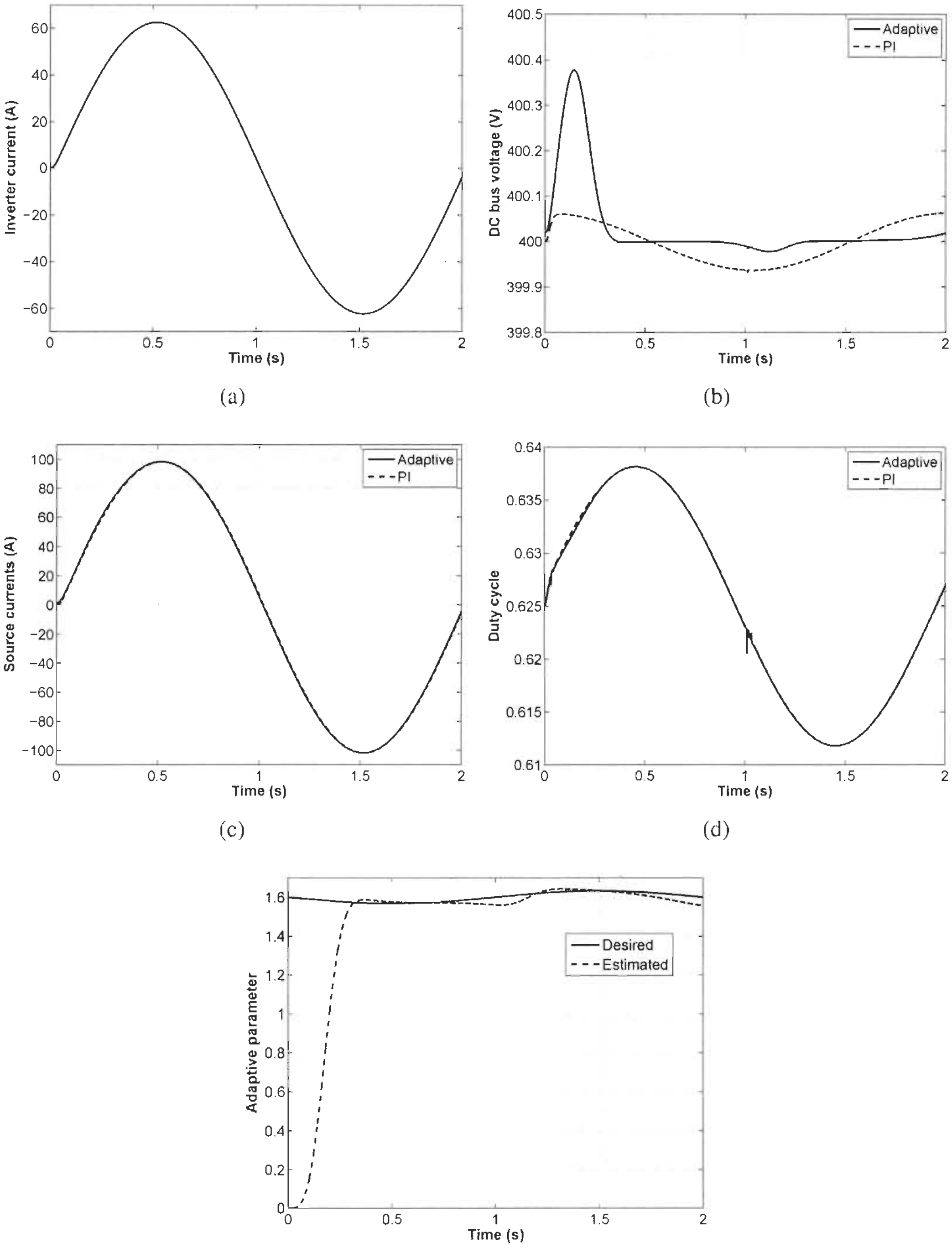
(b)

(c)

(d)

(e)

FIGURE 8.16 – DC bus control under square power demand : (a) inverter current i_o ; (b) DC bus voltage V_{DC} ; (c) source current i_s ; (d) duty cycle ρ ; and (e) adaptive parameter estimate $\hat{k}$.



(a)

(b)

(c)

(d)

(e)

FIGURE 8.17 – DC bus control under sinusoidal power demand : (a) inverter current i_o ; (b) DC bus voltage V_{DC} ; (c) source current i_s ; (d) duty cycle ρ ; and (e) adaptive parameter estimate $\hat{k}$.

8.5 Observer-based State-of-Charge (SOC) Estimation

Several conventional approaches exist for determining the SOC. The open circuit voltage (OCV) method is a reasonable way to estimate a battery SOC. The battery voltage is correlated with the concentration of electrolyte, which increases and decreases with the battery charge status. However, this holds only when the internal battery dynamics are in an equilibrium state (i.e., no current flows through the battery for several minutes or hours). This correlation is also affected during the life-time of batteries by temperature variations and aging, i.e., capacity is known to be gradually decreasing with the number of charge/discharge cycles and depth of discharge. On the other hand, the Coulomb counting method is a more straightforward method to estimate the SOC of a battery. In this method [95], also called Amp-hour (Ah) balancing method [96], the battery entering and leaving current is measured and the SOC is updated accordingly through integration. However, the start-up error and the current sensor errors are cumulative, which introduces a drift and poor precision problem since the estimator is open-loop based. Moreover, the battery's capacity changes as the batteries age are not taken into account. Although this method has some serious drawbacks and the estimates carry an average of 15% error [95], it remains the simplest approach for real-time industrial applications. Another method called hybrid estimation technique combines basically the two first methods. The battery SOC is estimated with the Coulomb counting method when the battery is under use and whenever the battery reaches an equilibrium state, the SOC is updated with the OCV method to reset the accumulated errors. However, some applications require a continuous operation of batteries, and hence, do not allow them to reach an equilibrium state for an accumulated error reset. This raises the urgency of considering other SOC estimation alternatives.

Since the open-circuit voltage (OCV) is directly correlated with the state of charge of the battery, V_{oc} is therefore used to estimate the state of charge. Henceforth, a precise OCV estimate leads to accurate SOC estimation. In this work, the relationship between OCV and SOC is considered to be constant, and hence, compensation for battery aging and temperature effects is out of the scope of this work. Let us consider the discharge mode so that we can set $R = R_d$, the dynamics formulation (8.2a) can be written as,

$$\dot{V}_p = -\frac{V_p}{RC} + \frac{V_{oc}}{RC} - \frac{I_b}{C}$$

Substituting V_p from (8.2c) :

$$\dot{V}_b = -R_b \dot{I}_b - \frac{1}{RC} V_b - \left(\frac{R_b}{RC} + \frac{1}{C}\right) I_b + \frac{1}{RC} V_{oc} \tag{8.3}$$

The system's dynamics (8.3) can also be written in a form,

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

where $x \in \mathbb{R}^2 = [V_b,\, V_{oc}]^T$ is the state vector, $u \in \mathbb{R}^2 = [I_b,\, \dot{I}_b]^T$ is the input vector and $A \in \mathbb{R}^{2\times2}$, $B \in \mathbb{R}^{2\times2}$, and $C \in \mathbb{R}^2$ are given by,

$$A = \begin{bmatrix} -\frac{1}{RC} & \frac{1}{RC} \\ 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} -\left(\frac{R_b}{RC} + \frac{1}{C}\right) & -R_b \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1\,0 \end{bmatrix}$$

Therefore, the observer is defined as,

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

with, G being the observer gain matrix.

Let us define the observer error as,

$$e = \hat{x} - x$$

Take the derivative of e and substitute for $\dot{x}$, and $\dot{\hat{x}}$,

$$\dot{e} = A\hat{x} + Bu + G(C\hat{x} - y) - Ax - Bu$$

Therefore, the observer dynamics is governed by,

$$\dot{e} = A_c e$$

where, $A_c = (A + GC)$ is a Hurwitz matrix. Therefore, G can be chosen to make the eigenvalues of the matrix $(A + GC)$ all have negative real parts such that,

$$e_z(t) = e^{(A+GC)t} e_z(t_0) \to 0$$

as $t \to \infty$. The observer gain matrix G can be found by solving the algebraic Riccati equation or by using a pole placement technique.

8.5.1 Results

The estimation strategy is implemented using Matlab/Simulink and the battery's model is implemented using PSIM software. Table 8.1 summarizes the battery's parameters along with their respective values. A lead-acid battery is chosen for the test because of its linear relationship between OCV and SOC. As such, the SOC is given by the following expression,

$$SOC = \frac{V_{oc} - V_b^0}{V_b^{100}}$$

where V_b^0 and V_b^{100} are the battery terminal voltages at $SOC = 0\%$, and $SOC = 100\%$, respectively.

In the case of the lead-acid battery in hand, $V_b^0 = 12V$ and $V_b^{100} = 13V$. The initial SOC is set to 100% and the initial open circuit voltage estimate $\hat{V}_{oc}$ is set to 12.5V, which corresponds to 50% SOC.

TABLE 8.1 – Battery's parameters

Parameter	Value	Unit
Charging resistance	$R_c = 3 \cdot 10^{-2}$	(Ω)
Discharging resistance	$R_d = 1 \cdot 10^{-2}$	(Ω)
Internal resistance	$R_b = 1 \cdot 10^{-3}$	(Ω)
Capacitor	$C = 1 \cdot 10^{-2}$	(F)

A set of computer simulations is carried out to study the proposed estimator's performance. The system's response is studied taking into account the battery voltage V_b and current I_b, the battery voltage estimation error e, and the SOC estimate error in (%). The aforementioned nominal values are used to simulate the system's dynamics. The simulation is conducted for time period of $t = 1$ sec, where the system is left at equilibrium state for $0 \leq t < 0.1$ sec, and the converter operates for $0.1 \leq t < 1$ sec. As shown in Fig. 8.18, a fast tracking is achieved at equilibrium state ($V_{oc} = V_b$). These errors are kept in a very negligible magnitude and decay gradually to zero in steady state, which yields high accurate SOC estimation.

Next, the system's capacitance is magnified 10 times its nominal value. As shown in Fig. 8.19, this change results in a system's time constant variation, which decreases the estimator's accuracy during transitions. However, the estimation error is kept in a small magnitude and still converges to zero in steady state.

Finally, the battery aging effect on the estimator's performance is considered. For that, the battery's impedance R_b is magnified 2 times its nominal value. This change causes a state of charge estimation error increase and a non-zero error in steady state, which is expected since as stated earlier, the battery aging effect compensation is out of the scope of this work. However, as shown in Fig. 8.20, the estimation error remains in an acceptable magnitude (around

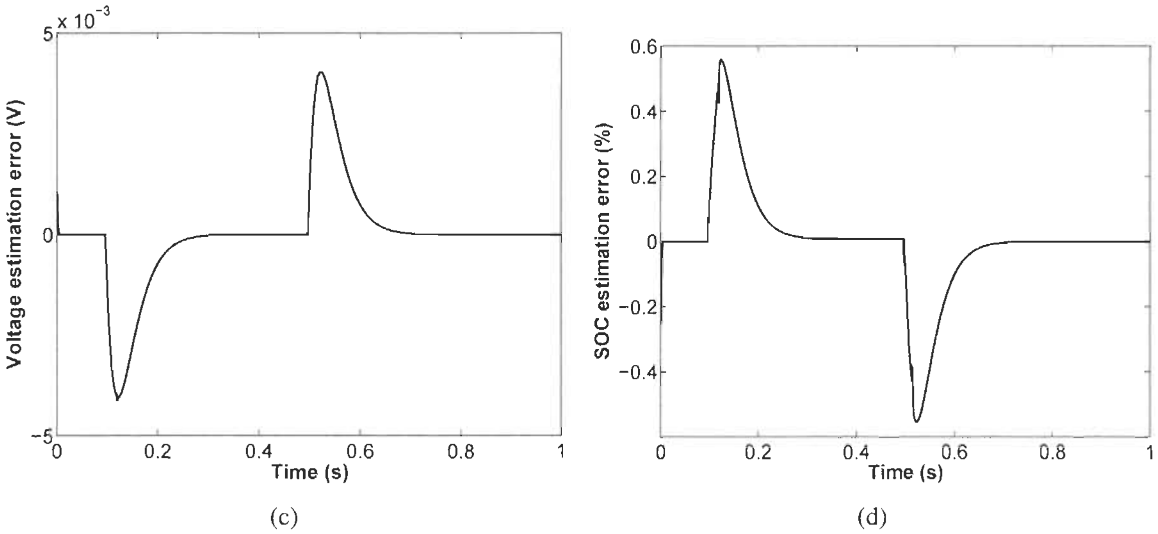


FIGURE 8.18 – Observer-based SOC response with nominal parameters : (a) battery voltage V_b ; (b) battery current I_b ; (c) battery voltage estimation error e ; and (d) SOC estimate error.

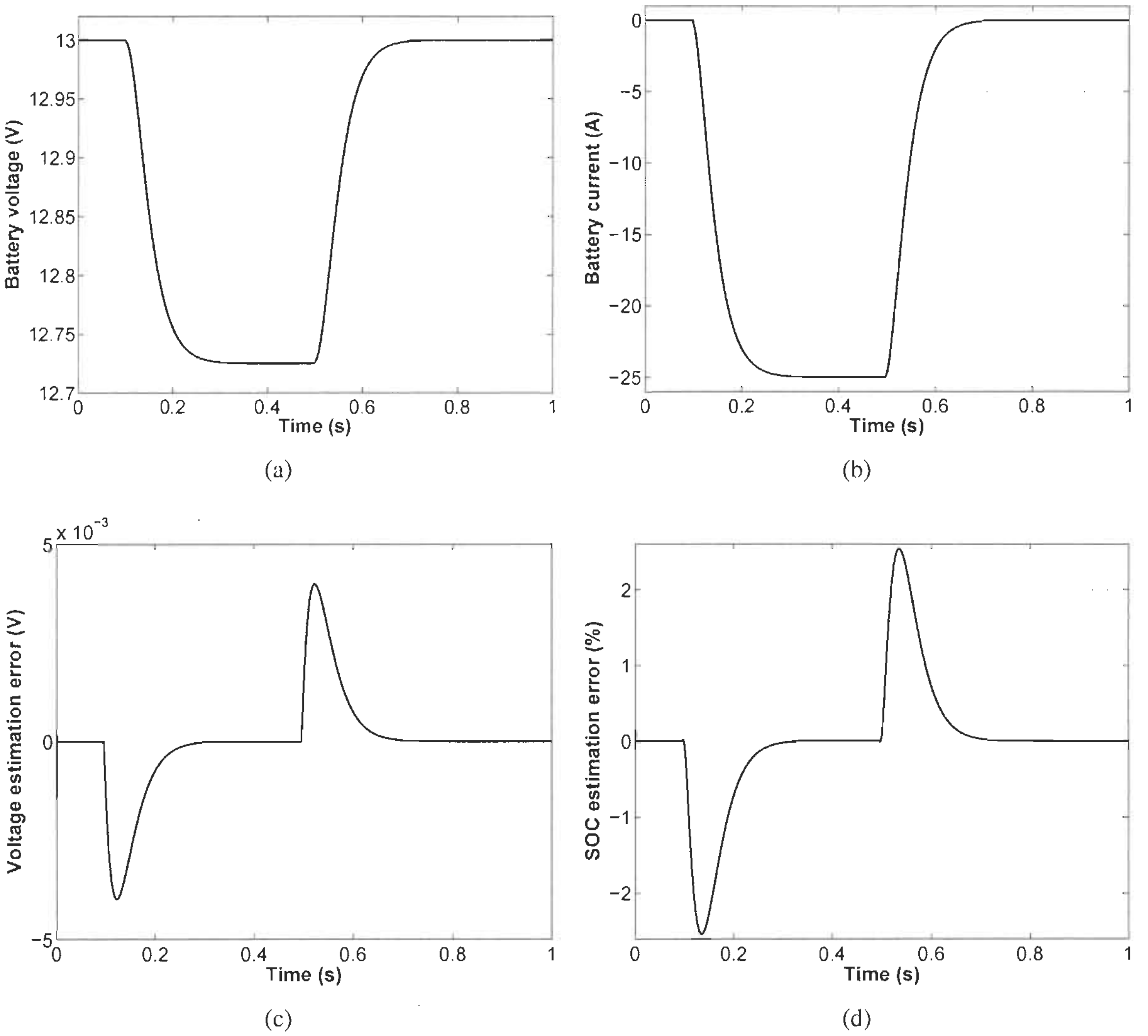


FIGURE 8.19 – Observer-based SOC response with 10 times capacitance : (a) battery voltage V_b ; (b) battery current I_b ; (c) battery voltage estimation error e ; and (d) SOC estimate error.

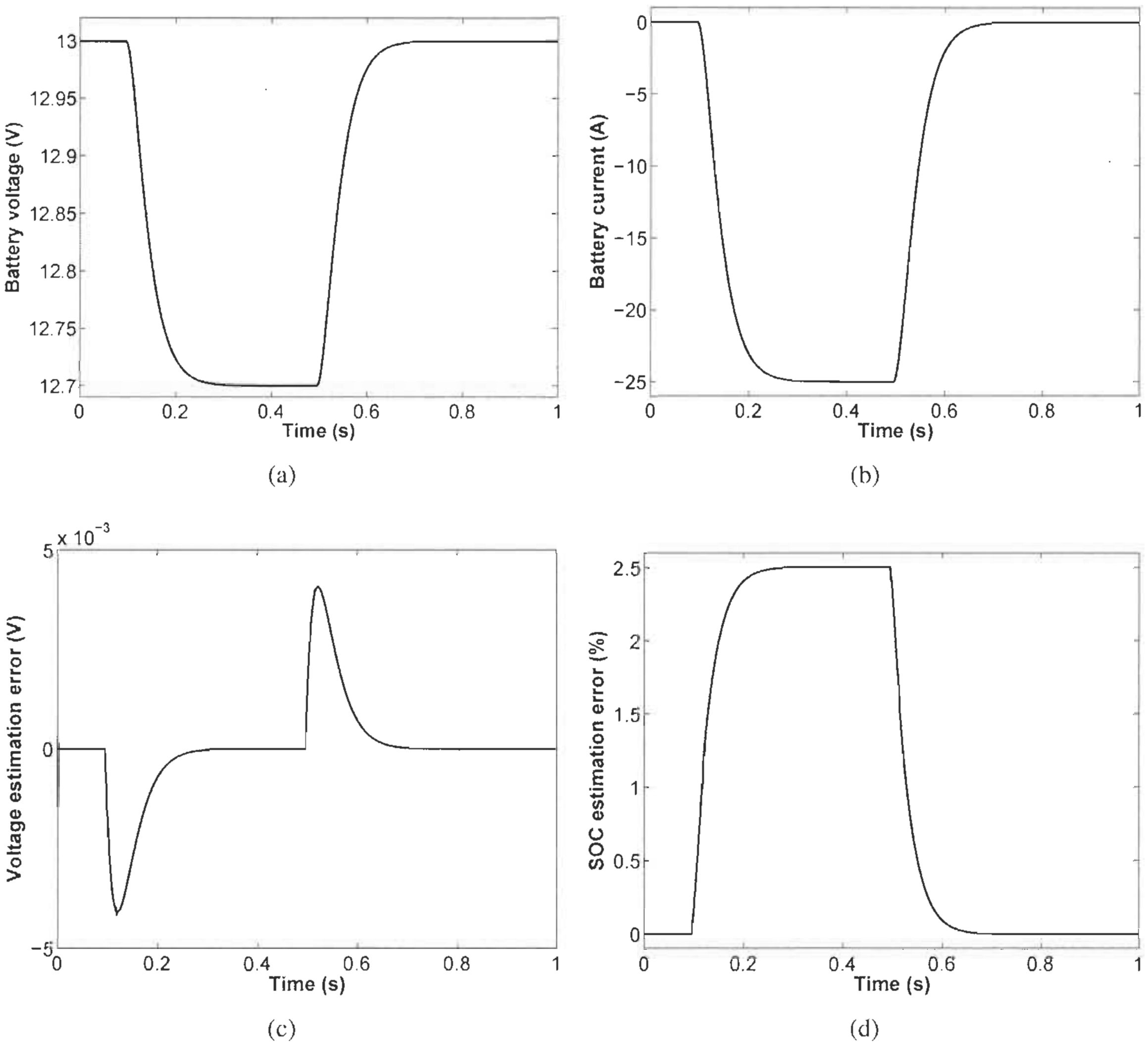


FIGURE 8.20 – Observer-based SOC response with 2 times impedance : (a) battery voltage V_b ; (b) battery current I_b ; (c) battery voltage estimation error e ; and (d) SOC estimate error.

2.5% SOC error). But, this error is likely to become more significant over time as the battery ages. Therefore, an online estimation of the battery's impedance can be considered to cope with aging effects and keep high accurate SOC estimation. Next section presents an adaptive estimation strategy to overcome this problem.

8.6 Adaptive State-of-Charge (SOC) Estimation

Many robust and accurate estimation techniques are available at the cost of high computational complexity. A sliding mode observer has been proposed in [97] to compensate for modeling uncertainties introduced by the use of a simple battery model. On the other hand, an NiMH battery discharge and charge characteristics under different constant currents have been established experimentally in [98]. This model takes into account the influence of temperature on the battery performance. The SOC algorithm is then derived from the experimental data. In [99], the parameters of the battery model are estimated online by an optimization procedure using measured current/voltage profiles. This way, the model is able to capture the relevant battery dynamics and to predict the SOC based on voltage estimation. Various neural network models have been applied for the SOC estimation problem, which have led a satisfactory performance [7]. However, despite the success witnessed by neural network-based control systems, they remain incapable of incorporating any human-like expertise already acquired about the dynamics of the system in hand, which is considered one of the main weaknesses of such soft computing methodologies. In [100], a fuzzy neural network (FNN) has been proposed to overcome this weakness. These techniques are among the intelligent management systems that can monitor the SOC and reduce gradually the load to prevent continuous operation at a low state of charge.

The goal is to estimate V_{oc}, which leads to SOC estimation, with all parameters R_c, R_d, R_b, and C assumed not to be known *a priori* and V_p not measurable. The system's measurable states are the battery voltage V_b and current I_b. The current I_b is taken as positive in discharge mode and negative otherwise.

Assumption 5 *The battery voltage V_b and current I_b along with their derivatives $\dot{V}_b$ and $\dot{I}_b$ are continuous and bounded.*

Assumption 6 *The open circuit voltage V_{oc} is a slowly time varying signal such that, $\dot{V}_{oc} \approx 0$.*

Assumption 7 *V_b and i_b are persistently excited.*

Let $e = \hat{V}_b - V_b$ denotes the battery voltage estimation error and define the following reference model as :

$$s = e + \psi \int e = V_r - V_b \tag{8.4}$$

where ψ being a positive constant and $V_r = \hat{V}_b + \psi \int e$.

Recall the formulation (8.3),

$$\dot{V}_b + R_b \dot{I}_b = -\frac{1}{RC} V_b - \frac{R_b}{RC} I_b + \frac{1}{RC} V_{oc} - \frac{1}{C} I_b$$

Multiplying by RC yields,

$$-RC\dot{V}_b - RCR_b\dot{I}_b - (R+R_b)I_b + V_{oc} = V_b \tag{8.5}$$

Based on this model, consider the following linear regression :

$$-RC\dot{V}_r - RCR_b\dot{I}_b - (R+R_b)I_b + V_{oc} = \Phi^T W \tag{8.6}$$

where $\Phi = [\dot{V}_r, \dot{I}_b, I_b, 1] \in \mathbb{R}^4$ is a vector of known functions (regressor), and $W \in \mathbb{R}^4$ is a vector of parameters :

$$W_1 = -RC$$

$$W_2 = -RCR_b$$

$$W_3 = -(R+R_b)$$

$$W_4 = V_{oc}$$

Since the open-circuit voltage (OCV) is directly correlated with the state of charge of the battery, Φ_4 is therefore used to estimate the state of charge. The estimated battery voltage is computed as follows :

$$V_b = \Phi^T \hat{W} - K_d\ s \tag{8.7}$$

Theorem 18 *Consider a nonlinear system in the form (8.2) with the estimation law (8.7), the following adaptation law guarantees the estimation error asymptotic stability and convergence to zero :*

$$\dot{\hat{W}} = -\Gamma\ \Phi\ s$$

where $\Gamma = [\gamma_1, \gamma_2, \dots, \gamma_4]$ and γ_i is a positive constant.

Proof 12 *Choose the following Lyapunov candidate :*

$$V = \frac{1}{2}\{s^T RCs + \tilde{W}^T \Gamma^{-1} \tilde{W}\}$$

Taking the derivative of V yields :

$$\dot{V} = s^T RC\dot{s} + \tilde{W}^T \Gamma^{-1} \dot{\hat{W}} \tag{8.8}$$

$\dot{\tilde{W}} = \dot{\hat{W}}$ *since parameters W are constant, considering Assumption 6.*
Take the time derivative of (8.4) :

$$\dot{s} = \dot{V}_r - \dot{V}_b$$

Multiplying both sides by RC yields :

$$RC\dot{s} = RC\dot{V}_r - RC\dot{V}_b$$

Substitute $RC\dot{V}_b$ from (8.5) :

$$RC\dot{s} = RC\dot{V}_r + RCR_b\dot{I}_b + (R + R_b)I_b - V_{oc} + V_b$$

Using (8.6) yields :

$$RC\dot{s} = V_b - \Phi^T W$$

Substitute V_b *from (8.7) :*

$$RC\dot{s} = \Phi^T \tilde{W} - K_d s \tag{8.9}$$

where, $\tilde{W} = \hat{W} - W$.

Substitute $RC\dot{s}$ *from (8.9) into (8.8) :*

$$\dot{V} = s^T \Phi^T \tilde{W} + \tilde{W}^T \Gamma^{-1} \dot{\hat{W}} - s^T K_d s$$

Setting the adaptation law as

$$\dot{\hat{W}} = -\Gamma \, \Phi \, s$$

leads to

$$\dot{V} = -s^T K_d s \leq 0$$

This shows that a positive Lyapunov function V*, which is decreasing* ($\dot{V} \leq 0$)*, must converge to a finite limit. Hence, from the Lyapunov function definition, the signals* s*,* $\tilde{W}$*, and* $\hat{W}$ *are also bounded and converge to finite values. It implies from (8.4) that* e *and* $\int e$*, and so,* $\hat{V}_b$*,* V_r*, and* $\dot{V}_r$ *are bounded. It follows from (8.7) that* V_b *is bounded, which implies from (8.9) that* $\dot{s}$ *is also bounded.*

Take the derivative of $\dot{V}$ *:*

$$\ddot{V} = -2s^T K_D \dot{s}$$

Therefore, $\ddot{V}$ *is also bounded.*

From Lemma 1 (Barbalat), V *has a finite limit as* $t \to \infty$ *and* $\dot{V}$ *is uniformly continuous shows that* $\lim_{t\to\infty} \dot{V} = 0$*, and hence,* $\lim_{t\to\infty} s = 0$*. On the other hand,* $\dot{e}$ *is also bounded since* $\dot{s}$ *is bounded. From Lemma 1 (Barbalat),* $\int e$ *has a finite limit as* $t \to \infty$ *and* e *is uniformly continuous shows that* $\lim_{t\to\infty} e = 0$*. Therefore,* $\lim_{t\to\infty} \hat{V}_b = V_b$*. Therefore, the system is asymptotically stable in the sense of Lyapunov.*

8.6.1 Setup

To demonstrate the performance of the proposed adaptive estimator, a buck converter, shown in Fig. 8.21, is designed to provide the proposed adaptive estimation scheme with a continuous and a persistently excited regressor vector. The converter is controlled using a conventional PI controller as shown in Fig. 8.22. The system's model is implemented using SimPowerSystems Simulink toolbox in Matlab and the switching frequency is set to 1 *KHz*. Table 8.2 summarizes the converter's parameters along with its respective values. Parameters R_c, R_d, R_b, R, C, C_v, L, and V_{oc} are assumed not to be known *a priori* and constant with respect to time. The parameters estimate vector is initialized to (0,0,0,12.5). Setting the initial open circuit voltage V_{oc} to 12.5V, which corresponds to 50% SOC, enables the system to start at most at half way from the desired unknown value of V_{oc}.

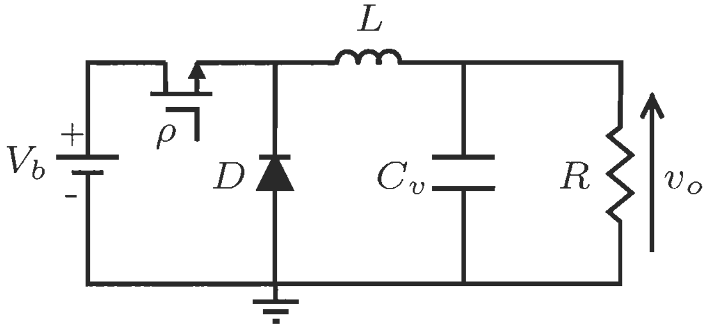


FIGURE 8.21 – Equivalent circuit of a buck converter

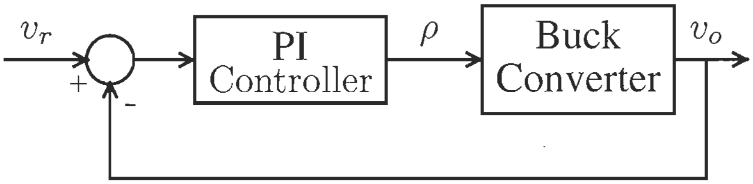


FIGURE 8.22 – Converter's control scheme

TABLE 8.2 – Buck converter's parameters

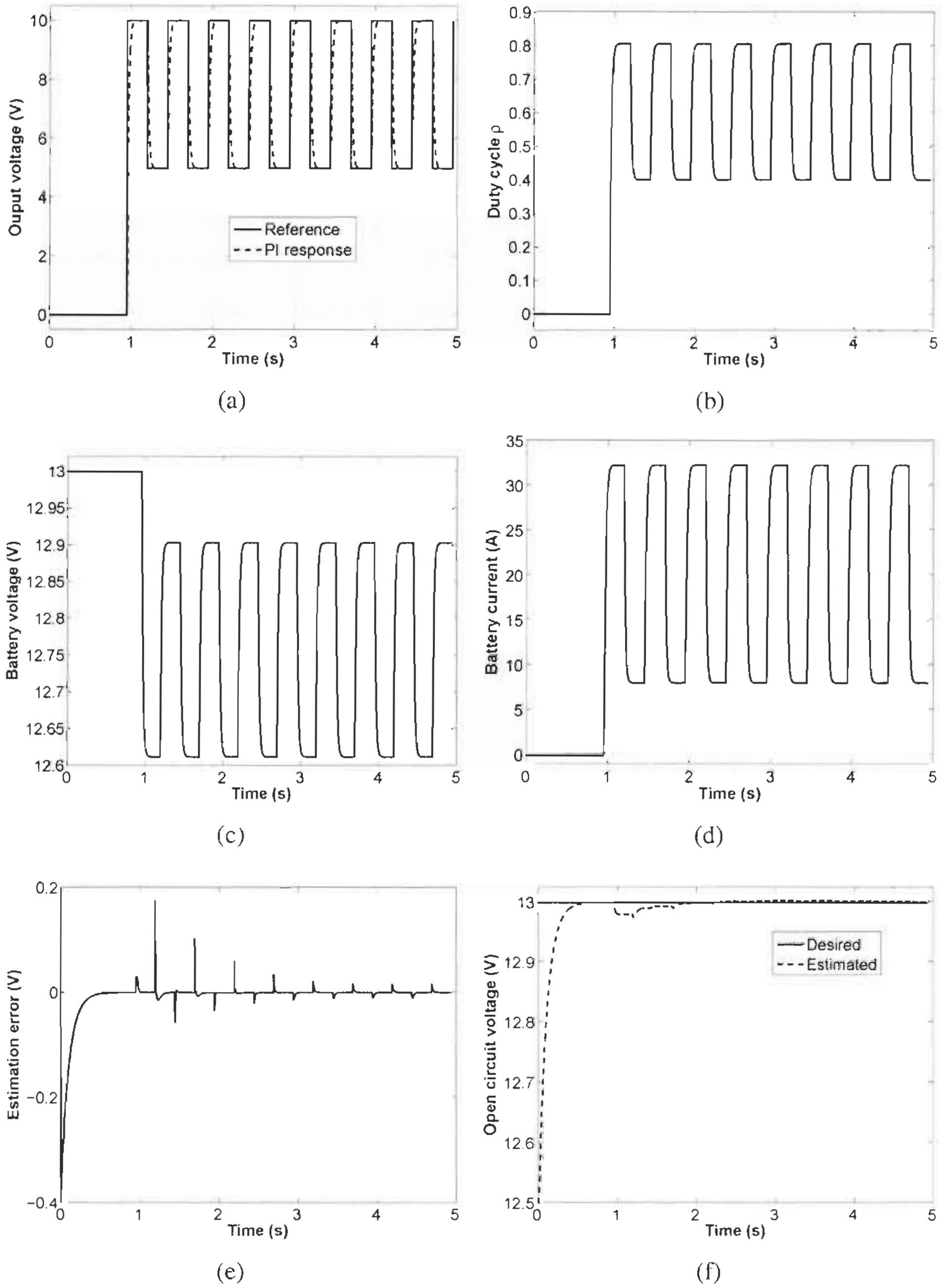
Parameter	Value	Unit
Inductance	$L = 2 \cdot 10^{-3}$	(H)
Capacitor	$C_v = 4 \cdot 10^{-2}$	(F)
Load resistance	$R = 5 \cdot 10^{-1}$	(Ω)

8.6.2 Results

A computer simulation is carried out to study the proposed estimator's performance. The system's response is studied taking into account the converter output voltage V_o, duty cycle ρ, battery voltage V_b and current I_b, the battery voltage estimation error e, and the open circuit voltage estimate $W_4 \approx V_{oc}$. The aforementioned nominal values are used to simulate the system's dynamics. The simulation is conducted for time period of $t = 5$ sec, where the system is left at equilibrium state for $0 \leq t < 1$ sec, and the converter operates for $1 \leq t < 5$ sec. As shown in Fig. 8.23, a fast tracking is achieved at equilibrium state ($W_4 \approx V_{oc} = V_b$). When the converter starts operating, the estimation and tracking error reach their maximum values (around 2% SOC error) since the parameters estimate initial values are still at zero, which causes a deviation of W_4 from its target. These errors decay gradually to a very negligible magnitude for the rest of the simulation, which yields high accurate SOC estimation.

8.7 Fuzzy Supervisory Energy Management for Multisource Electric Vehicles

Brand new battery units from the same batch are known to exhibit capacity variations. These discrepancies are mainly due to the manufacturing process and used materials. Moreover, these battery units cannot be installed physically at the same distance from the load. Therefore, the impedance of physical links between them and the load cannot be ignored. In other words, an optimal operation for a given battery unit is not necessarily optimal for the rest of the battery network. Henceforth, every unit should have its own operation rate parameter. Moreover every parameter should be allowed to vary at each iteration since the error typically



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 8.23 – Adaptive SOC response : (a) converter voltage V_o ; (b) duty cycle ρ ; (c) battery voltage V_b ; (d) battery current I_b ; (e) estimation error e ; and (f) open circuit voltage estimate $W_4 \approx V_{oc}$.

behaves differently along different operating regions of an energy source network. It is noteworthy that the use of a different time-varying operation rate parameter for each energy source unit in accordance to this approach modifies the standard energy management systems in a fundamental way.

Various energy management strategies have been presented over the years. These approaches can be either optimization-based techniques such as optimal control [101] and dynamic programming, or rules-based management techniques such as fuzzy logic and neural networks [102, 103]. This category outperforms the first one since no *a priori* knowledge of the power profile is required. The application of particle swarm optimization, which is a biologically inspired direct search method, is also applied in [104] for real-time optimal energy management. Multiagent-system technology is used in [105] for hybrid distributed energy management systems. Moreover, the integration of multiple energy sources in power systems has been studied in [106, 107]. The fast dynamics of supercapacitors is then combined with the higher batteries energy density to meet the dynamic performance of hybrid electric vehicles. This performance is further improved with a power-flow management algorithm for appropriate control of each of the energy sources. On the other hand, a fuzzy-logic approach is presented in [102] to the energy management of an embedded fuel-cell system while neural networks are proposed in [103] as an efficient energy management system for hybrid electric vehicles. The system minimizes the energy requirement of vehicles working with different power sources like fuel cells, microturbines, or batteries.

The aim of the proposed supervisory approach is efficient energy utilization. In other words, all battery units should keep the same state of charge in all operation ranges. For that, not all units should share the load and the regenerative brake energy equally. Hence, each unit should be charged and discharged with its own rate. Therefore, maintaining an optimal battery units operation throughout their life cycle results in a reduction of their frequent maintenance and an extension of their life span. Let $e = V_{DC} - V_{DC}^*$ denotes the DC bus voltage tracking error, where V_{DC}^* is the DC bus voltage reference signal. This error signal is then distributed through the n energy source units such that each unit has its own error signal e_i, which can be expressed in a weighted fashion as,

$$e_i = k_i\, e \tag{8.10}$$

where $k_i \geq 0$ is the operation rate parameter of the i^{th} energy source ($i = 1 \ldots n$) with the constraint of $\sum_{i=1}^{n} k_i = 1$. As such, each energy source contributes to the minimization of the tracking error e according to its characteristics by its operation rate k_i, which is computed based on a SOC equilibrium criterion using a fuzzy logic supervisor. Therefore, the operation rate parameter k_i is used in discharge mode. However, since an opposite behavior is expected from the supervisor in charge mode, a different operation rate parameter $\bar{k}_i$ for this mode is computed as,

$$\bar{k}_i = \frac{1-k_i}{n-1} \qquad \forall\, n > 1$$

where $\bar{k}_i \geq 0$ takes only positive values and is set to zero for negative values.

The fuzzy control strategy is based on a human operator experience to interpret a situation and initiate an appropriate action. A block diagram for the fuzzy logic supervisor is illustrated in Fig. 8.24. The linguistic terms used for the input membership functions are labeled as *"S (Small)", "M (Medium)", "L (Large)"*. To show the main idea behind the supervisor, let us take the case of two identical units (i.e., $n = 2$). Therefore, the supervisor takes two inputs SOC_1 and SOC_2 and provides operation rate parameters k_1 and k_2 according to the fuzzy logic rules depicted in Table 8.3. The output membership functions are chosen to satisfy the following condition, $\sum_{i=1}^{n} k_i = 1$. This way, the supervisor forces overtime energy source units to operate on the nominal surface $k_1 = 0.5$ and $k_2 = 0.5$ (i.e., diagonal line in Table 8.3). It is noteworthy that the operation rate parameter $\bar{k}_i$ for $n = 2$ reaches the same equilibrium, i.e., nominal surface $k_i = \bar{k}_i = 0.5$. The main idea is : (i) when the SOCs are far from their respective nominal surface, then the supervisor's distribution parameter vector k_i assumes a high value for high SOC units and a low value for low SOC units ; (ii) when the SOCs are approaching the nominal surface, the difference of distribution parameters is adjusted to a smaller value for a smoother approach ; (iii) once the SOCs are on the nominal surface, then the difference of distribution parameters is set to zero. In this work, without loss of generality, we assume that $n = 3$. The same principle applies for n energy source units. Therefore, given the designed input signals SOC_1, SOC_2, and SOC_3, the fuzzy logic supervisor provides operation rate parameters k_1, k_2, and k_3. The output membership functions are labeled as *"Z (Zero)"*, *"VS (Very Small)"*, *"MS (Medium Small)"*, *"S (Small)"*, *"LS (Large Small)"*, *"M (Medium)"*, *"L (Large)"*, *"VL (Very Large)"*. An empirical analysis for the parameters of membership functions is performed to improve the supervisor's performance and satisfy the condition, $\sum_{i=1}^{n} k_i = 1$. In this case, both modes also have the same equilibrium, i.e., nominal surface $k_i = \bar{k}_i = \frac{1}{3}$. The fuzzy rules for $n = 3$ are depicted in Table 8.4, 8.5, and 8.6. The input membership functions used in this case are shown in Fig. 8.25.

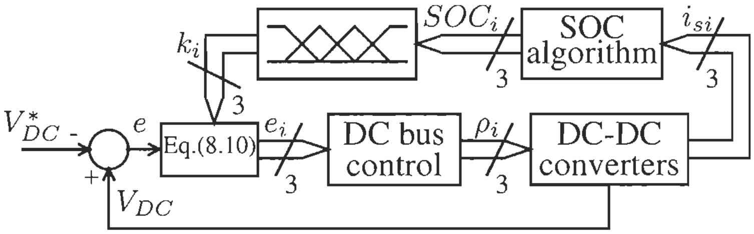


FIGURE 8.24 – Block diagram of the fuzzy supervisory energy management scheme.

TABLE 8.3 – Fuzzy Logic Rules for $n = 2$

		SOC_1		
	k_1/k_2	S	M	L
SOC_2	S	0.5/0.5	0.75/0.25	1/0
	M	0.25/0.75	0.5/0.5	0.75/0.25
	L	0/1	0.25/0.75	0.5/0.5

8.7.1 Setup

To demonstrate the performance of the proposed energy management system, a set of computer simulation runs is carried out on a three identical bidirectional DC-DC converters like the one depicted in Fig. 8.5. The inductance and capacitance are set, respectively, to

TABLE 8.4 – Fuzzy Logic Rules for $n = 3$ and $SOC_3 =$ "S"

	SOC_1		
SOC_2	S	M	L
S	0.33/0.33/0.33	0.5/0.25/0.25	0.6/0.2/0.2
M	0.25/0.5/0.25	0.4/0.4/0.2	0.5/0.3/0.2
L	0.2/0.6/0.2	0.3/0.5/0.2	0.4/0.4/0.2

TABLE 8.5 – Fuzzy Logic Rules for $n = 3$ and $SOC_3 =$ "M"

	SOC_1		
SOC_2	S	M	L
S	0.25/0.25/0.5	0.4/0.2/0.4	0.5/0.2/0.3
M	0.2/0.4/0.4	0.33/0.33/0.33	0.5/0.25/0.25
L	0.2/0.5/0.3	0.25/0.5/0.25	0.4/0.4/0.2

TABLE 8.6 – Fuzzy Logic Rules for $n = 3$ and $SOC_3 =$ "L"

	SOC_1		
SOC_2	S	M	L
S	0.2/0.2/0.6	0.3/0.2/0.5	0.4/0.2/0.4
M	0.2/0.3/0.5	0.25/0.25/0.5	0.4/0.2/0.4
L	0.2/0.4/0.4	0.2/0.4/0.4	0.33/0.33/0.33

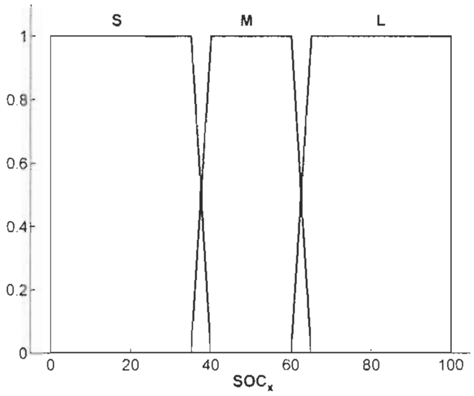


FIGURE 8.25 – Input fuzzy membership functions for $n = 3$.

$L = 250\mu$H and $C = 1$mF. The DC bus reference voltage is set to $V_{DC}^{*} = 400$V and the switching and sampling frequencies are both set to 5 kHz. The system is subjected to a ±15kW periodic square power demand with a period of 1 minute to enable the energy units to operate in both charge and discharge modes. The Coulomb counting method (Amphour (Ah) balancing method) is used to estimate the batteries SOC and since it is a slowly time-varying process, the system is simulated for 15 minutes. The system's performance metrics are the DC bus voltage V_{DC}, the source currents i_s of the three converters, their duty cycles ρ, state of charge SOC, and the operation rate parameters.

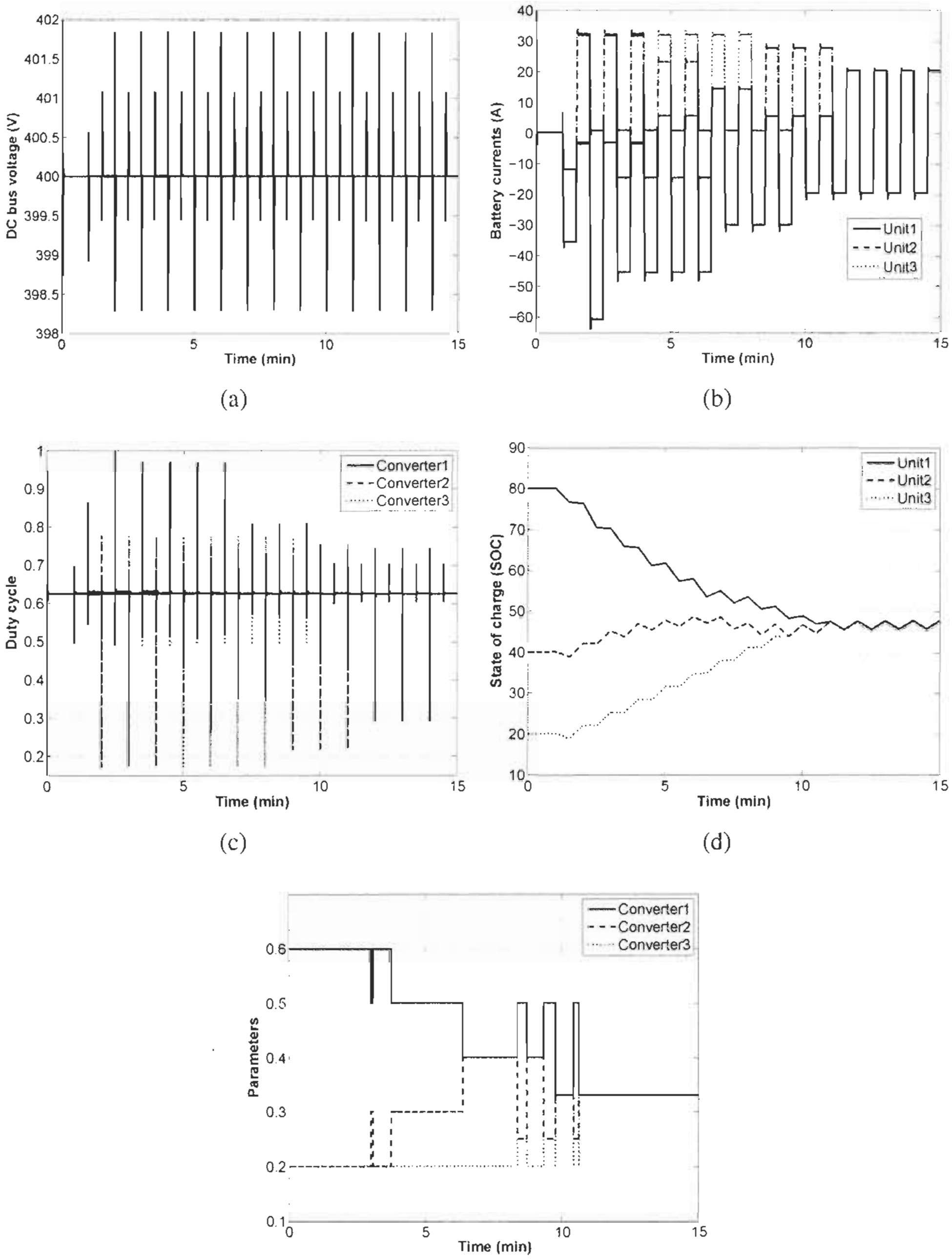
8.7.2 Results

A numerical simulation is carried out to study the proposed energy management system's performance. In this simulation, the aforementioned nominal values are used to simulate the system's dynamics and the inverter current is taken as the step response of a critically damped second order system with a natural frequency of 100 rad/s. The initial battery units state of charge is set, respectively, to 20%, 40%, and 80%. In order to study the system's performance in equilibrium, no power demand is requested from the system for the first minute. The results are depicted in Fig. 8.26. As expected, no current is flowing from or to any battery unit and the SOC of all battery units remain at their initial values. This observation is important since transferring energy from a given battery unit to another will result in a loss of energy and hence an inefficient operation of the overall system. When the system is subjected to the ±15kW square power demand, which corresponds to ±20A current from each battery unit, the DC bus voltage tracking error is kept in an acceptable range. Moreover, the fuzzy logic supervisor assigns different operation rate parameter to each battery unit depending on its SOC, which yields different battery unit currents as it is shown in Fig. 8.26(b). On the other hand, the difference between the SOC of all battery units (Fig. 8.26(d)) decreases over time and finally converges to zero. Hence, the fuzzy logic supervisor assigns a operation rate parameter of a $\frac{1}{3}$ letting the three battery units operate at the rate, which yields three square wave currents of ±20A.

8.8 Conclusion

In this chapter, an adaptive fuzzy logic controller is proposed for a DC-DC boost converter under large parametric and load uncertainties. The control strategy aims to achieve accurate voltage tracking with unknown dynamics, high parameter and load variations, and no current sensing. Therefore, robustness to uncertainties with large magnitudes is achieved without the inner current control loop, which reduces the number of sensors. The controller is also compared to a PI controller in similar operating conditions. Simulation results showed the superiority of the adaptive FLC in compensating for higher magnitude of uncertainties.

On the other hand, an adaptive DC bus voltage control strategy is proposed for renewable energy systems. The proposed adaptive technique achieves robust bidirectional DC-DC converters control with no source voltage sensing, which yields less DC bus voltage variations. Thus, accurate DC bus voltage tracking enables high efficient power transfer to the grid. The controller's performance is compared to a classical PI controller. Simulation results show better tracking using less control effort with the adaptive controller. Results also show its high tolerance to uncertainties unlike with PI controllers. Furthermore, the performance of the proposed strategy is a key to achieve the high efficiency needed for high performance power energy



(a)

(b)

(c)

(d)

(e)

FIGURE 8.26 – Fuzzy supervisor response under ±15kW square power demand : (a) DC bus voltage V_{DC} ; (b) battery currents i_{si} ; (c) duty cycle ρ_i ; (d) state of charge SOC_i ; and (e) parameters k_i.

systems.

Next, an observer-based state of charge (SOC) estimator was introduced for batteries. The estimation technique showed that the SOC can be determined with high accuracy based on only the measurements of the battery voltage and current. A stability analysis guarantees the convergence and stability of the proposed strategy. Simulation results also high SOC estimation accuracy in the presence of parameter variations. The battery aging effect is also verified. Furthermore, the proposed estimator is easier to implement as opposed to other estimation techniques with similar performance, such as intelligent-based battery management systems.

Then, an adaptive state of charge (SOC) estimator is introduced for batteries. The online parameters estimation technique showed that the SOC can be determined with high accuracy based on only the measurements of the battery voltage and current. A Lyapunov-based stability analysis guarantees the convergence and stability of the proposed strategy. It is easier to be implemented as opposed to other estimation approaches with similar performance, such as intelligent-based battery management systems. A computer simulation highlights the performance of the proposed estimator in determining the SOC with a very high accuracy. It is noteworthy that although parameters estimation needs persistent excitation in many adaptive systems, the fact the SOC can be estimated in a battery's equilibrium state makes the SOC estimator independent of this requirement.

Finally, a fuzzy supervisory energy management strategy is introduced for multisource electric vehicles. The proposed technique achieves energy utilization equilibrium for different battery units operating through multiple bidirectional DC-DC converters. This extends the life span of energy storage devices and reduces their frequent maintenance. Furthermore, the performance of the proposed energy management strategy is a key to achieve the high efficiency needed for high performance power energy systems such as electric vehicles. Generally, when a single battery unit's performance degrades, all units are replaced at the same time to preserve the overall system's integrity. The proposed energy management method enables the use of multiple units with different capacity and state of charge. Moreover, it also allows merging batteries from different manufacturers.

Chapitre 9

Conclusion

In this thesis, new soft-computing based adaptive control schemes have been proposed for complex nonlinear systems. The proposed controllers are designed to achieve stability and robustness in the presence of both structured and unstructured uncertainties. As a starting point, we showed the learning and approximation capabilities of artificial neural networks in an identification strategy design for spacecraft formation flying. ANNs ability to learn from *a priori* unknown nonlinear systems has been used to approximate the nonlinear behavior in thrusters dynamics. This has been used as a main motivation behind the use of such technique for control design of complex nonlinear systems.

Chapter 5 presents Lyapunov-based adaptation strategies for neural networks and fuzzy systems. These learning mechanisms are designed for general MIMO systems and contribute to the stability analysis of soft-computing techniques. An application to the well-known inverted pendulum problem has been proposed in [108, 109]. Moreover, the ANN Lyapunov-based adaptation technique has also been applied to piezoelectric actuators. The proposed approach achieves good performance with hysteresis compensation, which is a common issue in many microelectromechanical systems (MEMS). The results were published in [110].

In chapter 6, the difficulty of adaptive control theory to cope with unstructured uncertainties was shown. This weakness has been addressed by proposing a Lyapunov-based adaptive friction compensator. Comparison against a popular friction compensation technique shows better results when friction is magnified. Since disturbance is also an important issue, an ANN-based disturbance compensation strategy has been developed to cope with friction, disturbance and flexibility. Results show robustness to all these kinds of unstructured uncertainties. On the other hand, type-1 fuzzy logic is compared to its type-2 counterpart for flexible-joint manipulators subjected to uncertainties of various magnitudes. Simulation results show the superiority of type-2 FLC in compensating for uncertainties of higher magnitude. This work is one of the scarce attempts in type-2 fuzzy logic control of flexible-joint manipulators.

In chapter 7, several speed control and estimation strategies have been proposed for high performance PMSM drives. The ability of adaptive controllers to cope with parametric uncertainties has been shown by proposing few adaptive control structures with a speed and disturbance observer. These strategies have been simplified and an adaptive controller is proposed with uncertain dynamics. The stability of all these approaches is proven by Lyapunov stability theory. Therefore, this result is extended to prove the stability of many soft-computing based adaptive control strategies. Moreover, an ANN-based speed observer has been proposed in [111]. The stability of the closed-loop control system (controllers + observer) is guaranteed by Lyapunov. Results published in [112] show high tracking accuracy and tolerance to unstructured uncertainties of different magnitudes. However, the speed estimation shows sensitivity to flux variations. Therefore, a nonlinear ANN-based speed observer has been proposed to solve the sensitivity problem to parameter variations. The results were published in [113]. Finally, robustness to both structured and unstructured uncertainties has been achieved with a single adaptive fuzzy controller. Unlike other control techniques, no *a priori* offline training, weights initialization, mechanical/electrical parameters knowledge, voltage or current transducer is required. However, the efficiency is expected to be lower than in Maximum Torque Per Ampere (MTPA) technique since no current loop regulation is used. Nevertheless, this work is of one of the first attempts, if any, to achieve high tracking control performance in the presence of both structured and unstructured uncertainties without the use of currents sensing or loop regulation for PMSMs. The proposed approach has been published in [114]. In this chapter, the proposed controllers achieve a tracking precision within sensors resolution. Therefore, future work might consider the effect of low precision sensing and noise filtering on the control performance. This is an important step towards experimental validation of these controllers.

In chapter 8, advanced adaptive techniques have been presented in chapter 8 for energy production systems. An adaptive fuzzy control of DC-DC boost converters copes with large parametric and load uncertainties. The control strategy achieves robustness to uncertainties without the inner current control loop. Comparison results against a PI controller were published in [115] and show better performance in the presence of uncertainties. Similar performance is achieved with less complexity in [116]. This proposed adaptive DC bus voltage controller uses a single adaptive parameter to achieve high efficient power flow to the grid. Moreover, no source voltage sensing is required, which reduces the number of sensors. Comparison against a PI-based control structure is carried out. Results show higher accuracy and robustness of the proposed adaptive control technique. On the other hand, the state of charge estimation of batteries is proposed in [117] using observer-based theory. This technique uses only the measurements of the battery voltage and current. Moreover, a stability analysis guarantees its convergence and stability. Furthermore, it is easier to implement as opposed to other estimation techniques with similar performance. However, the battery aging effect results in an estimation error, which is likely to become significant over time as the battery ages. Therefore, an adaptive estimation strategy is introduced in [118] to overcome this problem. Results show high estimation accuracy. Furthermore, the stability of the proposed estimation technique is guaranteed by Lyapunov stability theory. Finally, a fuzzy supervisory energy management strategy is introduced in [119] for multisource electric vehicles. This technique extends the life span of energy storage devices and reduces their frequent maintenance, which yields better energy utilization. Results show accurate equilibrium for battery units with different capacity and state of charge.

This thesis contributes to the design of stable adaptive control strategies using soft-computing techniques. In continuation to this work, new control structures can be considered to reduce further the complexity of soft-computing tools. As far as neural networks are concerned, this thesis presents Lyapunov-based learning strategies for the most popular ANN, which is the MLP.

Henceforth, new stable adaptive control strategies can be considered for the so many ANN architectures. Moreover, new stable adaptation laws can also be developed for soft-computing optimization based techniques, such as genetic algorithms and ant colony.

Bibliographie

[1] K. Pathak, J. Franch, and S. Agrawal, "Velocity and position control of a wheeled inverted pendulum by partial feedback linearization," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 505–513, Jun. 2005.

[2] Z. Li and J. Luo, "Adaptive robust dynamic balance and motion controls of mobile wheeled inverted pendulums," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 1, pp. 233–241, Jan. 2009.

[3] T. Batzel and K. Lee, "An approach to sensorless operation of the permanent-magnet synchronous motor using diagonally recurrent neural networks," *IEEE Transactions on Energy Conversion*, vol. 18, no. 1, pp. 100–106, Mar. 2003.

[4] C.-H. Chen, C.-J. Lin, and C.-T. Lin, "Nonlinear system control using adaptive neural fuzzy networks based on a modified differential evolution," *IEEE Transactions on Systems, Man, and Cybernetics, Part C : Applications and Reviews*, vol. 39, no. 4, pp. 459–473, Jul. 2009.

[5] S. Ge and I. Postlethwaite, "Adaptive neural network controller design for flexible joint robots using singular perturbation technique," *Transactions of the Institute of Measurement and Control*, vol. 17, no. 3, pp. 120–131, 1995.

[6] H. Hagras, "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, pp. 524–539, August 2004.

[7] G. Hongyu, J. Jiuchun, and W. Zhanguo, "Estimating the state of charge for Ni-MH battery in HEV by RBF neural network," in *IEEE International Workshop on Intelligent Systems and Applications, ISA 2009*, Wuhan, China, May 2009, pp. 1–4.

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

[9] F.-J. Lin and P.-H. Chou, "Adaptive control of two-axis motion control system using interval type-2 fuzzy neural network," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 1, pp. 178–193, Jan. 2009.

[10] L. X. Wang, *Adaptive Fuzzy Systems and Control : Design and Stability Analysis*. PTR Prentice Hall, 1994.

[11] F.-J. Lin and C.-H. Lin, "A permanent-magnet synchronous motor servo drive using self-constructing fuzzy neural network controller," *IEEE Transactions on Energy Conversion*, vol. 19, no. 1, pp. 66–72, Mar. 2004.

[12] C.-W. Park, "Robust stable fuzzy control via fuzzy modeling and feedback linearization with its applications to controlling uncertain single-link flexible joint manipulators," *Journal of Intelligent and Robotic Systems : Theory and Applications*, vol. 39, no. 2, pp. 131–147, February 2004.

[13] S. Jung and S. Kim, "Control experiment of a wheel-driven mobile inverted pendulum using neural network," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 297–303, Mar. 2008.

[14] C. Tao, J. Taur, T. Hsieh, and C. Tsai, "Design of a fuzzy controller with fuzzy swing-up and parallel distributed pole assignment schemes for an inverted pendulum and cart system," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1277–1288, Nov. 2008.

[15] J. Lee, Y. Kim, and A. Ng, "Formation flying test-bed development : Modeling, control and simulations," Space Technologies, Canadian Space Agency (CSA), Tech. Rep., March 2006.

[16] J. Lee, S. Mondor, Y. Kim, and A. Ng, "Thrust calibration of flow control valves in formation flying test-bed," Space Technologies, Canadian Space Agency (CSA), Tech. Rep., July 2006.

[17] H. Chaoui, P. Sicard, and W. Gueaieb, "ANN-Based Adaptive Control of Robotic Manipulators with Friction and Joint Elasticity," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 8, pp. 3174 – 3187, Aug. 2009.

[18] S. Haykin, *Neural Networks : A Comprehensive Foundation.* Prentice Hall Publishing, 1999.

[19] J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems : Introduction and New Directions.* Prentice-Hall, 2001.

[20] M. Mitchell, *An Introduction to Genetic Algorithms.* The MIT Press, 1998.

[21] M. Dorigo and T. Stutzle, *Ant Colony Optimization.* The MIT Press, 2004.

[22] J. Slotine and W. Li, *Applied Nonlinear Control.* Prentice Hall, 1991.

[23] K. Astrom and B. Wittenmark, *Adaptive Control.* Addison-Wesley, 1995.

[24] M. El-Hawwary, A. Elshafei, H. Emara, and H. Fattah, "Adaptive fuzzy control of the inverted pendulum problem," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 6, pp. 1135–1144, Nov. 2006.

[25] L.-H. Chang and A.-C. Lee, "Design of nonlinear controller for bi-axial inverted pendulum system," *IET Control Theory and Applications*, vol. 1, no. 4, pp. 979–986, Jul. 2007.

[26] S.-J. Huang and C.-L. Huang, "Control of an inverted pendulum using grey prediction model," *IEEE Transactions on Industry Applications*, vol. 36, no. 2, pp. 452–458, Mar.-Apr. 2000.

[27] C.-M. Lin and Y.-J. Mon, "Decoupling control by hierarchical fuzzy sliding-mode controller," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 593–598, Jul. 2005.

[28] Y.-H. Chang, C.-W. Chang, J.-S. Taur, and C.-W. Tao, "Fuzzy swing-up and fuzzy sliding-mode balance control for a planetary-gear-type inverted pendulum," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 9, pp. 3751–3761, Sep. 2009.

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

[30] E. Kim, "Output feedback tracking control of robot manipulators with model uncertainty via adaptive fuzzy logic," *IEEE Transactions Fuzzy Systems*, vol. 12, no. 3, pp. 368–378, Jun. 2004.

[31] H. Chaoui, W. Gueaieb, M. C. Yagoub, and P. Sicard, "Hybrid Neural Fuzzy Sliding Mode Control of Flexible-Joint Manipulators with Unknown Dynamics," in *IEEE Industrial Electronics Conference*, 2006, pp. 4082–4087.

[32] D. Hui, S. Fuchun, and S. Zengqi, "Observer-based adaptive controller design of flexible manipulators using time-delay neuro-fuzzy networks," *Journal of Intelligent and Robotic Systems : Theory and Applications*, vol. 34, no. 4, pp. 453–466, August 2002.

[33] B. Subudhi and A. Morris, "Singular perturbation based neuro-H infinity control scheme for a manipulator with flexible links and joints," *Robotica*, vol. 24, no. 2, pp. 151–161, Mar. 2006.

[34] B. Armstrong and C. C. de Wit, "Friction modeling and compensation," *The Control Handbook*, vol. 77, pp. 1369–1382, 1996.

[35] H. Olsson, K. Astrom, C. D. Wit, M. Gafvert, and P. Lischinsky, "Friction models and friction compensation," *European Journal of Control*, vol. 4, no. 3, pp. 176–195, 1998.

[36] S. Bashash and N. Jalili, "Robust adaptive control of coupled parallel piezo-flexural nanopositioning stages," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 1, pp. 11–20, Feb. 2009.

[37] K. Leang, Q. Zou, and S. Devasia, "Feedforward control of piezoactuators in atomic force microscope systems," *IEEE Control Systems Magazine*, vol. 29, no. 1, pp. 70–82, Feb. 2009.

[38] J. Yi, S. Chang, and Y. Shen, "Disturbance-observer-based hysteresis compensation for piezoelectric actuators," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 4, pp. 456–464, Aug. 2009.

[39] H. Liaw, B. Shirinzadeh, and J. Smith, "Robust neural network motion tracking control of piezoelectric actuation systems for micro/nanomanipulation," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 356–367, Feb. 2009.

[40] S. Huang, K. K. Tan, and T. H. Lee, "Adaptive sliding-mode control of piezoelectric actuators," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3514–3522, Sep. 2009.

[41] D. Seidl, S.-L. Lam, J. Putman, and R. Lorenz, "Neural network compensation of gear backlash hysteresis in position-controlled mechanisms," *IEEE Transactions on Industry Applications*, vol. 31, no. 6, pp. 1475–1483, Nov. 1995.

[42] L. Sweet and M. Good, "Redefinition of the robot motion-control problem," *Control Systems Magazine, IEEE*, vol. 5, no. 3, pp. 18–25, Aug 1985.

[43] F. Ghorbel, J. Hung, and M. Spong, "Adaptive control of flexible-joint manipulators," *Control Systems Magazine, IEEE*, vol. 9, no. 7, pp. 9–13, Dec 1989.

[44] D. Seidl, T. Reineking, and R. Lorenz, "Use of neural networks to identify and compensate for friction in precision, position controlled mechanisms," in *Industry Applications Society Annual Meeting, Conference Record of the 1992, IEEE*, vol. 2, 4-9 Oct 1992, pp. 1937–1944.

[45] M. Benosman and G. L. Vey, "Control of flexible manipulators : A survey," *Robotica*, vol. 22, no. 5, pp. 533–545, September 2004.

[46] M. W. Spong, "Modeling and control of elastic joint robots," *Journal of Dynamic Systems, Measurement and Control, Transactions ASME*, vol. 109, no. 4, pp. 310–319, Dec. 1987.

[47] A. D. Luca, A. Isidori, and F. Nicolo, "Control of robot arm with elastic joints via nonlinear dynamic feedback," in *Proceedings of the IEEE Conference on Decision and Control Including The Symposium on Adaptive Pro*, Fort Lauderdale, FL, USA, 1985, pp. 1671–1679.

[48] K. Khorasani, "Nonlinear feedback control of flexible joint manipulators : A single link case study," *IEEE Transactions on Automatic Control*, vol. 35, no. 10, pp. 1145–1149, Oct. 1990.

[49] C. de Wit, "Robust control for servo-mechanisms under inexact friction compensation," *Automatica*, vol. 29, no. 3, p. 757Ű761, 1993.

[50] C. Ott, A. Albu-Schaffer, and G. Hirzinger, "Comparison of adaptive and nonadaptive tracking control laws for a flexible joint manipulator," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2, Lausanne, Switzerland, Sep. 2002, pp. 2018–2024.

[51] R. Al-Ashoor, R. Patel, and K. Khorasani, "Robust adaptive controller design and stability analysis for flexible-joint manipulators," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 2, pp. 589–602, Mar. 1993.

[52] F. Ghorbel and M. Spong, "Adaptive integral manifold control of flexible joint robot manipulators," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, Nice, France, Apr. 1992, pp. 707–714.

[53] H. Taghirad and M. Khosravi, "Design and simulation of robust composite controllers for flexible joint robots," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 3, Taipei, Taiwan, Sep. 2003, pp. 3108–3113.

[54] L. Huang, S. Ge, and T. Lee, "Adaptive position/force control of an uncertain constrained flexible joint robots - singular perturbation approach," in *Proceedings of the SICE Annual Conference*, Sapporo, Japan, Aug. 2004, pp. 1693–1698.

[55] K. Khorasani, "Adaptive control of flexible joint robots," *IEEE Transactions on Robotic and Automation*, vol. 8, pp. 250–267, Apr 1992.

[56] C. Ott, A. Albu-Schaffer, A. Kugi, S. Stramigioli, and G. Hirzinger, "A passivity based cartesian impedance controller for flexible joint robots - part i : Torque feedback and gravity compensation," in *Proceedings - IEEE International Conference on Robotics and Automation*, no. 3, New Orleans, LA, United States, Apr. 2004, pp. 2659–2665.

[57] L. Tian and A. Goldenberg, "Robust adaptive control of flexible joint robots with joint torque feedback," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, Nagoya, Jpn, May 1995, pp. 1229–1234.

[58] H. Chaoui, P. Sicard, A. Lakhsasi, and H. Schwartz, "Neural network based model reference adaptive control structure for a flexible joint with hard nonlinearities," in *IEEE International Symposium on Industrial Electronics*, vol. 1, Ajaccio, France, May 2004, pp. 271–276.

[59] H. Chaoui, P. Sicard, and A. Lakhsasi, "Reference model supervisory loop for neural network based adaptive control of a flexible joint with hard nonlinearities," in *IEEE Canadian Conference on Electrical and Computer Engineering*, vol. 4, Niagara Falls, Canada, May 2004, pp. 2029–2034.

[60] H. Chaoui and W. Gueaieb, "Type-2 Fuzzy Logic Control of a Flexible-Joint Manipulator," *Journal of Intelligent and Robotic Systems*, vol. 51, no. 2, pp. 159–186, Feb. 2008.

[61] H. Chaoui, W. Gueaieb, and M. C. Yagoub, "Artificial Neural Network Control of a Flexible-Joint Manipulator under Unstructured Dynamic Uncertainties," in *IEEE International Workshop on Robotic and Sensors Environments*, 2007.

[62] W. Pedryez, "Why triangular membership functions ?" *Fuzzy Sets and Systems*, vol. 64, no. 1, pp. 21–30, May 1994.

[63] F. Morel, J.-M. Retif, L.-S. Xuefang, and C. Valentin, "Permanent magnet synchronous machine hybrid torque control," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 2, pp. 501–511, Feb. 2008.

[64] A. Piippo, M. Hinkkanen, and J. Luomi, "Analysis of an adaptive observer for sensorless control of interior permanent magnet synchronous motors," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 2, pp. 570–576, Feb. 2008.

[65] M. F. Rahman, L. Z. E. Haque, and M. A. Rahman, "A direct torque-controlled interior permanent-magnet synchronous motor drive without a speed sensor," *IEEE Transactions on Energy Conversion*, vol. 18, no. 1, pp. 17–22, Mar. 2003.

[66] S. Bolognani, L. Tubiana, and M. Zigliotto, "Extended Kalman filter tuning in sensorless PMSM drives," *IEEE Transactions on Industry Applications*, vol. 39, no. 6, pp. 1741–1747, Nov.-Dec. 2003.

[67] M. Boussak, "Implementation and experimental investigation of sensorless speed control with initial rotor position estimation for interior permanent magnet synchronous motor drive," *IEEE Transactions on Power Electronics*, vol. 20, no. 6, pp. 1413–1422, Nov. 2005.

[68] J. Solsona, M. I. Valla, and C. Muravchik, "Nonlinear control of a permanent magnet synchronous motor with disturbance torque estimation," *IEEE Transactions on Energy Conversion*, vol. 15, no. 2, pp. 163–168, Jun. 2000.

[69] R. Yan, B. Li, and F. Zhou, "Sensorless control of PMSMs based on parameter-optimized MRAS speed observer," in *IEEE International Conference on Automation and Logistics*, Sep. 2008, pp. 1573–1578.

[70] L. Jiaxi, Y. Guijie, and Y. Pengfei, "Rotor position estimation for PMSM based on sliding mode observer," in *IEEE International Conference on Mechatronics and Automation*, Aug. 2007, pp. 3684–3689.

[71] B.-H. Bae, S.-K. Sul, J.-H. Kwon, and J.-S. Byeon, "Implementation of sensorless vector control for super-high-speed PMSM of turbo-compressor," *IEEE Transactions on Industry Applications*, vol. 39, no. 3, pp. 811–818, May-June 2003.

[72] C. French and P. Acarnley, "Direct torque control of permanent magnet drives," *IEEE Transactions on Industry Applications*, vol. 32, no. 5, pp. 1080–1088, Sep.-Oct. 1996.

[73] N. Idris and A. Yatim, "Direct torque control of induction machines with constant switching frequency and reduced torque ripple," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 4, pp. 758–767, Aug. 2004.

[74] G. Buja and M. Kazmierkowski, "Direct torque control of PWM inverter-fed AC motors - a survey," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 4, pp. 744–757, Aug. 2004.

[75] V. Ambrozic, G. Buja, and R. Menis, "Band-constrained technique for direct torque control of induction motor," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 4, pp. 776–784, Aug. 2004.

[76] K. Drobnic, M. Nemec, D. Nedeljkovic, and V. Ambrozic, "Predictive direct control applied to AC drives and active power filter," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1884–1893, Jun. 2009.

[77] S. Bolognani, L. Peretti, and M. Zigliotto, "Design and implementation of model predictive control for electrical motor drives," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1925–1936, Jun. 2009.

[78] H. Jin and J. Lee, "An RMRAC current regulator for permanent-magnet synchronous motor based on statistical model interpretation," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 1, pp. 169–177, Jan. 2009.

[79] H. Ndjana and P. Lautier, "Sensorless vector control of an IPMSM using unscented Kalman filtering," in *IEEE International Symposium on Industrial Electronics*, July 2006, pp. 2242–2247.

[80] R. A. Freeman and P. V. Kokotovic, "Lyapunov design," *The Control Handbook*, vol. 77, pp. 932–940, 1996.

[81] C. Olalla, R. Leyva, A. E. Aroudi, and I. Queinnec, "Robust LQR control for PWM converters : An LMI approach," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 7, pp. 2548–2558, Jul. 2009.

[82] T. Qi, L. Xing, and J. Sun, "Dual-boost single-phase PFC input current control based on output current sensing," *IEEE Transactions on Power Electronics*, vol. 24, no. 11, pp. 2523–2530, Nov. 2009.

[83] J. Chiasson and B. Vairamohan, "Estimating the state of charge of a battery," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 3, pp. 465–470, May 2005.

[84] F. Oettmeier, J. Neely, S. Pekarek, R. DeCarlo, and K. Uthaichana, "MPC of switching in a boost converter using a hybrid state model with a sliding mode observer," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3453–3466, Sep. 2009.

[85] S.-C. Tan, Y. Lai, and C. Tse, "Indirect sliding mode control of power converters via double integral sliding surface," *IEEE Transactions on Power Electronics*, vol. 23, no. 2, pp. 600–611, Mar. 2008.

[86] H.-S. Song and K. Nam, "Dual current control scheme for PWM converter under unbalanced input voltage conditions," *IEEE Transactions on Industrial Electronics*, vol. 46, no. 5, pp. 953–959, Oct. 1999.

[87] L. Guo, J. Hung, and R. Nelms, "Evaluation of DSP-based PID and fuzzy controllers for DC-DC converters," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 2237–2248, Jun. 2009.

[88] J. Morales-Saldaa, J. Leyva-Ramos, E. Carbajal-Gutierrez, and M. Ortiz-Lopez, "Average current-mode control scheme for a quadratic buck converter with a single switch," *IEEE Transactions on Power Electronics*, vol. 23, no. 1, pp. 485–490, Jan. 2008.

[89] P. Thounthong, S. Rael, and B. Davat, "Control algorithm of fuel cell and batteries for distributed generation system," *IEEE Transactions on Energy Conversion*, vol. 23, no. 1, pp. 148–155, Mar. 2008.

[90] B. Singh and J. Solanki, "An implementation of an adaptive control algorithm for a three-phase shunt active filter," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 8, pp. 2811–2820, Aug. 2009.

[91] L. Tzann-Shin, "Input-output linearization and zero-dynamics control of three-phase ac/dc voltage-source converters," *IEEE Transactions on Power Electronics*, vol. 18, no. 1, pp. 11–22, Jan. 2003.

[92] ——, "Lagrangian modeling and passivity-based control of three-phase ac/dc voltage-source converters," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 4, pp. 892–902, Jan. 2004.

[93] A. Payman, S. Pierfederici, and F. Meibody-Tabar, "Energy management in a fuel cell/supercapacitor multisource/multiload electrical hybrid system," *IEEE Transactions on Power Electronics*, vol. 24, no. 12, pp. 2681–2691, Dec. 2009.

[94] K.-H. Cheng, C.-F. Hsu, C.-M. Lin, T.-T. Lee, and C. Li, "Fuzzy-neural sliding-mode control for DC-DC converters using asymmetric gaussian membership functions," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1528–1536, Mar. 2007.

[95] T. Hansen and C.-J. Wang, "Support vector based battery state of charge estimator," *Journal of Power Sources*, vol. 141, no. 2, pp. 351–358, Mar. 2005.

[96] S. Duryea, S. Islam, and W. Lawrance, "A battery management system for stand-alone photovoltaic energy systems," *IEEE Industry Applications Magazine*, vol. 7, no. 3, pp. 67–72, Jun. 2001.

[97] I.-S. Kim, "Nonlinear state of charge estimator for hybrid electric vehicle battery," *IEEE Transactions on Power Electronics*, vol. 23, no. 4, pp. 2027–2034, July 2008.

[98] A. Szumanowski and Y. Chang, "Battery management system based on battery nonlinear dynamics modeling," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1425–1432, May 2008.

[99] P. van Bree, A. Veltman, W. Hendrix, and P. van den Bosch, "Prediction of battery behavior subject to high-rate partial state of charge," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 588–595, Feb. 2009.

[100] D.-T. Lee, S.-J. Shiah, C.-M. Lee, and Y.-C. Wang, "State-of-charge estimation for electric scooters by using learning mechanisms," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 544–556, Mar. 2007.

[101] S. Teleke, M. Baran, S.Bhattacharya, and A. Huang, "Optimal Control of Battery Energy Storage for Wind Farm Dispatching," *IEEE Transactions on Energy Conversion*, vol. 25, no. 3, pp. 787–794, Sep. 2010.

[102] M. Tekin, D. Hissel, M.-C. Pera, and J. Kauffmann, "Energy-Management Strategy for Embedded Fuel-Cell Systems Using Fuzzy Logic," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 595–603, Feb. 2007.

[103] J. Moreno, M. Ortuzar, and J. Dixon, "Energy-management system for a hybrid electric vehicle, using ultracapacitors and neural networks," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 2, pp. 614–623, Mar. 2006.

[104] S. Pourmousavi, M. Nehrir, C. Colson, and C. Wang, "Real-Time Energy Management of a Stand-Alone Hybrid Wind-Microturbine Energy System Using Particle Swarm Optimization," *IEEE Transactions on Sustainable Energy*, vol. 1, no. 3, pp. 193–201, Oct. 2010.

[105] J. Lagorse, M. Simoes, and A. Miraoui, "A Multiagent Fuzzy-Logic-Based Energy Management of Hybrid Systems," *IEEE Transactions on Industry Applications*, vol. 45, no. 6, pp. 2123–2129, Nov./Dec. 2009.

[106] H. Yoo, S.-K. Sul, Y. Park, and J. Jeong, "System Integration and Power-Flow Management for a Series Hybrid Electric Vehicle Using Supercapacitors and Batteries," *IEEE Transactions on Industry Applications*, vol. 44, no. 1, pp. 108–114, Jan./Feb. 2008.

[107] M. Camara, H. Gualous, F. Gustin, and A. Berthon, "Design and New Control of DC/DC Converters to Share Energy Between Supercapacitors and Batteries in Hybrid Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 5, pp. 2721–2735, Sep. 2008.

[108] H. Chaoui and P. Sicard, "Adaptive Fuzzy Logic Motion and Posture Control of Inverted Pendulums with Unstructured Uncertainties," in *IEEE Automation Science and Engineering International Conference*, 2010, pp. 638–643.

[109] ——, "Motion and Balance Neural Control of Inverted Pendulums with Nonlinear Friction and Disturbance," in *IEEE Canadian Conference on Electrical and Computer Engineering*, 2011, pp. 1222–1227.

[110] H. Chaoui, P. Sicard, and M. Sawan, "High Precision ANN-Based Adaptive Displacement Tracking of Piezoelectric Actuators for MEMS," in *IEEE Circuits and Systems International Conference*, 2010, pp. 85–88.

[111] H. Chaoui and P. Sicard, "Sensorless Neural Network Speed Control of Permanent Magnet Synchronous Machines with Nonlinear Stribeck Friction," in *IEEE/ASME Advanced Intelligent Mechatronics International Conference*, 2010, pp. 926–931.

[112] ——, "Adaptive Lyapunov-Based Neural Network Sensorless Control of Permanent Magnet Synchronous Machines," *Neural Computing and Applications*, vol. 20, no. 5, July 2011.

[113] ——, "Robust ANN-Based Nonlinear Speed Observer for Permanent Magnet Synchronous Machine Drives," in *IEEE International Conference on Electric Machines and Drives*, 2011, pp. 597–602.

[114] ——, "Adaptive Fuzzy Logic Control of Permanent Magnet Synchronous Machines with Nonlinear Friction," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 12, December 2011.

[115] H. Chaoui, S. Miah, and P. Sicard, "Adaptive Fuzzy Logic Control of a DC-DC Boost Converter with Large Parametric and Load Uncertainties," in *IEEE/ASME Advanced Intelligent Mechatronics International Conference*, 2010, pp. 757–580.

[116] H. Chaoui and P. Sicard, "Adaptive DC Bus Voltage Control for Renewable Energy Production Systems with uncertainties," in *IEEE International Conference on Industrial Technology & Southeastern Symposium on System Theory*, 2011, pp. 36–39.

[117] ——, "Accurate State of Charge (SOC) Estimation for Batteries using a Reduced-order Observer," in *IEEE International Conference on Industrial Technology & Southeastern Symposium on System Theory*, 2011, pp. 31–35.

[118] H. Chaoui, P. Sicard, and H. Ndjana, "Adaptive State of Charge (SOC) Estimation for Batteries with Parametric Uncertainties," in *IEEE/ASME Advanced Intelligent Mechatronics International Conference*, 2010, pp. 703–707.

[119] H. Chaoui and P. Sicard, "Fuzzy Logic Based Supervisory Energy Management for Multisource Electric Vehicles," in *IEEE International Conference on Vehicle Power and Propulsion*, 2011, pp. 1–5.

Publications

2011 :

1. H. Chaoui, and P. Sicard : **"Adaptive Fuzzy Logic Control of Permanent Magnet Synchronous Machines with Nonlinear Friction"**, IEEE Transactions on Industrial Electronics, Volume : 59, Issue : 2, pages 1123-1133, February 2012.
2. H. Chaoui, and P. Sicard : **"Adaptive Control of Permanent Magnet Synchronous Machines with Disturbance Estimation"**, Journal of Control Theory and Applications, (accepted), September 2011.
3. H. Chaoui, and P. Sicard : **"Adaptive Displacement Tracking Control of Piezo-Actuated Manipulation Mechanisms with Hysteresis and Disturbance"**, IEEE International Workshop on RObotic and Sensors Environments (ROSE 2011), Montreal, Quebec, Canada, pages 49-53, September 2011.
4. H. Chaoui, and P. Sicard : **"Fuzzy Logic Based Supervisory Energy Management for Multisource Electric Vehicles"**, IEEE Vehicle Power and Propulsion Conference (VPPC 2011), Chicago, Illinois, USA, pages 1-5, September 2011.
5. H. Chaoui, and P. Sicard : **"Adaptive Lyapunov-Based Neural Network Sensorless Control of Permanent Magnet Synchronous Machines"**, Neural Computing and Applications, Springer, Volume : 20, Issue : 5, pages 717-727, July 2011.
6. H. Chaoui, and P. Sicard : **"Robust ANN-Based Nonlinear Speed Observer for Permanent Magnet Synchronous Machine Drives"**, IEEE International Conference on Electric Machines and Drives (IEMDC 2011), Niagara Falls, Ontario, Canada, pages 597-602, May 2011.
7. H. Chaoui, and P. Sicard : **"Motion and Balance Neural Control of Inverted Pendulums with Nonlinear Friction and Disturbance"**, IEEE Canadian Conference in Electrical and Computer Engineering (CCECE 2011), Niagara Falls, Ontario, Canada, pages 1222-1227, May 2011.
8. H. Chaoui, and P. Sicard : **"Adaptive DC Bus Voltage Control for Renewable Energy Production Systems with uncertainties"**, IEEE International Conference on Industrial Technology (ICIT-SSST 2011), Auburn, Alabama, USA, pages 40-44, March 2011.
9. H. Chaoui, and P. Sicard : **"Accurate State of Charge (SOC) Estimation for Batteries using a Reduced-order Observer"**, IEEE International Conference on Industrial Technology (ICIT-SSST 2011), Auburn, Alabama, USA, pages 35-39, March 2011.

2010 :

1. H. Chaoui, and P. Sicard : **"Adaptive Fuzzy Logic Motion and Posture Control of Inverted Pendulums with Unstructured Uncertainties"**, IEEE International Conference on Automation Science and Engineering, CASE 2010, Toronto, Ontario, Canada, pages 638-643, August 2010.
2. H. Chaoui, and P. Sicard : **"Sensorless Neural Network Speed Control of Permanent Magnet Synchronous Machines with Nonlinear Stribeck Friction"**, IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 2010, Montreal, Quebec, Canada, pages 926-931, July 2010.
3. H. Chaoui, P. Sicard, and H. J. Nanga Ndjana : **"Adaptive State of Charge (SOC) Estimation for Batteries with Parametric Uncertainties"**, IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 2010, Montreal, Quebec, Canada, pages 703-707, July 2010.
4. H. Chaoui, S. Miah, and P. Sicard : **"Adaptive Fuzzy Logic Control of a DC-DC Boost Converter with Large Parametric and Load Uncertainties"**, IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 2010, Montreal, Quebec, Canada, pages 575-580, July 2010.
5. H. Chaoui, P. Sicard, and M. Sawan : **"High Precision ANN-Based Adaptive Displacement Tracking of Piezoelectric Actuators for MEMS"**, IEEE International Conference on Circuits and Systems, NEWCAS 2010, Montreal, Quebec, Canada, pages 85-88, June 2010.

Annexe A

Résumé

A.1 Introduction

Dans la littérature, les approches de commande des systèmes non-linéaires peuvent être divisées en trois catégories. La première catégorie consiste en une linéarisation des systèmes non-linéaires autour d'un point de fonctionnement [1]. Dans ce cas, les lois de commande linéaires classiques sont appliquées pour le système approximé. En dépit de la simplicité des lois de commande, la performance du système de commande n'est pas garantie pour l'ensemble du système. La deuxième catégorie porte sur la conception des contrôleurs non-linéaires basés sur la dynamique des systèmes non-linéaires. Dans cette catégorie, les caractéristiques des systèmes non-linéaires sont conservées. Toutefois, les difficultés de conception augmentent avec la complexité de la dynamique des systèmes non-linéaires [2]. En outre, ces approches se basent sur un modèle mathématique précis du système et ont tendance à bien fonctionner en théorie. Mais, leurs performances se dégradent en présence de conditions de fonctionnement variables, d'incertitudes dynamiques et des perturbations externes. En pratique, dériver un modèle mathématique précis des processus industriels complexes pourrait être une tâche difficile à entreprendre. En outre, d'autres facteurs pourraient être imprévisibles, tels que le bruit, la température et la variation des paramètres. Par conséquent, la dynamique du système ne peut pas être basée avec efficacité sur des modèles mathématiques précis. La troisième catégorie met en oeuvre des contrôleurs non-linéaires basés sur l'intelligence artificielle, tels que les réseaux de neurones artificiels (RNA) et la logique floue [3–8]. Ces techniques ont été crédités dans diverses applications comme des outils puissants capables de fournir une robuste approximation pour les systèmes mathématiquement mal définis qui peuvent être soumis à des incertitudes structurées et non structurées [9, 10]. Le théorème d'approximation universelle a été la principale force motrice derrière la popularité croissante de ces méthodes puisqu'ils sont théoriquement capables d'approximer uniformément toute fonction continue réelle à n'importe quel degré de précision. Divers réseaux de neurones artificiels et modèles de logique floue ont été proposées pour résoudre de nombreux problèmes complexes, qui ont permit d'obtenir une performance satisfaisante [11–14], offrant une alternative aux techniques de commande classiques.

Les principales contributions de cette thèse sont la conception des structures de commande, des lois d'adaptation et des preuves de stabilité pour les stratégies de commande adaptative intelligente pour obtenir : :

- Apprentissage et approximation des systèmes non-linéaires *a priori* inconnu.
  - L'identification de la dynamique des propulseurs pour satellites.
  - Approximation des incertitudes pour la commande de la pendule inversée.
  - Compensation d'hystérésis pour actionneurs piézoélectriques dans les systèmes microélectromécaniques (MEMS).
- Analyse de la stabilité pour les techniques basées sur l'intelligence artificielle.
  - Mécanisme d'apprentissage adaptatif basé sur Lyapunov pour les réseaux de neurones.
  - Technique d'adaptation stable selon Lyapunov pour les systèmes de logique flou.
- Stratégies de commande adaptative basées sur Lyapunov pour les systèmes électromécaniques tels que des manipulateurs robotiques.
  - Compensation adaptive du frottement et de ses non-linéarités.
  - Atténuation de perturbation pour assurer une robustesse aux incertitudes non-structurées.
  - Réduction des effets d'élasticité pour améliorer la stabilité interne de manipulateurs avec articulations flexibles.
  - Adaptation des paramètres pour obtenir une robustesse aux incertitudes paramétriques.

- Haute performance et robustesse des machines synchrones à aimants permanents.
  - Estimation du couple de friction et de charge pour une haute précision de suivi de trajectoire.
  - Techniques de commande adaptative pour une robustesse aux incertitudes paramétriques.
  - Approches basées sur l'intelligence artificielle pour une tolérance élevée à des incertitudes non-structurées.
  - Stratégies d'estimation de vitesse.
- Haute efficacité pour les systèmes de production d'énergie.
  - Estimation précise de l'état de charge des batteries.
  - Commande adaptative à base d'intelligence artificielle pour les convertisseurs de puissance.
  - Commande adaptative de la tension de bus CC pour améliorer l'efficacité de transfert d'énergie.

A.2 L'identification à base de réseaux de neurones

Les réseaux de neurones artificiels (RNA) sont constitués d'un ensemble d'éléments de base appelés neurones. La philosophie derrière ces systèmes est d'imiter le cerveau humain, mais l'écart entre les réseaux de neurones et le cerveau est toujours grand, dû à la complexité de ce dernier. Grâce aux développements des recherches sur le cerveau et la disponibilité des outils de simulation, les chercheurs étudièrent des ensembles de neurones formels interconnectés. Ces réseaux, déjà développés à l'époque, permettaient d'effectuer quelques opérations logiques simples. Jusqu'aux années 1980, la recherche était freinée par la limitation théorique du perceptron. Peu après cette époque, Hopfield lança de nouveau en 1982 la recherche dans ce domaine après avoir montré l'analogie entre les RNA et les systèmes physiques. Après les années 1990, quelques travaux scientifiques ont vu le jour dans le domaine de la commande des systèmes d'entraînement et des systèmes de positionnement de haute performance.

Les approches de commande utilisant les réseaux de neurones artificiels tentent d'imiter la structure connexionniste du système nerveux. Leur caractéristique fondamentale réside dans le fait que les fonctions de mémoire et de traitement y sont intimement liées, cela est à l'instar du cerveau qui autorise un certain flou et des imprécisions qui n'affectent pas la fiabilité de l'ensemble d'où une sûreté de fonctionnement ainsi qu'une grande capacité d'adaptation, d'apprentissage et de tolérance aux pannes. Il n'est donc pas surprenant que les recherches dans le domaine de la commande utilisent les réseaux de neurones artificiels avec des résultats satisfaisants. Cependant, ces tentatives ont été fortement critiquées. Une critique ayant fortement influencé les recherches dans ce domaine soutient que les réseaux sont incapables par leur nature de représenter les structures essentielles à la cognition. Ils auraient le double effet de synthétiser les difficultés auxquelles les réseaux font face et de tirer des enseignements sur les différentes tendances actuelles. Les RNA sont une formulation mathématique simplifiée des neurones biologiques. Ils ont la capacité de mémorisation, de généralisation et surtout d'apprentissage qui est le phénomène le plus important.

- Avantages : Les réseaux de neurones ont une capacité d'adaptabilité et d'auto organisation ainsi que de résoudre des problèmes non-linéaires avec une bonne approximation. Ils offrent une bonne robustesse aux bruits et se prêtent bien à une implantation parallèle. La rapidité d'exécution est une qualité importante et elle justifie souvent à elle seule le choix d'implanter un réseau de neurones.
- Inconvénients : La difficulté d'interpréter le comportement d'un réseau de neurones est un inconvénient pour la mise au point d'une application. Il est souvent impossible d'utiliser les résultats obtenus pour améliorer ce comportement. Il est également hasardeux de généraliser à partir d'expériences antérieures et de conclure ou de créer des règles sur le fonctionnement et le comportement des réseaux de neurones. Ces outils sont basés sur des d'heuristiques et plusieurs paramètres doivent être ajustés. De plus, aucune méthode ne permet de choisir des valeurs optimales et la stabilité de la plupart des algorithmes d'apprentissage n'est pas garantie.

Malgré la diversité des algorithmes d'apprentissage il n'y a pas de méthode systématique pour la détermination d'un réseau de neurones, en particulier le choix de l'architecture du réseau, le nombre de neurones, le nombre de couches ou le choix des paramètres internes de l'algorithme comme l'erreur quadratique qu'on veut atteindre et le nombre d'itérations. Théoriquement, un réseau qui possède deux couches cachées avec suffisamment de nIJuds cachés peut être utilisé pour résoudre n'importe quel problème. En ce qui concerne le nombre de neurones cachés (de l'ensemble des couches cachées), il faut qu'il soit supérieur au plus grand nombre de neurones constituant soit la couche d'entrée soit la couche de sortie. Finalement, les réseaux de neurones sont des programmes informatiques qui en simulant le fonctionnement des neurones du cerveau humain, permettent aux ordinateurs d'apprendre à effectuer certaines tâches simples. Par contre, ils sont très gourmands en capacité de calcul et nécessitent des ordinateurs très puissants pour apprendre à exécuter des tâches complexes.

Un problème est rencontré lorsque l'apprentissage converge vers une solution sous optimale. Ce type de problème est difficile à résoudre car généralement la surface d'erreur est inconnue. Le nombre de neurones cachés est particulièrement important parce qu'il détermine la capacité de calcul du réseau. Un nombre insuffisant de neurones cachés peut compromettre la capacité du réseau à résoudre le problème. Inversement, trop de neurones permettent au réseau d'apprendre par cIJur au détriment des performances de la généralisation. Lorsque l'apprentissage d'un réseau reflète trop les particularités du problème au détriment de la tâche réelle, la matrice d'apprentissage ne reflète pas toujours adéquatement la tâche. Il en résulte que le réseau généralise mal. Même les paramètres propres à l'algorithme d'apprentissage sont difficiles à choisir comme par exemple le pas d'apprentissage. Cette thèse contribue aux développements d'algorithmes d'apprentissage assurant la stabilité du système de commande en boucle fermée.

A.2.1 Application aux nanosatellites

Les nanosatellites utilisent un petit système de propulsion avec des capacités de propulsion élevée. De nombreux chercheurs ont étudié les technologies de propulsion pour identifier ceux qui maintiennent des performances élevées à petite échelle. Toutefois, indépendamment de la taille des propulseurs, ces derniers émettent en échappement un contaminent néfaste pour les engins spatiaux voisins. L'agence spatiale canadienne (ASC) a lancé, comme une collaboration internationale avec l'agence japonaise d'exploration aérospatiale, une étude de faisabilité de la mission du nanosatellite JC2Sat-FF. L'objectif est de maintenir la formation d'engins spatiaux (nanosatellites) avec la traînée aérodynamique. Le principal avantage est l'élimination du risque de contamination des propulseurs. De plus, la durée de vie des missions ne sera pas limitée par la quantité de carburant à bord. Cependant, les temps de réponse aux changements dans la formation sont lents, les positions relatives ne peuvent être contrôlées avec une précision très élevée et certains types de formation ne sont pas possibles. Une étude théorique, algorithmique et expérimentale de la formation de satellites est menée à l'ASC, où, un banc d'essai a été développé pour émuler un scénario de formation typique pour la validation expérimentale.

Les propulseurs ont reçu récemment une attention particulière et ont été utilisés dans de nombreuses applications telles que, les vaisseaux de haute performance et de la robotique sous-marine. Les non-linéarités des propulseurs rendent la modélisation de leur dynamique une tâche complexe. L'utilisation de modèles simplifiés des propulseurs résultent en de mauvaises performances en raison de non-linéarités et de la dynamique variable dans le temps. Donc, un système de contrôle avancé avec des capacités d'apprentissage est requis pour s'adapter aux changements dans la dynamique des propulseurs.

Cette application démontre la puissance des réseaux de neurones en identification et en apprentissage pour des nanosatellites (section 1.2). Cette technique est utilisée pour l'approximation de la dynamique des propulseurs de satellites. Sa performance est étudiée en utilisant des données expérimentales d'un propulseur. Les résultats montrent une précision d'approximation très élevée, ce qui confirme la capacité des réseaux de neurones à modéliser des systèmes mal définis mathématiquement. Cette observation sera utilisée dans cette thèse pour la conception des techniques de commande adaptative intelligente. Ces stratégies devront garantir la stabilité des algorithmes d'adaptation et seront appliquées à des divers systèmes dynamiques complexes.

A.3 Commande adaptative à base d'intelligence artificielle

Parmi les techniques de commande connues, on trouve la commande adaptative. Elle permet à la fois d'assurer la stabilité et une bonne performance en présence d'incertitudes paramétriques. Cette approche adapte les coefficients du contrôleur en ligne pour compenser les variations dans l'environnement ou dans le système lui-même. Un grand nombre de travaux de recherche s'orientent vers la commande adaptative, car les paramètres des modèles utilisés dans la commande des systèmes d'entraînement peuvent être inconnus ou variables dans le temps. Par contre, ces techniques ne sont pas très utilisées à cause de la difficulté pour compenser des grandes perturbations. De plus, la plupart des lois de commande qui ont été proposées requièrent la connaissance de l'état complet du système et même quelques fois la mesure de l'accélération.

En 1965, le professeur Lofti Zadeh de l'université de Berkley en Californie introduit le concept de la logique floue, il déclara qu'un contrôleur électromécanique doté d'un raisonnement humain serait plus performant qu'un contrôleur classique. Ceci est dû au fait que l'être humain possède une caractéristique de raisonnement basée sur des données imprécises ou incomplètes alors que l'ordinateur est basé sur des données exactes. Un contrôleur standard demande le plus précis modèle possible du système à contrôler. Ainsi, pour mieux représenter la réalité physique, le développement d'un modèle analytique précis est requis. Par contre, un contrôleur flou ne demande pas de modèle du système à régler. Les décisions sont prises sans avoir recours au modèle analytique. Les algorithmes de réglage se basent sur des règles linguistiques de la forme Si ... Alors ...

La logique floue est utilisée dans plusieurs domaines de recherche et d'application tels que l'automatisation, l'instrumentation et le traitement d'information. La plupart des lois de commande qui ont été proposées requièrent la connaissance de l'état complet du système, donc il est important de concevoir une loi de commande robuste qui demande une connaissance minimale sur le système. Idéalement, la loi de commande donne une grande région d'attraction applicable et robuste dans le cas de l'incertitude des paramètres autonomes non-linéaires du système, d'une erreur du modèle, de la variation de la charge et d'une caractéristique inconnue.

La structure de commande présentée dans la Fig. A.1 est utilisée dans cette thèse comme une structure de base pour concevoir des contrôleurs intelligents pour des systèmes dynamiques non-linéaires. Il est à noter qu'un contrôleur de rétroaction pourrait être inclus pour des problèmes plus complexes.

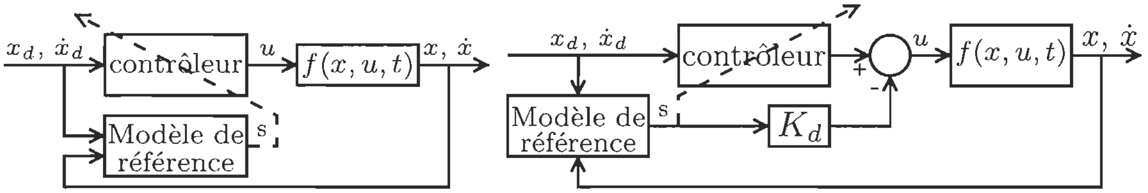


FIGURE A.1 – Structures de commande générales

A.3.1 Application à la pendule inversée

Le problème de la pendule inversée a été largement utilisé pour démontrer l'efficacité de différents types de contrôleurs. Il est considéré comme un défi de référence bien établi pour de nombreux problèmes de commande, tels que les manipulateurs robotiques et les missiles.

L'approche de commande est conçue pour les systèmes non-linéaires en général à plusieurs entrées-sorties (MIMO). Dans cette stratégie, le système de commande capitalise sur les capacités d'apprentissage et de généralisation des réseaux neurones et de la logique floue pour assurer un suivi précis de trajectoire en présence des incertitudes non-structurées. Une adaptation neurale et floue basée sur des stratégies de commande adaptative a été appliquée au problème de la pendule inversée (section 5.4). Les structures de commande sont présentées dans la Fig. A.2 et Fig. A.3. Les résultats montrent de bonnes performances en présence de non-linéarités de friction et les perturbations externes.

A.3.2 Application aux actionneurs piézoélectriques

Les matériaux piézoélectriques sont habituellement utilisés dans la conception d'actionneurs et de capteurs de haute précision pour des micro et nanosystèmes. Cependant, leur perfor-

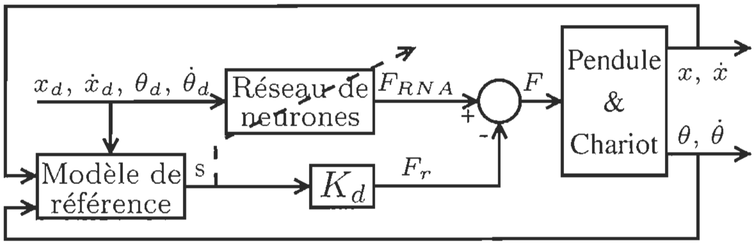


FIGURE A.2 – Structure de commande neuronale pour la pendule inversée

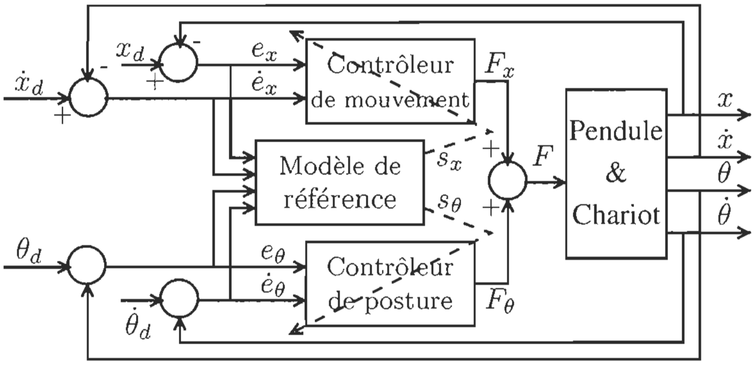


FIGURE A.3 – Structure de commande floue pour la pendule inversée

mance est limitée par la présence d'hystérésis. En effet, les caractéristiques de ce phénomène sont généralement inconnues et peuvent mener à des cycles limites et même l'instabilité à défaut de compenser cette incertitude. Il est donc urgent d'envisager des approches de commande pour ce type de systèmes.

La stratégie d'apprentissage à base de RNA est alors appliquée comme le montre la Fig. A.4 (section 5.5). Les résultats montrent que la position et la vitesse de l'actionneur piézoélectrique peuvent être contrôlées avec une haute précision. Le contrôleur proposé démontre son efficacité dans la compensation d'hystérésis. Contrairement à d'autres contrôleurs, la stabilité est assurée avec la théorie de Lyapunov.

Jusqu'à présent, les techniques d'apprentissage neuronales et floues à base de Lyapunov ont été proposées pour des systèmes non-linéaires sans tenir compte de leur dynamique. La stabilité de ces lois d'adaptation universelle est étudiée en utilisant la théorie de stabilité de Lyapunov. Toutefois, certains systèmes non-linéaires peuvent introduire des contraintes supplémentaires sur la stabilité de la commande et, par conséquent, il est important de considérer la dynamique

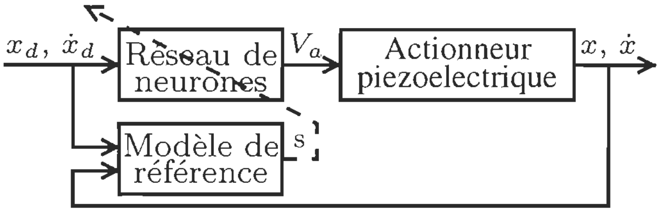


FIGURE A.4 – Structure de commande neuronale pour les actionneurs piézoélectriques

d'un système lors de l'étude de sa stabilité.

A.4 Commande adaptative des manipulateurs robotiques

Au fil des années, de nombreuses stratégies de commande adaptative ont été proposées pour les manipulateurs rigides. Ces développements représentent un pas important vers les applications robotiques de haute précision. Cette thèse utilise, comme point de départ, l'une des techniques les plus populaires d'adaptation [22] pour la conception de techniques de commande adaptative robuste capable de faire face à différents types d'incertitudes (section 6.3).

Bien que cette technique offre un rendement satisfaisant, elle néglige des aspects importants tels que le frottement et la perturbation. Cela a des effets négatifs sur les performances et la stabilité de ces systèmes, qui est considéré comme le problème majeur de la commande adaptative classique. Par la suite, nous allons résoudre ces inconvénients avec la conception de compensateurs de friction et de perturbation pour les manipulateurs à articulations flexibles.

La commande adaptative des robots manipulateurs rigides peut aussi être utilisée pour les articulations flexibles. Cela a été rendu possible grâce à la théorie des perturbations singulières. Cependant, il suppose une grande rigidité. En outre, la présence de la friction et de la perturbation affecte de manière significative les performances et la stabilité. Avec l'hypothèse d'une grande rigidité, un contrôleur rigide utilisé en anticipation approxime le modèle inverse de l'articulation flexible. D'autre part, un compensateur de friction et de perturbation est utilisé en rétroaction pour éliminer les erreurs résiduelles. Les structures de commande sont présentées dans la Fig. A.5 et Fig. A.6.

Les résultats mettent en évidence la qualité de la compensation de la flexibilité, les non-linéarités de friction et la perturbation. Un suivi précis de la trajectoire de la charge est obtenu et la stabilité interne, un problème potentiel avec un tel système, est également atteinte. En outre, la stabilité est prouvée par la méthode directe de Lyapunov (section 6.4 and 6.5).

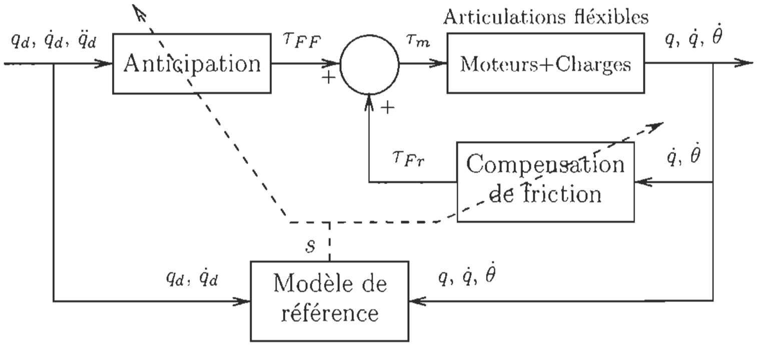


FIGURE A.5 – Structure de commande adaptative de compensation de friction

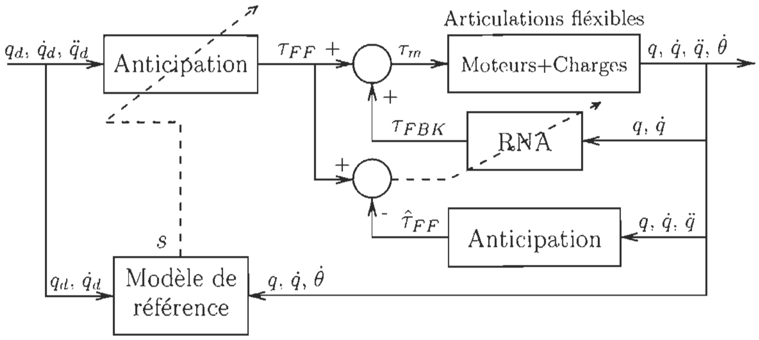


FIGURE A.6 – Structure de commande adaptative neuronale de compensation de perturbation

D'autre part, une comparaison entre un contrôleur flou de type-1 et de type-2 a été menée sur une articulation flexible en présence d'incertitudes de diverses grandeurs. Pour ce faire, un contrôleur flou a été proposé comme le montre la Fig. A.7. De plus, une adaptation basée sur Lyapunov a été introduite sur la Fig. A.8 comme une alternative aux méthodes traditionnelles d'adaptation heuristiques. Dans les deux approches de commande, les résultats montrent la supériorité du type-2 avec un meilleur amortissement des oscillations due à l'élasticité. Cette constatation confirme une plus grande tolérance des régulateurs flous à une modélisation imprécise. Jusqu'à présent, ce travail est l'un des rares tentatives dans la conception et la mise en oeuvre d'une architecture de commande floue de type-2 pour les manipulateurs avec articulations flexibles (section 6.6 and 6.7).

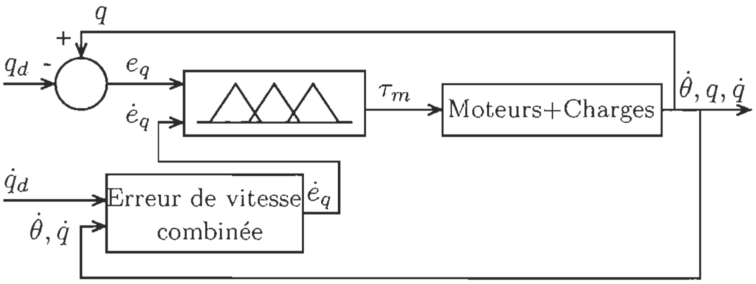


FIGURE A.7 – Structure de commande floue pour les manipulateurs robotiques

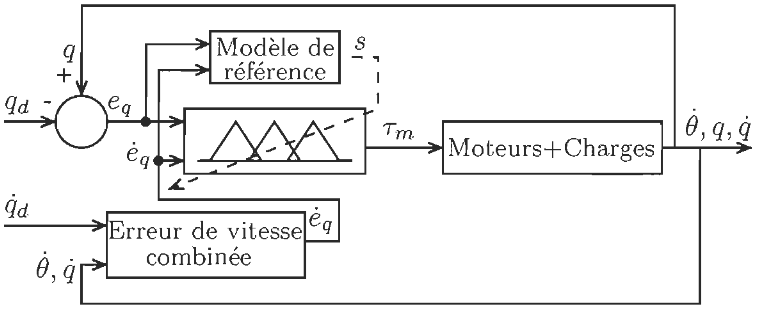


FIGURE A.8 – Structure de commande adaptative floue pour les manipulateurs robotiques

A.5 Les machines synchrones à aimants permanents

Les machines synchrones à aimants permanents sont largement utilisées dans de nombreuses applications industrielles comme les véhicules électriques, les éoliennes et la robotique. Toutefois, afin de fonctionner efficacement, des encodeurs à haute résolution sont requis. Ces capteurs attachés à l'arbre du rotor augmentent la longueur de la machine, les coûts, l'inertie du rotor et nécessitent un câblage additionnel. Récemment, les techniques sans capteurs ont reçu un intérêt croissant pour des applications industrielles où il ya des limites à l'utilisation de capteurs. En outre, la commande sans capteurs réduit la sensibilité aux bruits et aux vibrations, le coût, la taille et l'entretien tout en augmentant la fiabilité du système dans son ensemble et la robustesse. Toutefois, le contrôle de tels systèmes se heurte encore à de nombreux défis qui doivent être abordées tels que les conditions de fonctionnement variables, les perturbations externes et les incertitudes dynamiques structurées et non-structurées.

A.5.1 Commande adaptative

Tel que mentionné auparavant, la commande adaptative se base sur la connaissance du modèle du système à contrôler. Ce qui permet d'écrire ce modèle sous forme d'une régression ($\Phi^T W$) et de définir des lois de commande et d'adaptation pour obtenir une stabilité asymptotique. Cette technique a été combinée avec la commande vectorielle pour concevoir un contrôleur adaptatif tel que montré sur la Fig. A.9. Cette stratégie offre des bonnes performances face aux variations de paramètres (section 7.3). De plus, un observateur de vitesse et de couple de charge augmente sa robustesse lorsque la charge varie. Par contre, l'adaptation des paramètres des trois contrôleurs ajoutent à sa complexité. Une version simplifiée est alors proposée dans la Fig. A.10 où deux contrôleurs adaptatifs sont combinés. Les résultats démontrent une performance similaire avec une réduction de complexité (section 7.4).

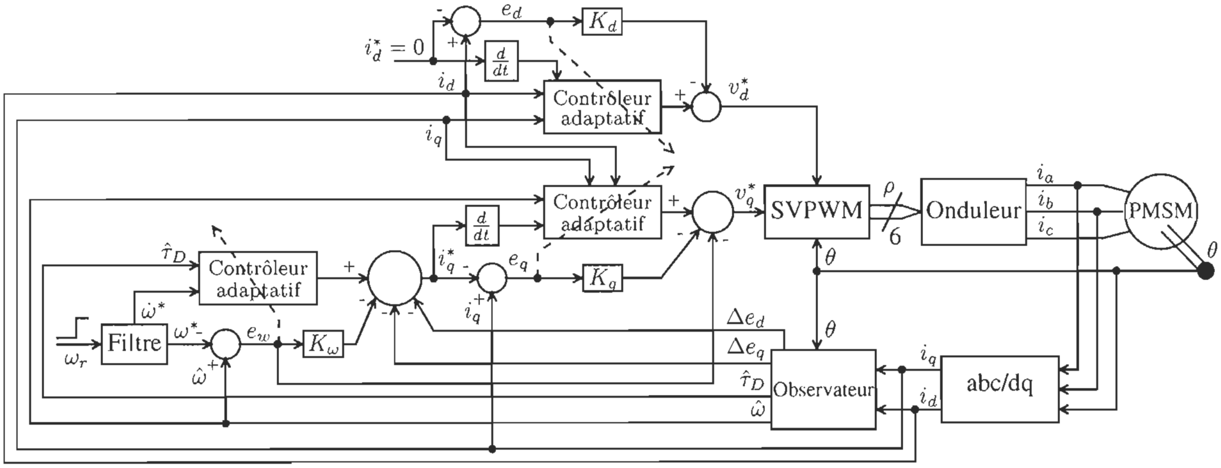


FIGURE A.9 – Structure de commande adaptative vectorielle pour les machines synchrones

Par ailleurs, une connaissance partielle du système nous permet d'approximer le vecteur de régression Φ par $\hat{\Phi}$. Cette approximation introduit une incertitude qui se traduit en une

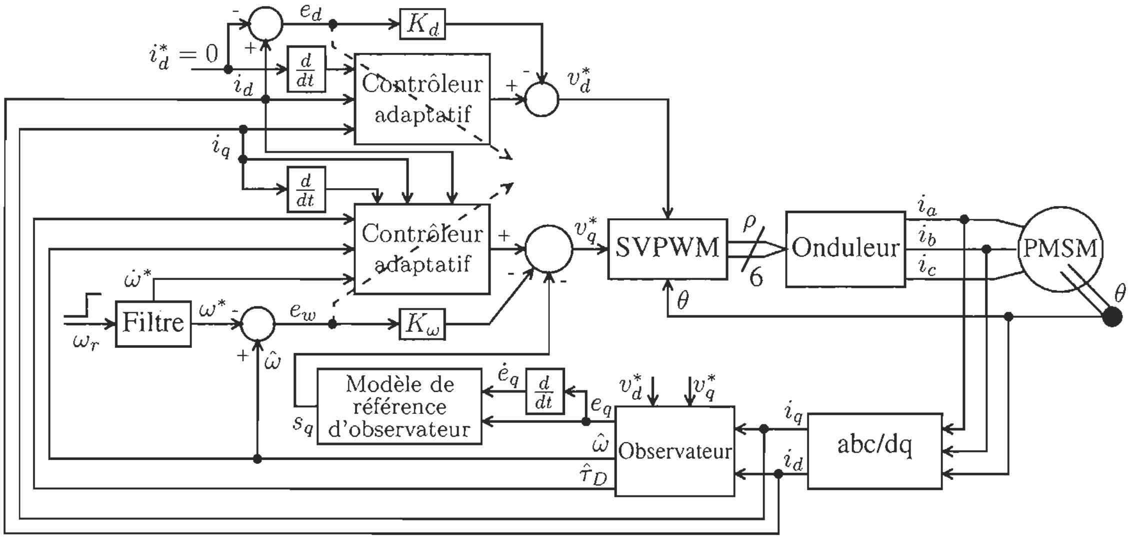


FIGURE A.10 – Structure de commande adaptative pour les machines synchrones

contrainte pour la stabilité. Donc, la stabilité asymptotique ne peut plus être garantie mais plutôt une stabilité en région. Il a aussi été démontré que cette région est définie par l'erreur d'approximation qu'au fur et au mesure qu'elle tend vers zéro, la région devient de plus en plus petite. Basé sur une connaissance partielle de la machine, un seul contrôleur adaptatif a été développé (section 7.5). La structure de commande est présentée dans la Fig. A.11. La simplicité de cette méthode rend l'adaptation des paramètres plus facile, ce qui se traduit en de meilleures performances au niveau de la commande. De plus, la stabilité du contrôleur proposée est garantie par la théorie de stabilité de Lyapunov.

A.5.2 Commande intelligente

Plusieurs contrôleurs ont été développés à base d'intelligence artificielle. Puisque ces derniers ne nécessitent pas une connaissance du modèle du système à contrôler, il était difficile de prouver leur stabilité. Par contre, nous avons déjà prouvé une stabilité en région lors d'une connaissance partielle du système. Donc, l'objectif est d'utiliser des contrôleurs neuronaux et flous pour approximer la dynamique du système, en d'autres mots, le vecteur de régression Φ. Cette approximation ($\hat{\Phi}$) nous permet alors de concevoir des contrôleurs stables basés sur

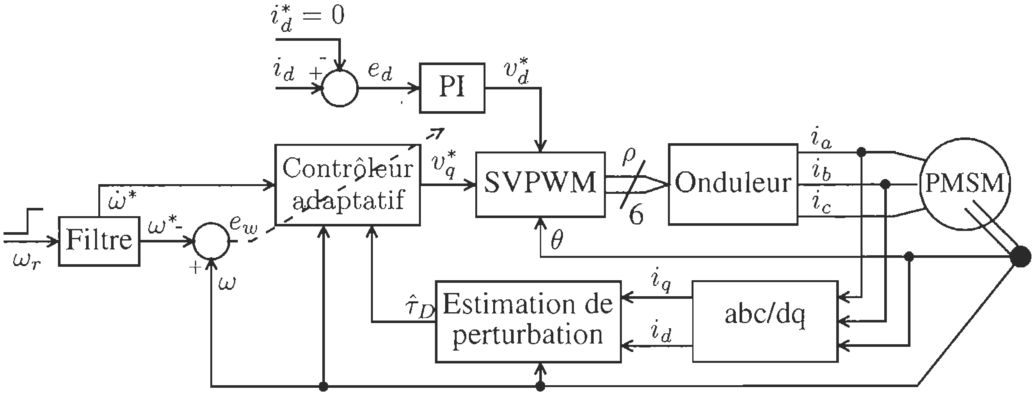


FIGURE A.11 – Structure de commande adaptative avec incertitudes pour les machines synchrones

la théorie des réseaux de neurones et de la logique floue. Ainsi, plusieurs contrôleurs basés sur différentes structures de commande ont été développés et une meilleure robustesse a été obtenue. De plus, des comparaisons avec des techniques de commande classiques confirment la valeur ajoutée des contrôleurs proposés.

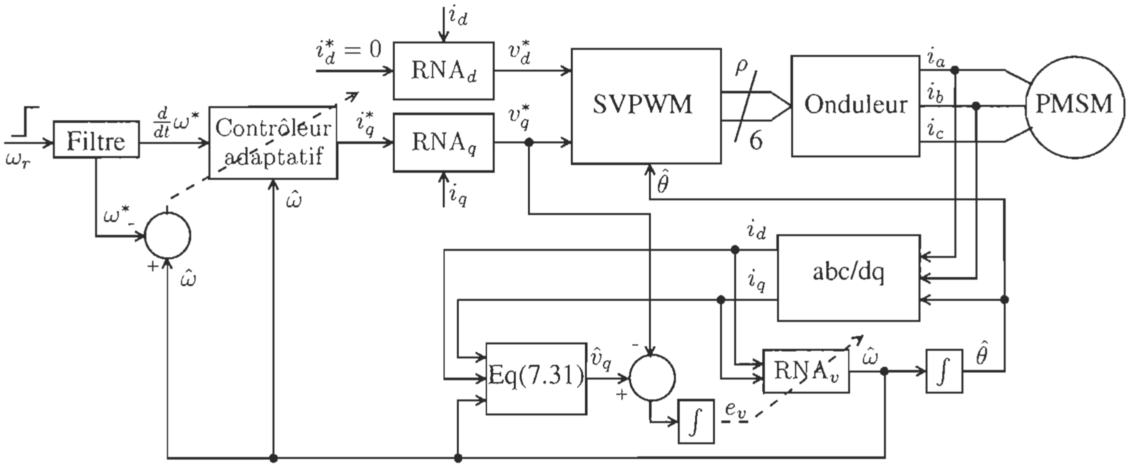


FIGURE A.12 – Structure de commande neuronale sans capteur pour les machines synchrones

Deux réseaux de neurones sont utilisés pour le contrôle du courant sur les axes d-q puisque les paramètres électriques de la machines sont variables dans le temps (section 7.6). Comme le montre la structure de commande de la Fig. A.12, un observateur neuronale de vitesse est aussi proposé. De plus, la stabilité du système de contrôle en boucle fermée (contrôleurs + ob-

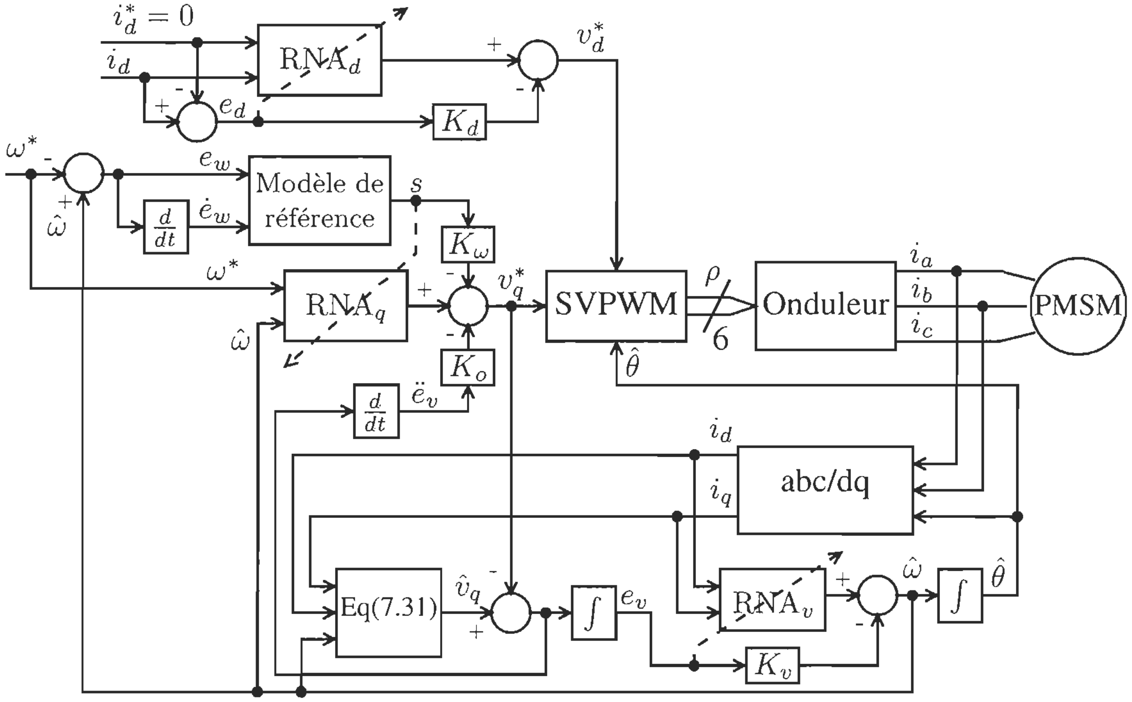


FIGURE A.13 – Structure de commande adaptative neuronale pour les machines synchrones

servateur) est garantie par Lyapunov. Les résultats montrent une grande précision et tolérance à des incertitudes non-structurées de différentes grandeurs. Une version améliorée démontre qu'une performance similaire peut être atteinte avec un contrôleur en moins comme le montre la Fig. A.13 (section 7.7). Toutefois, l'estimation de la vitesse montre une sensibilité aux variations de flux. Par conséquent, un observateur non-linéaire de vitesse à base de réseaux de neurones a été proposé pour résoudre le problème de sensibilité aux variations des paramètres (section 7.8). La structure de l'observateur est présentée dans la Fig. A.14.

Enfin, la robustesse à la fois des incertitudes structurées et non-structurées a été obtenue avec un seul contrôleur adaptatif flou (Fig. A.15). Contrairement aux autres techniques de commande, pas de capteur de tension ou de courant n'est nécessaire. Cependant, une perte en efficacité est prévue puisqu'aucune boucle de courant n'est utilisée. Néanmoins, ce travail est l'une des premières tentatives, s'il en a, pour atteindre de hautes performances en présence des deux incertitudes structurées et non-structurées sans boucle de régulation des courants pour les machines synchrones (section 7.9).

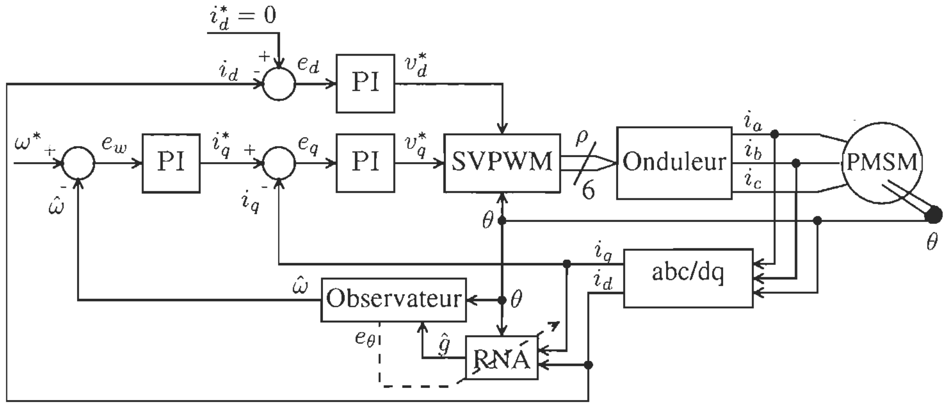


FIGURE A.14 – Structure de l'observateur neuronale de vitesse pour les machines synchrones

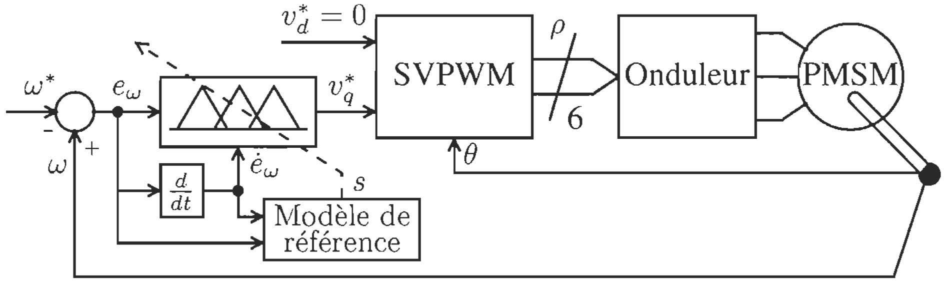


FIGURE A.15 – Structure de commande adaptative floue pour les machines synchrones

A.6 Les systèmes intelligents à base d'énergies renouvelables

Les systèmes d'énergie renouvelable sont de plus en plus populaires et considérés comme un moyen de lutte contre le changement climatique. Ils sont habituellement utilisés comme complément d'énergie pour les systèmes de production autonome. Ces systèmes sont composés de sources de stockage d'énergie telles que des batteries qui ont une quantité limitée d'énergie. Par conséquent, l'utilisation optimale de l'énergie est l'un des différents défis à relever. Cependant, un fonctionnement optimal dépend de la précision de l'estimation de l'état de charge des batteries et de la performance du contrôle des convertisseurs. Une mauvaise performance au niveau de l'estimation et du contrôle des convertisseurs résultent inévitablement en une réduction d'efficacité des systèmes d'énergie renouvelable. Il est donc urgent d'envisager de nouvelles approches de d'estimation, de commande et de gestion d'énergie pour un meilleur transfert énergétique.

Ainsi, un contrôleur adaptatif flou est proposé pour un contrôle en haute performance d'un convertisseur élévateur CC-CC comme le montre la Fig. A.16 (section 8.3). L'objectif de la commande est obtenu avec des importantes variations de paramètres et sous la contrainte de la dynamique inconnue. De plus, aucune boucle interne de régulation du courant n'est requise, ce qui réduit le nombre de capteurs. D'autre part, un contrôleur adaptatif est proposé dans la Fig. A.17 pour le contrôle du bus CC (section 8.4). La technique adaptative proposée assure un contrôle bidirectionnel du convertisseur CC-CC sans mesure de la tension de la source, ce qui résulte en moins de variations de la tension du bus CC. Les deux contrôleurs sont comparés à une structure de commande classique à base de régulateurs PI dans des conditions d'opérations similaires. Les résultats montrent la supériorité des contrôleurs proposés pour compenser des incertitudes de plus grande amplitude. La performance des stratégies proposées est un élément clé pour une haute efficacité nécessaire pour les systèmes à haute performance énergétique.

Par la suite, un estimateur d'état de charge basé de la théorie des observateurs d'états est proposé pour des batteries (section 8.5). La technique se caractérise par sa simplicité, sa preuve de stabilité et sa facilité de mise en oeuvre. Par contre, sa performance est affectée par le

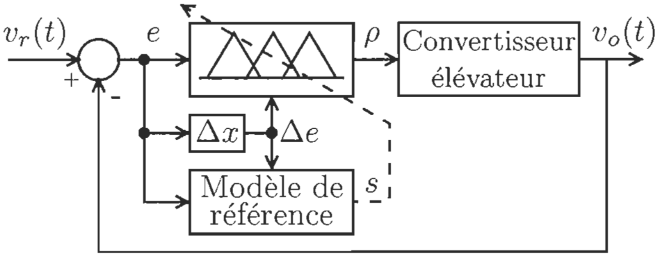


FIGURE A.16 – Structure de commande flou du convertisseur élévateur

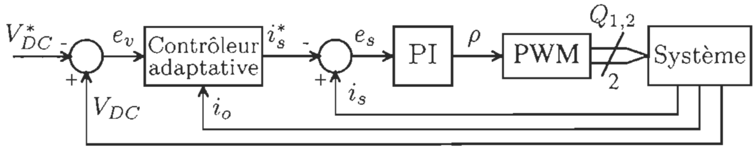


FIGURE A.17 – Structure de commande adaptative du bus CC

vieillissement des batteries d'où la nécessité d'utiliser une stratégie d'estimation en ligne des paramètres pour s'ajuster aux changements de dynamique. Donc, une technique d'estimation adaptative a été proposée et a montrée que l'état de charge peut être déterminé avec une grande précision en présence d'incertitudes (section 8.6). De plus, la convergence et la stabilité de la stratégie proposée est garantie par la théorie de Lyapunov.

Enfin, une stratégie de gestion d'énergie à base de logique floue est proposée pour les systèmes multisources (section 8.7). La technique proposée vise une utilisation optimale d'énergie ce qui se traduira en une prolongation de la durée de vie des dispositifs de stockage d'énergie et en une réduction de leur entretien fréquent. Les résultats montrent un bon équilibrage de l'état de charge des sources d'énergie. En général, lorsque les performances d'une seule source se dégradent, toutes les sources sont remplacées en même temps de préserver l'intégrité du système globale. La méthode proposée permet l'utilisation de plusieurs unités avec des capacités différentes. En outre, ça permet également de fusionner des unités de différents manufacturiers.

A.7 Conclusion

Dans cette thèse, des nouvelles structures de commande adaptative à base d'intelligence artificielle ont été proposées pour des systèmes non-linéaires à haute complexité. Les contrôleurs proposés sont conçus pour assurer la stabilité et la robustesse en présence des deux types d'incertitudes structurées et non-structurées. Premièrement, nous avons montré les capacités d'apprentissage et d'approximation des réseaux de neurones artificiels avec une stratégie d'identification de la dynamique des propulseurs de satellites. Cette capacité à approximer des systèmes non-linéaires *a priori* inconnus a été utilisée comme motivation dans le reste de la thèse avec la conception de contrôleurs intelligents.

Le chapitre 5 présente des stratégies d'adaptation neuronales et floues à base de Lyapunov. Ces mécanismes d'apprentissage sont conçus pour les systèmes MIMO en général. Une application au problème de la pendule inversée a été proposée. La technique d'adaptation neuronale a également été appliquée aux actionneurs piézoélectriques. L'approche proposée offre une bonne performance en compensation d'hystérésis, un problème connu des systèmes microélectromécaniques (MEMS).

Dans le chapitre 6, des stratégies de commande adaptative stable selon Lyapunov ont été présentées pour les manipulateurs robotiques. Dans ces techniques, la compensation de la friction et de la perturbation a permit une robustesse à des incertitudes non-structurées. La présence d'élasticité et de la variation des paramètres ont également été abordées.

Dans le chapitre 7, plusieurs méthodes d'estimation et de contrôle de vitesse ont été proposées pour les machines synchrones à aimants permanents. Ces stratégies combinent la force de la commande adaptative avec celle de l'intelligence artificielle pour atteindre une grande précision dans le suivi de trajectoire et une grande tolérance à des incertitudes non-structurées de différentes grandeurs. L'estimation de la vitesse, de la friction, de la perturbation et du couple de charge a également été obtenue à l'aide de nombreux observateurs.

Enfin, des techniques avancées d'estimation, de commande et de gestion d'énergie ont été présentées dans le chapitre 8 pour les systèmes de production d'énergie renouvelable. L'estima-

tion précise de l'état de charge des batteries est obtenue en utilisant les théories d'observateurs d'états et de commande adaptative. Par ailleurs, une commande adaptative floue des convertisseurs CC-CC a démontrée une robustesse face à de grandes incertitudes paramétriques et de charge. D'autre part, la commande adaptative de la tension du bus CC est présentée pour améliorer l'efficacité de transfert d'énergie. De plus, une stratégie de gestion d'énergie à base de logique floue des systèmes multisources est proposée. Cette technique s'ajoute aux différentes stratégies développées dans ce chapitre pour une utilisation plus optimale d'énergie renouvelable.